



**HAL**  
open science

# Techniques avancées pour la compression d'images médicales

Jonathan Taquet

► **To cite this version:**

Jonathan Taquet. Techniques avancées pour la compression d'images médicales. Réseaux et télécommunications [cs.NI]. Université Rennes 1, 2011. Français. NNT: . tel-00629429v1

**HAL Id: tel-00629429**

**<https://theses.hal.science/tel-00629429v1>**

Submitted on 14 Oct 2011 (v1), last revised 6 Jan 2012 (v3)

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

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



**THÈSE / UNIVERSITÉ DE RENNES 1**  
*sous le sceau de l'Université Européenne de Bretagne*

pour le grade de

**DOCTEUR DE L'UNIVERSITÉ DE RENNES 1**

*Mention : Traitement du signal et Télécommunications*

**École doctorale Matisse**

présentée par

**Jonathan TAQUET**

préparée à l'INRIA

Institut National de Recherche en Informatique et en Automatique  
Centre INRIA Rennes - Bretagne Atlantique

---

**Techniques  
avancées  
pour la  
compression  
d'images  
médicales**

**Version préliminaire en attente des rapports.  
Thèse à soutenir à Rennes  
le 15 Décembre 2011**

devant le jury composé de :

**Atila BASKURT**

Professeur à l'INSA Lyon / *Président*

**Jean-Pierre GUÉDON**

Professeur des Universités / *Rapporteur*

**Jean-Marie MOUREAUX**

Professeur des Universités / *Rapporteur*

**Christine CAVARO-MÉNARD**

Maître de conférences à Angers / *Examinatrice*

**Christine GUILLEMOT**

Directeur de recherches INRIA / *Examinatrice*

**Emmanuel CORDONNIER**

Président-Directeur-Général ETIAM S.A. / *Membre invité*

**Claude LABIT**

Directeur de recherches INRIA / *Directeur de thèse*



*à la mémoire de Papa*



*«Science does not aim at establishing immutable truths and eternal dogmas :  
its aim is to approach the truth by successive approximations,  
without claiming that at any stage  
final and complete accuracy has been achieved.»*

Bertrand Arthur William RUSSELL  
*The ABC of Relativity, 1925.*

## Remerciements

# Table des matières

<b>Table des matières</b>	<b>1</b>
<b>Glossaire</b>	<b>7</b>
<b>Introduction générale</b>	<b>11</b>
<b>I Contexte de l'étude</b>	<b>15</b>
<b>1 L'imagerie médicale : quelques éléments de contexte</b>	<b>17</b>
Introduction . . . . .	17
1.1 Modalités d'images ciblées . . . . .	17
1.1.1 Images médicales volumiques . . . . .	18
1.1.1.1 TDM . . . . .	19
1.1.1.2 IRM . . . . .	21
1.1.2 Lames virtuelles (LV) . . . . .	24
1.1.3 Bases d'images utilisées pour les expérimentations . . . . .	25
1.1.3.1 Images volumiques . . . . .	25
1.1.3.2 Lames virtuelles . . . . .	26
1.2 Contexte médico-légal et options d'archivage . . . . .	27
1.2.1 Contraintes quantitatives . . . . .	27
1.2.2 Archivage sans perte . . . . .	29
1.2.3 Contraintes qualitatives . . . . .	30
1.2.4 Archivage avec pertes . . . . .	30
1.2.5 Archivage avec pertes contrôlées . . . . .	31
1.2.5.1 Notion de région d'intérêt . . . . .	31
1.2.5.2 Compression presque sans perte . . . . .	31
1.2.6 Remarques . . . . .	32
1.3 Fonctionnalités intéressantes pour la télé-consultation et l'archivage . . . . .	33
1.3.1 Scalabilité en qualité . . . . .	33
1.3.2 Scalabilité en résolution . . . . .	34
1.3.3 Accès aléatoire . . . . .	34
1.3.4 Découpage en objets . . . . .	35



Conclusion . . . . .	35
<b>2 Outils génériques pour la compression d'images</b>	<b>37</b>
Introduction . . . . .	37
2.1 Compression de données . . . . .	37
2.1.1 Théorie de l'information . . . . .	37
2.1.2 Codage entropique . . . . .	40
2.1.2.1 Codes à longueur variable . . . . .	41
2.1.2.1.1 Codes de Huffman . . . . .	41
2.1.2.1.2 Codes de Golomb . . . . .	42
2.1.2.2 Codage arithmétique . . . . .	43
2.1.2.3 Autres outils de codage . . . . .	45
2.1.2.4 Quel codeur entropique utiliser ? . . . . .	46
2.2 Réduction de l'information . . . . .	46
2.2.1 Signaux numériques . . . . .	46
2.2.2 Décorrélation . . . . .	47
2.2.2.1 Prédiction . . . . .	48
2.2.2.2 Transformation . . . . .	48
2.2.2.2.1 Transformée en cosinus . . . . .	49
2.2.2.2.2 Transformée en ondelettes . . . . .	50
2.2.2.2.3 Autres transformées . . . . .	53
2.2.3 Quantification . . . . .	55
2.2.3.1 Quantification scalaire . . . . .	55
2.2.3.2 Quantification vectorielle . . . . .	57
2.2.4 Optimisation débit-distorsion . . . . .	57
2.2.4.1 Théorie débit-distorsion . . . . .	58
2.2.4.2 Mesures de distorsion . . . . .	58
2.2.4.3 Quantification scalaire par plans de bits . . . . .	60
Conclusion . . . . .	60
<b>3 Algorithmes de référence en compression d'images</b>	<b>61</b>
Introduction . . . . .	61
3.1 Références usuelles et standards de compression pour les images fixes . .	61
3.1.1 Compression prédictive . . . . .	62
3.1.1.1 Modèle DPCM . . . . .	62
3.1.1.1.1 JPEG sans perte . . . . .	63
3.1.1.1.2 JPEG-LS (LOCO-I) . . . . .	63
3.1.1.1.3 CALIC . . . . .	65
3.1.1.1.4 Autres algorithmes DPCM . . . . .	66
3.1.1.2 Modèle prédictif hiérarchique . . . . .	67
3.1.2 Compression par transformée . . . . .	68
3.1.2.1 DCT . . . . .	68
3.1.2.2 Ondelettes . . . . .	69
3.1.2.2.1 Codage inter-bande . . . . .	70

3.1.2.2	Codage intra-bande . . . . .	72
3.1.2.3	Codage mixte . . . . .	74
3.1.3	Compression mixte . . . . .	74
3.1.4	Compression presque sans perte . . . . .	75
3.2	Études et standardisation pour la communication des images médicales . . . . .	76
3.2.1	Images bidimensionnelles . . . . .	77
3.2.1.1	Accès aléatoire . . . . .	78
3.2.1.2	Régions d'intérêt . . . . .	78
3.2.1.3	Progressivité . . . . .	79
3.2.2	Images volumiques . . . . .	79
3.2.2.1	Codage prédictif . . . . .	80
3.2.2.1.1	Prédiction de coupe . . . . .	81
3.2.2.1.2	Prédiction séquentielle . . . . .	81
3.2.2.1.3	Prédiction hiérarchique . . . . .	82
3.2.2.2	Codage par transformée . . . . .	83
3.2.2.3	Accès aléatoire . . . . .	84
3.2.2.4	Région d'intérêt et objets . . . . .	85
3.2.3	Standard DICOM . . . . .	86
	Conclusion . . . . .	87

## II Techniques avancées pour la compression d'images médicales 89

4	<b>Prédiction hiérarchique, orientée et adaptative</b>	<b>91</b>
	Introduction . . . . .	91
4.1	Prédiction hiérarchique . . . . .	92
4.1.1	Décomposition hiérarchique . . . . .	92
4.1.2	Prédicteurs hiérarchiques orientés . . . . .	94
4.1.2.1	Prédicteurs statiques . . . . .	94
4.1.2.2	Prédicteurs adaptatifs optimisés au sens des moindres carrés . . . . .	95
4.1.2.3	Seuil $T_{\text{Noise}}$ . . . . .	97
4.1.3	Correction des biais de prédiction . . . . .	97
4.1.3.1	Bref rappel . . . . .	97
4.1.3.2	Correction séquentielle de l'erreur contextuelle (SCEC) . . . . .	98
4.1.3.3	Validation de la SCEC . . . . .	99
4.1.4	Extensions presque sans perte . . . . .	101
4.1.4.1	IHINT presque sans perte . . . . .	101
4.1.4.2	HOP presque sans perte . . . . .	101
4.2	Compression . . . . .	102
4.2.1	Réorganisation des valeurs résiduelles . . . . .	102
4.2.2	Codage progressif en résolution . . . . .	102
4.2.3	Validation de l'approche pour la scalabilité en résolution . . . . .	104
4.2.4	Détails du codeur entropique . . . . .	104

4.3	Résultats . . . . .	109
4.3.1	Compression sans perte . . . . .	109
4.3.2	Compression presque sans perte . . . . .	110
	Conclusion et perspectives . . . . .	112
<b>5</b>	<b>Décomposition adaptative en paquets d'ondelettes</b>	<b>113</b>
	Introduction . . . . .	113
5.1	Base de décomposition . . . . .	114
5.1.1	Modèle de représentation de la structure de décomposition . . . . .	114
5.1.1.1	Représentation par arbre binaire . . . . .	114
5.1.1.2	Paramètres d'adaptation supplémentaires . . . . .	115
5.1.2	Familles d'ondelettes utilisées . . . . .	115
5.1.2.1	Filtres interpolatifs de Lanczos normalisés . . . . .	115
5.1.2.2	Filtres interpolatifs de Deslauriers et Dubuc . . . . .	116
5.1.2.3	Ondelettes construites par lifting . . . . .	116
5.2	Optimisation de la base pour la compression sans perte . . . . .	117
5.2.1	Sélection du filtre pour une étape de décomposition . . . . .	117
5.2.2	Construction de l'arbre de décomposition . . . . .	118
5.2.2.1	Décomposition d'une composante image . . . . .	118
5.2.2.2	Décomposition d'une bande fréquentielle . . . . .	118
5.3	Méthode de codage . . . . .	119
5.4	Résultats pour les images médicales volumiques . . . . .	120
5.4.1	Analyse en compression 2D . . . . .	121
5.4.2	Analyse en compression 3D . . . . .	127
5.5	Extensions . . . . .	127
5.5.1	Modèle prédictif pour la compression presque sans perte . . . . .	127
5.5.2	Compression d'une région d'intérêt uniquement . . . . .	132
5.5.3	Utilisation d'ondelettes orientées . . . . .	134
	Conclusion et perspectives . . . . .	134
<b>6</b>	<b>Compression scalable exploitant un apprentissage multi-bases, appli-</b>	
	<b>cation aux lames virtuelles</b>	<b>137</b>
	Introduction . . . . .	137
6.1	Base optimale pour la décorrélation . . . . .	138
6.1.1	KLT, PCA, POD, SVD . . . . .	138
6.1.2	Propriétés de la KLT . . . . .	138
6.2	Apprentissage multi-bases . . . . .	140
6.2.1	Motivations . . . . .	140
6.2.2	Algorithme d'apprentissage : construction itérative des KLT . . . . .	141
6.2.3	Note concernant l'implémentation . . . . .	142
6.3	Application en compression d'image dans un schéma scalable en qualité	143
6.3.1	Algorithme SPIHT . . . . .	144
6.3.2	Application dans le domaine direct (ADD) . . . . .	145
6.3.3	Application après une transformée en ondelettes . . . . .	145

6.3.3.1	Approche globale (AG)	145
6.3.3.2	Approche intra sous-bande (AISB)	148
6.3.3.3	Autres approches envisageables	152
6.3.4	Résultats pour compression irréversible d'images anatomopathologiques	155
6.3.4.1	Approche directe	155
6.3.4.2	Approche globale	157
6.3.4.3	Approche intra sous-bande	158
6.3.4.4	Comparaison des meilleurs résultats	158
6.4	Approximation des transformées pour une compression sans perte	160
6.4.1	Approximation réversible des KLT par factorisation PLUS	160
6.4.1.1	Choix de l'implémentation pour la transformée	163
6.4.1.2	Remarques sur la factorisation PLUS	164
6.4.2	Remarque concernant la normalisation réversible d'ondelettes bi-orthogonales	166
6.4.3	Taux de compression sans perte	166
6.4.4	Scalabilité en qualité pour une compression <i>lossy-to-lossless</i>	167
6.5	Améliorations des résultats	169
6.5.1	Déquantification	169
6.5.2	Codage arithmétique	171
6.5.3	Comparaisons expérimentales sur les patchs des lames virtuelles	173
6.5.3.1	Compression scalable irréversible des patchs de test	173
6.5.3.2	Compression scalable réversible des patchs de test	175
	Conclusion et perspectives	180

## **Conclusion et perspectives générales** **183**

<b>A Images</b>	<b>187</b>
A.1 TDM	187
A.2 IRM	190
A.3 Lames virtuelles	193
A.4 Patchs de lames virtuelles	196

## **Bibliographie** **216**

## **Publications** **217**

## **Table des figures** **222**



# Glossaire

ACR	Association Canadienne de Radiologie
ADL	Adaptive Directional Lifting
AT-SPIHT	Asymmetric Tree SPIHT
AVC	Advanced Video Coding
CALIC	Context-Based, Adaptive, Lossless Image Co-der
CAR	Canadian Association of Radiology
CCT	Continuous Cosine Transform
CDF	Cohen-Daubechies-Feauveau
CIPR	Center for Image Processing Research
CR	Computed Radiography
CT	Computed Tomography
DCT	Discret Cosine Transform
DICOM	Digital Imaging and Communications in Medicine
DPCM	Differential Pulse Code Modulation
DPWT	Discret Packet Wavelet Transform
DWT	Discret Wavelet Transform
EBCOT	Embedded Block Coding with Optimized Truncation
EZBC	Embedded Zero Block Coder
EZW	Embedded Zero-tree Wavelet
FMP	Fuzzy logic based Matching Pursuit
FOV	Field Of View
GAP	Gradient Adjusted Predictor
GIF	Graphic Interchange Format
HINT	Hierarchical INTerpolation
HIP	Hierarchical Interpolative Prediction

HOP	Hierarchical Oriented Prediction
HVS	Human Vision System
IRM	Image par Résonance Magnétique
ISA-DWT	Integer SA-DWT
IWT	Integer Wavelet Transform
JBIG	Joint Bi-Level Image Experts Group
JND	Just Noticeable Difference
JPEG	Joint Photographic Experts Group
KLT	Karhunen-Loeve Transform
LAR	Locally Adaptive Resolution
LBG	Linde-Buzo-Gray
LOCO-I	Low Complexity Lossless Compression for Images
LV	Lame Virtuelle
LZC	Layered Zero Coding
LZW	Lempel-Ziv-Welch
MDCT	Modified DCT
MeDEISA	Medical Database for the Evaluation of Image and Signal processing Algorithms
MLZC	Multi Layered Zero Coding
MPEG	Moving Picture Experts Group
MRI	Magnetic Resonance Image
MRP	Minimum-Rate Predictors
MS-SSIM	Multi-Scale SSIM
MSE	Mean Squared Error
NLM-VHP	the National Library of Medicine's Visible Human Project
ODC	Optimized Decomposition Coder
PACS	Picture Archiving and Communication System
PAE	Peak Absolute Error
PCA	Principal Component Analysis
PDF	Probability Density Function
PET	Positron Emission Tomography
PMR	Personal Medical Records

PNG	PNG's Not GIF
POD	Proper Orthogonal Decomposition
PSNR	Peak Signal to Noise Ratio
RDCT	Reversible DCT
RFC	Requests For Comments
RLE	Run Length Encoding
ROI	Region Of Interest
RONI	Region Of No Interest
ROPD	Rank Order Polynomial Decomposition
SA-DWT	Shape Adaptive DWT
SFR	Société Française de Radiologie
SPECK	Set Partitioned Embedded bloCK coder
SPIHT	Set Partitioning in Hierarchical Trees
SSIM	Structural SIMilarity
SVD	Singular Value Decomposition
TDAC	Time Domain Aliasing Cancellation
TDM	image TomoDensitoMétrique
VDP	Visible Differences Predictor
VLC	Variable Length Code
WBTC	Wavelet Block Tree Coder
WHT	Walsh-Hadamard Transform





# Introduction générale

L'imagerie médicale est désormais un support indispensable pour le dépistage, le diagnostic, la mise en place de protocoles de traitement, le suivi des patients, la préparation opératoire et le contrôle post-opératoire. Les avancées scientifiques et technologiques permettent de mettre en place de nouvelles modalités d'imagerie, souvent complémentaires aux existantes, mais aussi de progressivement améliorer leur précision. Il en résulte une progression, non seulement du nombre d'acquisitions pratiquées pour un même patient, mais également de la quantité d'information produite pour chaque examen. Ces images devant être conservées pour une certaine durée, l'espace de stockage requis pour l'archivage de toutes ces données est en constante évolution. Les images sont souvent consultées localement, mais peuvent également l'être à distance par le biais de réseaux ayant un bande passante limitée. L'utilisation de la compression se révèle donc rapidement être indispensable, que ce soit pour faciliter le stockage ou la navigation à distance dans ces masses de données.

Les résultats des travaux réalisés dans cette thèse sont principalement focalisés sur la compression des images médicales tomodynamométriques (TDM) et par résonance magnétique (IRM), mais également sur la compression d'une modalité émergente en imagerie biologique : les lames virtuelles (LV). Les deux premières sont actuellement les modalités d'acquisition les plus problématiques en terme de stockage, tandis que la dernière pourrait rapidement le devenir. En effet, les LV sont très volumineuses et, même si elles sont encore assez peu archivées, un besoin assez croissant se fait ressentir.

Pour les TDM et les IRM, la pratique courante est d'utiliser une compression réversible (sans perte), de sorte à garantir que l'image obtenue après la décompression soit identique à l'originale. Cependant, avec une telle contrainte, les taux de compression dépassent rarement 3:1, ce qui est assez peu, et pourront difficilement aller au-delà à cause des limites théoriques. L'emploi de la compression irréversible (avec pertes) permet d'obtenir des taux de compression supérieurs (20:1 par exemple), mais au détriment de la qualité de l'image restituée, et une telle pratique a encore beaucoup de mal à être acceptée par les radiologues qui veulent être certains de conserver l'intégrité de l'information diagnostique. Une solution intermédiaire peut alors être envisagée. Celle-ci consiste à n'introduire que des pertes contrôlées de sorte à garantir une certaine qualité de restitution des images aux médecins qui les exploitent pour leur diagnostic médical, tout en améliorant la compression en fonction de la marge d'erreur tolérée. Ce modèle de compression est souvent qualifié de presque sans perte.

Concernant les LV, certaines d'entre elles peuvent être conservées après une compression avec pertes, généralement dans des buts pédagogiques, ou après une compression sans perte, pour la recherche en cancérologie par exemple. Les images utilisées pour le diagnostic sont généralement détruites, le support physique pouvant être conservé et renumérisé si nécessaire.

Des fonctionnalités peuvent être introduites dans le modèle de compression dans le but d'accélérer la consultation à distance, notamment avec des schémas dits scalables ou progressifs. Ceux-ci permettent d'avoir accès à une ou plusieurs résolutions intermédiaires et/ou à un contenu de qualité variable en fonction de la portion du fichier compressé transmise, et ne nécessitent donc pas une étape supplémentaire de recompression de l'information pour envoyer un aperçu. Les images de qualités inférieures ainsi obtenues peuvent être suffisantes, si la pleine qualité n'est pas requise par l'observateur. Elles peuvent aussi être temporaires, dans le cas où le débit serait trop limité par exemple, et permettre de faire patienter l'utilisateur le temps que l'information complémentaire soit transférée. La scalabilité peut également servir à récupérer un ensemble d'images de qualité sous-optimale pour choisir uniquement celles pour lesquelles une qualité supérieure est nécessaire.

Les contributions de ce manuscrit cherchent à répondre efficacement à ces problèmes de scalabilité.

**Organisation du manuscrit** Ce manuscrit s'organise en deux parties. La première est relative au contexte de l'étude et la seconde présente principalement les contributions de cette thèse pour la compression des images médicales.

**Contexte de l'étude** Dans cette première partie, est approfondi, au **chapitre 1**, le contexte médical cadrant cette étude : ce chapitre détaille les trois modalités d'images traitées (TDM, IRM et LV), les contraintes liées au contexte médico-légal dans lequel s'intègre ce travail, leurs relations avec les différentes options d'archivage, et les différentes fonctionnalités pouvant être utiles en compression pour l'archivage et la consultation des images à distance. Au **chapitre 2** sont rappelés les principaux outils utiles à la compression de données, puis ceux plus spécifiques à la compression de signaux numériques et d'images qui sont employés dans les contributions de cette thèse et/ou dans les principaux algorithmes du **chapitre 3**. Ce dernier fait un état de l'art des principaux algorithmes de compression sans perte, presque sans perte, et/ou scalables pour les images fixes, puis présente l'essentiel des travaux plus spécifiques à l'imagerie médicale qui en sont dérivés.

**Contributions** Cette thèse propose principalement trois techniques de compression scalables pour les images médicales.

La première, décrite dans le **chapitre 4**, met en place une nouvelle méthode de prédiction hiérarchique, multi-orientée et adaptative. Celle-ci fusionne l'approche prédictive hiérarchique classique, qui s'appuie sur une prédiction par interpolation, avec le modèle de prédiction adaptatif de type DPCM qui offre généralement les meilleurs taux de compression sans perte et presque sans perte. Ce modèle prédictif est exploité de sorte à pouvoir non seulement effectuer une compression sans perte ou presque sans perte efficace, mais également proposer un modèle scalable en résolution. Cette nouvelle méthode de compression 2D est principalement développée pour la compression d'images bruitées telles que la grande majorité des coupes TDM et IRM natives.

Les principales contributions de cet algorithme sont un nouveau modèle de prédiction hiérarchique adaptatif, une nouvelle méthode pour la correction des biais de prédiction et un schéma de codage entropique adapté.

Le seconde algorithme de compression, présenté dans le **chapitre 5**, s'appuie sur un modèle de décomposition en paquets d'ondelettes. Pour chaque image, celui-ci est spécifiquement adapté afin d'améliorer leur compression sans perte. L'approche permet d'optimiser non seulement la décomposition en paquets, qui est plutôt caractéristique du contenu structurel de l'image, mais également le choix des atomes d'ondelettes à utiliser pour chacune des étapes de décomposition intermédiaires. Ce choix se fait au sein d'une famille d'ondelettes paramétrées par la taille de leur support de décorrélation. Il est ainsi possible de s'adapter à la nature plus ou moins douce/régulière du contenu sans devoir transmettre les coefficients des filtres utilisés : seul leur indice dans la famille (i.e. la taille de leur support) est requis. Deux familles d'ondelettes bi-orthogonales sont comparées. Elles sont issues d'un schéma de lifting en deux étapes qui s'appuie sur des filtres interpolatifs dont la taille du support est facilement paramétrable : soit les filtres de Lanczos, soit ceux de Deslauriers et Dubuc. L'utilisation des ondelettes est motivée par la possibilité de proposer une compression réversible scalable en qualité ou en résolution efficace, et pour être facilement utilisable sur des images 2D ou volumiques telles que les TDM ou les IRM. Le schéma de lifting est ensuite simplifié en supprimant le filtrage passe-bas afin d'obtenir une approche hiérarchique prédictive généralisée pour la décomposition en paquets et permet alors la mise en place d'un nouvel algorithme pour son exploitation dans un schéma de compression presque sans perte.

Les principales contributions de cette approche sont le modèle paramétrable de la décomposition en paquets d'ondelettes adaptatives, son optimisation et son extension presque sans perte.

Enfin, le contenu du **chapitre 6** permet d'effectuer une modélisation des spécificités d'une classe d'images dans le but d'optimiser leur représentation dans un schéma de compression scalable en qualité. Dans ce chapitre, cette optimisation est réalisée pour les LV qui ont des caractéristiques structurelles assez particulières.

La modélisation des spécificités du contenu s'effectue à l'aide de plusieurs bases orthonormées. Ce choix n'est pas anodin, il peut permettre, entre autres, d'approximer facilement la transformée par des étapes de lifting et rendre ainsi l'approche réversible. Afin d'adapter le modèle aux données, une optimisation est réalisée (hors ligne) pour

maximiser la compacité de l'énergie des vecteurs d'apprentissage après leur projection sur la base qui elle même maximise cette compacité. La base optimale pour l'ensemble des vecteurs qui lui sont associés n'est autre que leur KLT. L'optimisation de ces  $K$ -KLT peut alors être rendue itérative, avec un algorithme dont le principe est assez proche de celui des  $K$ -Means, pour faire converger l'apprentissage de manière certaine.

Ce modèle est appliqué classiquement dans le domaine image, pour des vecteurs formés par des blocs de pixels, ou après une transformée en ondelettes afin d'éviter les artefacts de type blocs et bénéficier de leur support spatial étendu. Pour les ondelettes, sont étudiées deux stratégies distinctes, ayant chacune leurs avantages et leurs inconvénients : la première s'appuie sur un apprentissage global, où les vecteurs à modéliser sont formés par des coefficients de l'ensemble des sous-bandes ; et la seconde considère les sous-bandes indépendamment et génère un modèle pour les blocs de coefficients de chacune d'entre elles. L'algorithme SPIHT est étendu pour chacune des approches afin de permettre une comparaison assez fiable de leurs performances pour la scalabilité en qualité. Les extensions de SPIHT sont optimisées par la suite en exploitant le codage arithmétique.

Les principales contributions de ce chapitre sont : l'optimisation du modèle par  $K$ -KLT, l'application d'un tel modèle dans le domaine ondelettes et les extensions de SPIHT pour le codage des différentes approches étudiées.

Même si les algorithmes ont initialement été développés pour des images médicales, ils ne leur sont pas spécifiques et pourraient très bien être appliqués à des images satellitaires ou naturelles par exemple.

**Conclusion et perspectives** Le manuscrit s'achève sur une conclusion générale concernant les différents travaux menés au cours de cette thèse, et sur les perspectives pouvant en découler.

Première partie

Contexte de l'étude



# Chapitre 1

## L'imagerie médicale : quelques éléments de contexte

### Introduction

Ces dernières décennies, l'imagerie médicale connaît une évolution spectaculaire aussi bien au niveau du développement des techniques de production qu'au niveau de leur utilisation. Aujourd'hui, ces nouvelles technologies s'avèrent indispensables pour les diagnostics et leur usage intensif pose des problèmes de stockage et de transmission.

Ce chapitre présente d'abord (sec. 1.1) les trois modalités et techniques d'acquisitions des images médicales ciblées pour cette thèse, ainsi qu'un descriptif des bases d'images utilisées pour les expérimentations. Le contexte dans lequel s'insère ce travail est ensuite décrit (sec. 1.2) afin de mettre en évidence les contraintes et la nécessité d'une compression efficace et adaptée à l'utilisation de ces images volumineuses. Suit (sec. 1.3) une introduction à différentes fonctionnalités pouvant être mis en place afin de faciliter la manipulation des clichés compressés.

Pour toute information complémentaire à cette introduction à l'imagerie médicale, le lecteur pourra également se référer à [NACM07, NACM08] et en particulier aux chapitres [TJLBF08] et [CMNAT<sup>+</sup>08], respectivement consacrés à l'intérêt de la compression pour les images médicales et à la présentation de diverses modalités d'acquisition.

### 1.1 Modalités d'images ciblées

Trois modalités d'images médicales, pour lesquelles la compression semble être d'un intérêt économique important à court ou moyen terme, seront abordées dans ce document. Les deux premières sont des modalités volumiques : l'imagerie tomographique et l'imagerie par résonance magnétique. La troisième est une modalité bidimensionnelle de très grande taille : l'imagerie anatomo-pathologique.



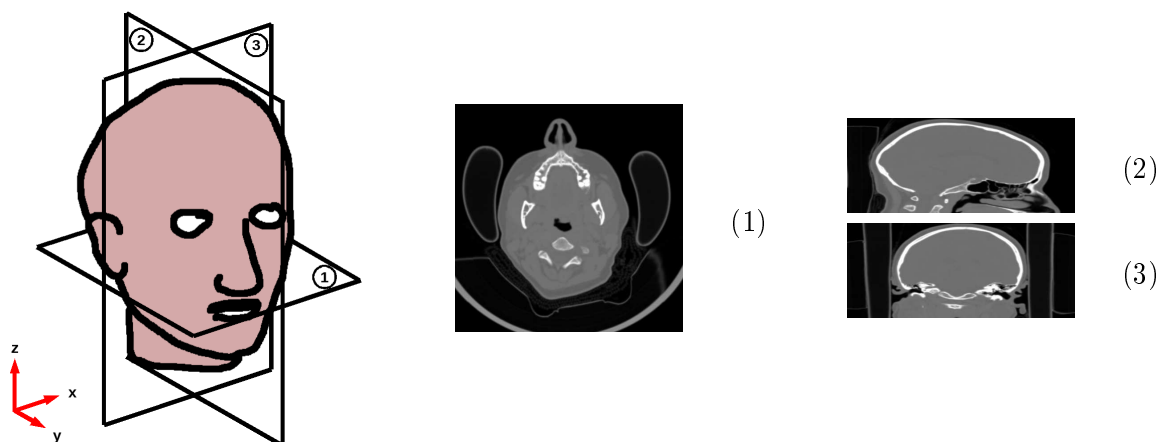


FIG. 1.1 – Représentations : transverse (1), sagittale (2), coronale (3)

### 1.1.1 Images médicales volumiques

Hormis la radiographie, les technologies les plus fréquemment utilisées en milieu hospitalier permettent d'acquérir des images volumiques. Ces volumes sont organisés comme une succession d'images bidimensionnelles (coupes) prises à distance régulière le long d'un axe transversal. Le pixel d'une image appartenant à une telle séquence correspond à la quantification d'une énergie présente dans un petit volume, c'est pourquoi il est couramment appelé voxel.

La visualisation coupe à coupe du volume peut se faire le long de chacun des 3 axes d'acquisition. Le plus souvent, les coupes sont successivement obtenues transversalement à l'axe tête/pieds du patient qui est allongé sur une table d'examen. Une telle organisation des images est alors dite « représentation transverse » ou « transaxiale ». Une « représentation sagittale » correspond à une organisation des voxels pour former des coupes perpendiculaires à l'axe épaule/épaule. Enfin une « représentation coronale » est triée le long de l'axe dos/ventre (cf. FIG. 1.1).

En se plaçant dans le repère d'acquisition original, la résolution spatiale inter-coupe (axe  $z$ ) n'est pas nécessairement la même que la résolution spatiale interne à la coupe (axes  $x,y$ ). Ainsi, le volume d'un voxel n'est pas toujours cubique. Il peut également y avoir un recouvrement entre coupes successives, lorsque leur épaisseur est plus importante que le pas d'acquisition.

Pour le milieu médical, les deux modalités d'images volumiques les plus coûteuses en terme d'espace de stockage sont la tomodensitométrie (TDM), dite aussi tomographie axiale calculée par ordinateur, CT-scan (*Computed Tomography*) ou plus simplement scanner ; et l'Imagerie par Résonance Magnétique (IRM).

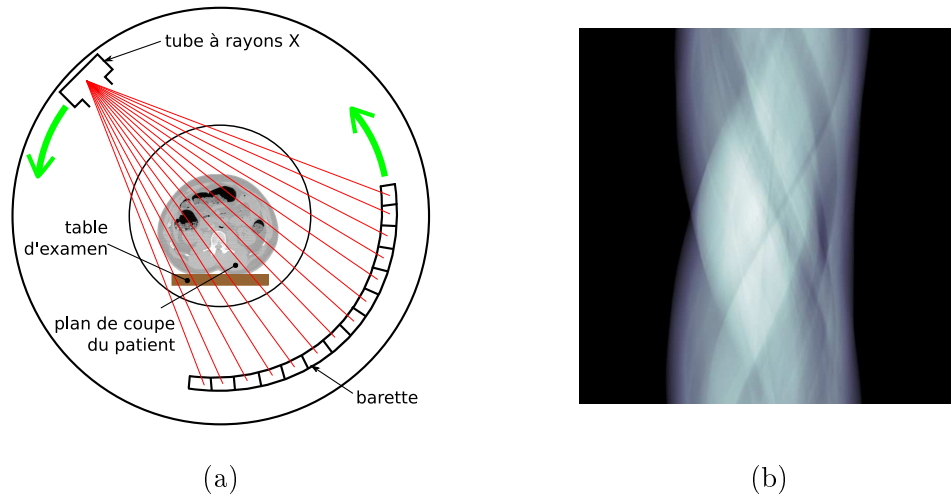


FIG. 1.2 – Acquisition tomodensitométrique (a) et sinogramme (b).

#### 1.1.1.1 TDM

Un tube à rayons X et un capteur (barrette) sont situés de part et d'autre de l'objet à observer (patient allongé sur la table d'examen). Le tube, qui émet de minces faisceaux de rayons X, et la barrette, qui permet de quantifier la quantité de rayons X absorbés selon une orientation précise, subissent une rotation autour du plan à capturer (plan de coupe) (FIG. 1.2-a). L'acquisition des faisceaux ainsi effectuée selon diverses orientations permet d'obtenir le *sinogramme des projections* (FIG. 1.2-b), qui sert ensuite à calculer numériquement une image de la coupe (FIG. 1.3).

Plusieurs générations de matériels d'acquisition ont vu le jour : les plus anciens prenaient les coupes une à une, la table avançant par pas et s'arrêtant pour chaque prise. Dans les années 90 un tel examen pouvait prendre un peu moins d'une heure, et des artefacts importants dus aux mouvements (respiration par exemple) du patient pouvaient apparaître. Cette technique a été quasiment abandonnée. Le scanner hélicoïdale, quant à lui, fait avancer la table de manière constante, les faisceaux de rayons X décrivent alors une hélice autour du patient et l'acquisition se fait de façon continue. Les coupes acquises ainsi sont souvent assez épaisses et les représentations coronales et sagittales souffrent d'un manque de précision. Enfin, les techniques actuelles permettent de prendre jusqu'à 128 (et plus) coupes simultanément à l'aide de plusieurs rangées de détecteurs (scanners multibarrettes). Le tube à rayons X effectue une rotation complète autour du patient en un temps inférieur à la demi-seconde pour le matériel le plus récent. L'épaisseur des coupes peut varier entre 0.5mm et 5mm et atteindre 10mm sur un matériel plus ancien. Certains scanners permettent également de choisir parmi une gamme d'épaisseurs, ainsi qu'entre une acquisition axiale ou hélicoïdale.

A titre d'exemple, citons un matériel relativement récent, inauguré fin 2008, au centre hospitalier Duchenne à Boulogne, le *Somatom Définition AS+* de la société Sie-



FIG. 1.3 – Exemple de coupe TDM

mens qui comporte 128 barrettes permet d'obtenir l'acquisition d'un corps entier en 4 secondes. Il effectue une rotation complète en 0.3 seconde, a une résolution spatiale de 0.33mm, une épaisseur de coupe de 5mm et une distance inter-coupe de 3mm.

Le sinogramme des projections est organisé de sorte à fournir une représentation similaire à l'espace des projections de Radon :

$$p(u, \theta) = \int_{-\infty}^{\infty} f(t \sin \theta + u \cos \theta, -t \cos \theta + u \sin \theta) dt,$$

dont l'inverse est :

$$f(x, y) = \int_0^{\pi} p(x \cos \theta + y \sin \theta, \theta) d\theta.$$

La discrétisation de  $u$  et  $\theta$ , pouvant varier selon le matériel, compromet la reconstruction parfaite de l'image. Un certain nombre de traitements doivent alors être effectués afin d'améliorer le rendu.

La méthode d'inversion la plus répandue dans les dispositifs d'acquisition est la rétroprojection filtrée FBP (*Filtered BackProjection*) : les coefficients de chacune des orientations  $\theta$  sont tout d'abord filtrés avant d'être rétroprojetés à l'aide d'un algorithme itératif. Cette technique de reconstruction repose sur le théorème de la coupe centrale

CST (*Central Slice Theorem*) qui met en évidence la relation entre la transformée de Fourier de  $p(u, \theta)$  selon  $u$ , et la transformée de Fourier de l'image dans un espace polaire. Ainsi, l'espace de Fourier de l'image acquise à l'aide d'un sinogramme est discrétisée selon une grille polaire et l'énergie se trouve concentrée autour des basses fréquences. De façon théorique (lorsque le pas d'échantillonnage tend vers 0), pour que la rétroprojection soit une inversion exacte de la transformée, la représentation dans l'espace de Radon doit d'abord être filtrée par un filtre rampe (qui permet de normaliser l'énergie).

Dans la pratique, la représentation discrétisée filtrée permet seulement de reconstruire une approximation de l'image originale et génère des artefacts. Ces artefacts, bien que réduits par le filtrage, sont tout de même visibles : des halos de formes étoilées ou des droites diffusant l'énergie des zones où celle-ci est concentrée de manière importante peuvent apparaître. Il est à noter que l'acquisition des sinogrammes souffre (comme en radiographie) d'un bruit poissonnien qui perturbe également la reconstruction. Afin de réduire les artefacts, il est possible d'utiliser d'autres filtres que le filtre rampe. Cependant, l'image générée aura tendance à subir d'autres distorsions (être plus floue par exemple). Bertram et al. [BRS<sup>+</sup>04] proposent également un algorithme d'interpolation du sinogramme avant la reconstruction afin d'augmenter la précision dans l'espace de Radon. Des techniques similaires peuvent également permettre de générer des images volumiques avec une résolution plus importante que celle d'acquisition.

Il existe diverses autres techniques d'inversion du sinogramme qui ne provoquent pas forcément les mêmes artefacts. Elle peuvent être catégorisées en deux familles : les méthodes algébriques qui résolvent un système d'équations linéaires (ART, SIRT, ILST, méthode du gradient conjugué, etc.) [GBH70, Gil72, HR73, HLL75] et qui ignorent la présence du bruit dans les données contrairement aux méthodes statistiques (MLEM, OSEM, RAMLA, DRAMA, etc.) [SV82, LBL87, HL94, BdP96, TK03].

Des techniques de reconstruction ont été étendues pour utiliser l'information tridimensionnelle (sinogrammes de plusieurs coupes) et prendre en considération le positionnement (géométrie) des capteurs. Ceci permet de réduire le bruit ainsi que les artefacts de reconstruction. Parmi les approches volumiques, on peut citer la généralisation de la rétroprojection filtrée (3D FBP).

### 1.1.1.2 IRM

L'IRM utilise les propriétés magnétiques quantiques (spin) de certains atomes comme l'hydrogène, qui est très présent dans les tissus biologiques au travers des molécules telles que l'eau. Des ondes magnétiques oscillantes sont appliquées à ces atomes, qui, déjà soumis à un fort champ magnétique constant  $\vec{B}$ , vont entrer en résonance magnétique (écho de spin). Ce phénomène survient à une fréquence particulière  $\omega$  : la fréquence de Larmor ou fréquence de résonance, proportionnelle à  $B$  (et différente selon les atomes), et se traduit par une modification de l'aimantation du noyau des atomes (protons). Celle-ci va effectuer un mouvement de rotation perpendiculaire au champ  $\vec{B}$  avec une fréquence  $\omega$  : la précession. Après l'arrêt de la fréquence excitatrice, l'aimantation continue d'osciller et va progressivement se restabiliser pour suivre le champ  $B$ . La résonance magnétique



FIG. 1.4 – Exemple de coupe IRM

émise durant cette phase de stabilisation est captée à l'aide d'antennes et génère un signal électrique d'intensité proportionnelle à la quantité de protons en résonance et oscille donc également à la fréquence  $\omega$ . Pour pouvoir mesurer localement la concentration d'atomes, le champ magnétique  $\vec{B}$  est appliqué de manière variable dans l'espace (gradient), afin de générer une fréquence de résonance différente en chacun des points à étudier. Pour obtenir des images avec un meilleur contraste, on mesure généralement le temps nécessaire aux atomes pour revenir à l'équilibre longitudinalement (relaxation T1) et transversalement (relaxation T2). Les fréquences ainsi captées permettent de former une image dans un espace couramment appelé *espace K*. Cette représentation offre un codage de la phase et de l'amplitude pour diverses fréquences et orientations et est organisée de telle sorte qu'elle forme un espace de Fourier discrétisé. Elle est ainsi facilement inversible à l'aide d'algorithmes de transformée de Fourier rapides et permet de générer l'image de la localisation des sources de résonance (FIG. 1.4).

Sur un principe similaire à l'écho de spin, on trouve également l'écho de gradient, qui consiste aussi à donner un mouvement de rotation à l'aimantation du noyau, mais avec un angle plus faible ( $< 90^\circ$ ,  $90^\circ$  correspondant à l'écho de spin), qui permet un temps d'excitation également plus faible et une stabilisation plus rapide après arrêt de l'excitation. Ainsi, l'écho de gradient offre un gain de temps qui permet de remplir plus

rapidement un espace  $K$  tridimensionnel, offrant ainsi la possibilité de construire des IRM de façon volumique directement (IRM3D). Les coupes d'un tel volume peuvent être plus fines car leur rapport signal à bruit est meilleur, et sont reconstruites à l'aide d'une transformée de Fourier 3D inverse.

Plus le champ magnétique constant est élevé, plus le rapport signal sur bruit tend à être meilleur. Pour l'imagerie médicale, il est souvent compris entre 0.1 et 3 Tesla et peut parfois dépasser 11 Tesla. On distingue trois types d'aimants : permanents, résistifs et supraconducteurs. Les premiers produisent un champ magnétique permanent sans consommer d'énergie. Ils sont donc très fiables et tendent à se développer. Ils sont cependant très lourds (plusieurs tonnes) et dépassent rarement 0.4 Tesla. L'aimant résistif se compose d'une simple bobine de cuivre qui génère un champ magnétique lorsqu'elle est traversée par un courant électrique. Bien que peu coûteux à la fabrication, il consomme beaucoup d'énergie, souffre de l'effet Joule et génère un champ magnétique peu stable qui atteint difficilement 0.5 Tesla. C'est pourquoi il est peu utilisé depuis l'apparition des aimants supraconducteurs. Ces derniers se composent d'une bobine en Niobium-Titane (Nb-Ti) baignée constamment dans de l'hélium liquide afin de réduire sa résistivité. Ils permettent ainsi de générer de forts champs magnétiques. Ces appareils sont très coûteux à l'achat comme à l'utilisation à cause de leur consommation importante en électricité et en hélium liquide.

Il existe également deux catégories de systèmes d'acquisition : l'une dite fermée et l'autre dite ouverte. Les systèmes fermés sont les plus répandus. Ils se présentent sous la forme d'un tunnel autour duquel on trouve l'aimant, les bobines de gradient de champ magnétique et les antennes émettrices/réceptrices générant ou captant les fréquences de résonance. Les systèmes ouverts ont des configurations variables en fonction de la forme de l'aimant utilisé (« fer à cheval » par exemple). Ces derniers génèrent des champs magnétiques moins importants, mais permettent plus facilement de faire passer des IRM à des personnes claustrophobes, enceintes ou encore obèses. Les derniers modèles arrivent tout de même à dépasser les 1.0 Tesla.

Les techniques d'acquisition les plus rapides permettent de remplir l'espace  $K$  en moins d'une seconde mais sont moins précises (plus floues et plus bruitées). Ainsi la durée d'acquisition pour un examen est en général d'une dizaine de minutes. On peut actuellement obtenir des résolutions proches du demi-millimètre pour des épaisseurs de coupes pouvant varier entre 2 et 5 mm. Ces coupes épaisses sont nécessaires pour obtenir un rapport signal à bruit intéressant.

La majorité des artefacts visibles sont liés aux propriétés physiques de la technique d'acquisition, et non à la méthode de reconstruction. La présence d'autres atomes entrant en résonance ou l'orientation des tissus selon un certain angle (« *magic angle* ») génèrent une énergie qui peut perturber l'interprétation des images. De faibles mouvements du patient durant l'acquisition génèrent également des artefacts. Enfin, des oscillations peuvent apparaître aux abords des contours très marqués ; c'est le phénomène de Gibbs.

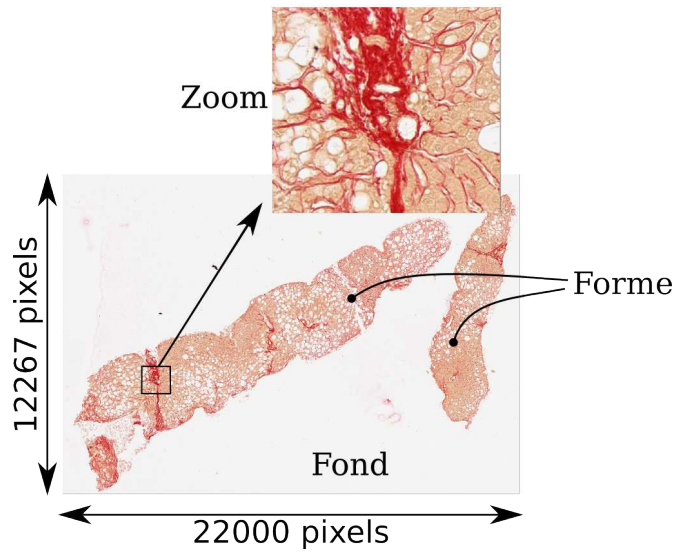


FIG. 1.5 – Exemple de lame virtuelle

Pour obtenir des informations plus détaillées, un cours interactif très bien conçu, et récompensé par plusieurs organismes de radiologie, est disponible en anglais [HMa] et en français [HMb]. On peut également citer [BBM<sup>+</sup>04] qui passe en revue des techniques récentes pour accélérer l'acquisition (IRM parallèle).

### 1.1.2 Lames virtuelles (LV)

En médecine, l'étude des tissus biologiques (histologie) s'effectue généralement au microscope après biopsie (prélèvement d'un petit échantillon de tissu). Elle est pratiquée en anatomo-pathologie qui étudie les lésions macro et microscopique (histopathologie) pour le diagnostic.

Traditionnellement, le tissu est préparé et fixé sur lame de verre, avant d'être observé « manuellement » au microscope, puis conservé physiquement. L'inconvénient majeure étant que l'échantillon risque de s'abîmer avec le temps : les marqueurs (fluorescent, coloré, ...) choisis en fonction du problème diagnostique peuvent parfois s'altérer, ou les tissus mal se conserver. En revanche, s'il est numérisé son archivage sous forme physique n'est plus spécialement nécessaire, et sur un point de vu légal, l'image conservée reste la même que celle utilisée lors du diagnostic.

Ainsi, une lame virtuelle [CBC<sup>+</sup>03, RGI07, TJLBF08] (FIG. 1.5) est une image de très grande taille qui essaie de fournir la même qualité des détails que lors d'une observation traditionnelle au microscope. Elle est obtenue à l'aide d'un objectif de grossissement (généralement  $\times 20$  ou  $\times 40$ ), qui, couplé à un capteur CCD, permet d'obtenir une très haute définition (inférieure au micron par pixel). L'acquisition d'une image (en  $752 \times 560$  par exemple) du « champs de vision » (FOV : *fields of view* ou *tile*), qui

correspond à ce que verrait un histologiste derrière son microscope, peut être faite sur trois canaux (Rouge, Vert, Bleu) ou plus en imagerie multi-spectrale, avec une précision de 8 ou 12 bits par canal. Afin de numériser l'échantillon, la totalité de la zone d'acquisition est balayée séquentiellement afin de quadriller l'espace par des prises locales de FOV qui sont ensuite assemblées pour former la lame virtuelle. Avant chaque prise de FOV le déplacement de l'échantillon devrait s'effectuer avec une précision extrême afin que toutes soient correctement alignées. Pour contourner ce problème et rendre les systèmes moins coûteux, les FOV peuvent être acquises de manière à se recouvrir légèrement et des algorithmes de recalage peuvent être appliqués pour permettre de réaligner correctement chaque FOV par rapport à ses voisins. D'autres contraintes, comme des légers changements d'illumination, peuvent également intervenir et doivent être prises en compte lors de l'assemblage.

Les images ainsi obtenues deviennent rapidement volumineuses : un grossissement  $\times 20$  de  $1\text{cm}^2$  peut produire environ  $20000 \times 20000$  pixels, soit, pour 3 canaux RGB échantillonnés à 8 bits par canal, un volume numérique de 1.2 GB par  $\text{cm}^2$  de lame virtuelle.

### 1.1.3 Bases d'images utilisées pour les expérimentations

#### 1.1.3.1 Images volumiques

Afin de permettre de comparer différents algorithmes, une sélection de bases d'images volumiques (CT et IRM) natives et prétraitées a été retenue à partir de plusieurs sources :

PMR – Ces images proviennent d'un dossier médical patient relativement récent contenant notamment des TDM et des IRM moins bruitées que dans les autres bases (TDM construites de manières volumiques, IRM et IRM-3D en haute résolution ( $512 \times 512$ )).

MeDEISA – Cette base est destinée à évaluer différents algorithmes de traitements d'images médicales : MeDEISA<sup>1</sup> (*Medical Database for the Evaluation of Image and Signal Processing Algorithms*). Elle se compose d'images tomographiques cardiaques post-traitées (une partie du bruit a été supprimée par filtrage convolutif), d'IRM et d'IRM-3D natives et post-traitées. Celles post-traitées ont des portions du bruit d'arrière plan (sans intérêt diagnostique) supprimées (valeurs mises à zéro).

VHP-Male/Female/Harvard – Ces trois ensembles contiennent des images en provenance directe du matériel d'acquisition : ils sont issus de la base NLM-VHP<sup>2</sup> (*The National Library of Medicine's Visible Human Project*), et comprennent deux corps complets (un Homme, une Femme) et une tête (Harvard), disponibles sous forme d'IRM, de scanners, et de photographies couleurs de cryosections (ces dernières n'ont pas été utilisées). VHP-Harvard contient une IRM-3D haute résolution probablement générée (par le matériel) par compressive sensing. Les résultats

---

<sup>1</sup><http://www.medeisa.net>

<sup>2</sup>[http://www.nlm.nih.gov/research/visible/visible\\_human.html](http://www.nlm.nih.gov/research/visible/visible_human.html)



pour cette IRM sont très différents de ceux pour les autres volumes. Elle sera donc considérée indépendamment (VHP-Harvard-3D).

CIPR – Les TDM et IRM contenues dans cette base sont requantifiées sur 8 bits. Elles proviennent du Mallinckrodt Institute of Radiology, Image Proc. Lab. et sont disponibles sur le site du CIPR<sup>3</sup> (*Center for Image Processing Research*). Bien que la requantification sur 8 bits ait un impact important sur la compression (suppression d'une grosse partie du bruit), cette base d'images est souvent utilisée comme référence en compression d'images volumiques.

Exceptée la base CIPR, toutes les images sont sur 12 bits. Sur l'ensemble des bases, toutes les valeurs négatives ont été mises à zéro. Celles-ci apparaissent parfois sur certaines tomographies (cas de MeDEISA) pour signaler les zones de l'image ne contenant aucune information reconstruite (extérieur de la zone de reconstruction des TDM) et n'ont donc aucune incidence diagnostique.

Le lecteur intéressé pourra également trouver une base d'images<sup>4</sup> principalement destinée à l'évaluation des performances du logiciel OSIRIX<sup>5</sup>, comportant environ 20 giga octets de données. Ces images ont pour la plupart subi des post-traitements (images non natives).

Cette sélection permet d'utiliser une bonne variété d'images volumiques pour évaluer l'efficacité de diverses techniques de compression sur les TDM et les IRM. : des images avec une résolution axiale fine (majorité des TDM et IRM-3D de VHP-Harvard) qui peuvent contenir des artefacts de reconstruction/bruits inter-coupe corrélés ou non (filtrage et/ou techniques de reconstruction différentes), d'autres avec une résolution axiale plus faible, possédant également un bruit corrélé ou non : IRM et IRM-3D natives. Les propriétés intra-coupes sont également diversifiées avec des images peu et très bruitées, et des IRM basse ( $256 \times 256$ ) et haute ( $512 \times 512$ ) résolutions. La TAB. 1.1 donne quelques détails sur les bases utilisées, et des images d'exemple sont données en annexe : A.1 pour les TDM et A.2 pour les IRM.

Les images post-filtrées et celles reconstruites à l'aide d'algorithmes volumiques sont donc moins bruitées que les autres images, et leur contenu est plus corrélé dans l'espace et donc visuellement plus doux, avec plus de régularité. Dans la suite du manuscrit, elles seront donc qualifiées d'images « douces » ou « régulières ».

### 1.1.3.2 Lames virtuelles

Pour les images anatomopathologiques, les expérimentations ont porté sur une base de 30 lames virtuelles<sup>6</sup>. Ces images sont accompagnées d'une segmentation en quatre classes (fond, tissu sain, stéatose et fibrose) réalisée automatiquement<sup>6</sup>. Seule la différenciation fond/forme (ROI) est utilisée dans ces travaux. La TAB. 1.2 précise quelques

<sup>3</sup><http://www.cipr.rpi.edu/resource/sequences/sequence01.html>

<sup>4</sup><http://pubimage.hcuge.ch:8080>

<sup>5</sup><http://www.osirix-viewer.com>

<sup>6</sup>Nous tenons à remercier Christine Cavaro-Ménard du Laboratoire d'Ingénierie des Systèmes Automatisés d'Angers pour nous avoir fourni ces données.

Bases	TDM			IRM		
	volumes	coupes	X,Y res.	volumes	coupes	X,Y res.
CIPR (8 bits)	4	557	256,256	4	251	256,256
MeDEISA	5	1050	512,512	8	777	256,256
				2	58	512,512
PMR	2	426	512,512	5	188	512,512
VHP-Female	1	1734	512,512	18	923	256,256
VHP-Male	2	2395	512,512	18	973	256,256
VHP-Harvard	1	463	512,512	3	228	256,256
VHP-Harvard-3D	-	-	-	1	229	512,512

TAB. 1.1 – Composition des bases d'images retenues

informations quantitatives sur les données utilisées, et quelques exemples sont proposés en annexe : A.3.

Images	Dimensions moyennes	Taille moyenne image	Taille moyenne ROI
30	$\approx 19000 \times 17000$	324 Mpixels	61 Mpixels

TAB. 1.2 – Informations quantitatives relatives à la base de lames virtuelles utilisée.

Pour des raisons pratiques (coût mémoire et rapidité d'exécution des algorithmes), quatre patches représentatifs de dimensions  $1024 \times 1024$  ont également été extraits de la région d'intérêt de chacune des 30 lames et convertis en images monochromatiques de luminance (8 bpp). Ces patches seront référencés par XX-YY, où XX correspond à l'indice de la lame et YY à celui du patch. Quelques patches sont mis en annexe : A.4.

## 1.2 Contexte médico-légal et options d'archivage

### 1.2.1 Contraintes quantitatives

Pour des raisons médicales (suivi des patients), juridiques (expertises en cas de litige) et afin d'éviter des examens redondants, les clichés médicaux doivent être archivés à plus ou moins long terme. La loi française prévoit une conservation des données relatives à un patient durant au minimum 20 ans, même après son dernier contact hospitalier. Pour les problèmes de nature héréditaire, la conservation est illimitée dans le temps. En cas de procès, les informations doivent rester disponibles jusqu'au règlement définitif du dossier. De plus, « la jurisprudence a tendance à considérer que l'impossibilité pour le médecin de fournir les documents est un argument à charge » (SFR : Société Française de Radiologie)<sup>7</sup>. Il est donc plutôt conseillé de conserver les informations sur 30 ans au lieu de 20. Bien que les durées d'archivage puisse varier d'un pays à l'autre, la trentaine

<sup>7</sup><http://www.sfrnet.org/sfr/grandpublic/2-dossier-informatise/problematique/index.phtml>

d'années est assez courante en Europe. Ces longues périodes de conservation ajoutées à une constante évolution de l'imagerie médicale posent de sérieux problèmes de stockage et de transfert.

La tomographie est devenue très populaire durant les dernières décennies et son usage s'est beaucoup intensifié. La quantité d'images ainsi produites chaque année a explosé de manière quasi exponentielle. L'IRM, quant à elle, offre souvent une résolution transversale ( $z$ ) plus faible que la tomographie et son coût et sa durée d'acquisition rendent sa fréquence d'utilisation moins élevée. Bien qu'elle soit ainsi moins gourmande en espace de stockage, elle reste la troisième modalité la plus encombrante après les radiographies et les scanners d'après des études localement menées au sein de centres d'archivage, communément appelés PACS (*Picture Archiving and Communication System*) [LLK<sup>+</sup>05, OBO05, OBBO06]. Pour ces deux modalités, les évolutions technologiques ont également conduit à l'augmentation de la résolution ( $x,y,z$ ) et de la précision d'acquisition (bits par pixels) des appareils. De ce fait, les images deviennent de plus en plus volumineuses.

Une étude menée dans le PACS du département de radiologie du centre médical universitaire de Groningen aux Pays-Bas [OBO05] montre que leur production de scanners a évolué de 19875 coupes par mois en moyenne pour l'année 2000 à 552773 pour l'année 2004, tandis que celle des IRM s'est plus faiblement accrue de 66315 à 104457. Ainsi en 2004 plus de 3 TB ont été produits par les scanners et 657 GB par les IRM, pour un total de plus de 5 TB produits, toutes modalités d'images confondues. En Janvier 2006, leurs prévisions de production étaient de quasiment 16 TB pour l'année 2010 alors qu'elles n'étaient que de 13 TB 7 mois plus tôt [OBBO06].

En ce qui concerne les lames virtuelles, les images non compressées atteignent des tailles allant 0.5 GB et pouvant dépasser les 10 GB, et comme mentionné dans plusieurs publications ([CBC<sup>+</sup>03, LLK<sup>+</sup>05] par exemple), des établissements spécialisés tels que l'hôpital John Hopkins (Maryland) peuvent traiter plusieurs centaines de milliers de lames histologiques par an. Il n'est donc, a priori, pas envisageable de conserver de telles images pendant 20 années (par exemple, la quantité de données à stocker représenterait environ  $10^7$  TB = 10 EB pour les USA et environ  $5 \cdot 10^4$  TB = 50 PB pour John Hopkins seulement). Vu la taille des images et les quantités potentielles même pour du stockage à moyen terme (quelques semaines ou quelques mois) il faudrait des algorithmes de compression très efficaces, et inévitablement avec pertes. C'est pourquoi ce sont majoritairement les supports physiques qui sont conservés. Si le stockage à long terme de la totalité des images n'est actuellement pas envisageable, seules des portions contenant les informations relatives au diagnostic pourraient cependant être archivées. Dans ce cas, un modèle attractif serait une représentation multi-résolution dont seules les zones géographiques intéressantes seraient conservées en bonne qualité, et ceci en fonction de la résolution nécessaire. Ce modèle pourrait être généré de manière interactive pour permettre de recréer le parcours effectué par le spécialiste lors de son diagnostic, et éventuellement couplé à une système d'*eye tracking* pour permettre d'analyser les comportements lors de l'observation de ces images, et d'y adapter la

compression.

Actuellement, des lames virtuelles sont parfois conservées dans des buts pédagogiques, généralement après une compression avec pertes, ou dans des buts de recherches scientifiques, de préférence après une compression sans perte.

Bien qu'il existe une augmentation « perpétuelle » des capacités de stockage et des débits grâce aux avancées scientifiques, les coûts matériels pour le transfert et l'archivage deviennent faramineux. Les réseaux internes des hôpitaux peuvent s'encombrer si il y a trop de consultations simultanées d'images volumineuses. Les accès intra-hospitalier, mais encore plus les accès extra-hospitaliers (télémedecine) qui s'effectuent le plus souvent sur des réseaux à capacités réduites, peuvent bénéficier d'approches visant à faciliter la navigation en réduisant la quantité d'information à transmettre à un instant donné, et en se focalisant sur celle qui est vraiment nécessaire pour pouvoir la faire parvenir en priorité. L'enjeu de l'utilisation d'algorithmes de compression efficaces pour les images médicales en vue de leur stockage et/ou de leur transmission est donc important.

### 1.2.2 Archivage sans perte

Le plus souvent, pour satisfaire des raisons éthiques liées au diagnostic, les données des images médicales sont archivées soit de manière brute (sans compression) soit après une compression sans perte. Ceci permet de conserver une copie de l'image identique à l'originale.

Le stockage brut consiste simplement à enregistrer les valeurs des voxels selon un ordre prédéfini (généralement coupe par coupe, ligne par ligne et colonne par colonne). Ce mode de stockage utilise un nombre de bits par voxel multiple de 8 (un octet) de manière à faciliter les manipulations. Ainsi des scanners en niveaux de gris sur 12 bits seront stockés sur 16 bits (2 octets) par voxel, ce qui rend les données encore plus volumineuses.

La compression sans perte, quant à elle, peut être effectuée à l'aide d'algorithmes généralistes de compression de données. Les formats tels que ZIP, RAR, GZ, BZ2, 7Z ou encore PAQ8 sont assez communs et aisés à mettre en place puisqu'ils permettent de s'astreindre facilement des diverses organisations numériques (données sur plus de 8 bits : little endian, big endian ; formats de fichiers ; ...). Cependant, même s'ils permettent facilement (en terme d'ingénierie à mettre en place) d'obtenir un gain d'espace de stockage, les algorithmes les plus performants, tels que PAQ8 ou LZMA (format 7Z) sont également les plus complexes et restent bien moins efficaces que des algorithmes spécifiques aux images : les taux de compression sont souvent moins bons pour des temps de calcul pouvant être beaucoup plus importants. Les algorithmes de compression d'images sans perte sont donc bien souvent préférés. Cependant, bien que plus performants, ces systèmes offrent encore des taux de compression peu satisfaisants. Ceux-ci peuvent être inférieurs à 2:1 et dépassant rarement 6:1, aussi bien avec des algorithmes bidimensionnels qu'avec des algorithmes volumiques qui sont encore plus efficaces.

### 1.2.3 Contraintes qualitatives

Les performances de la compression sans perte sont très faibles en comparaison à des systèmes supprimant une partie de l'information et dits avec pertes. Cependant un second aspect juridique contraignant porte sur la qualité des images archivées. En effet, afin d'éviter tout litige au cours d'une expertise en cas de poursuite judiciaire, les images conservées doivent être strictement les mêmes que celles utilisées lors du diagnostic. Ainsi, si des pertes doivent être introduites lors de la compression, celles-ci doivent apparaître en amont de la présentation des clichés au radiologue (afin d'éviter l'altération des preuves).

### 1.2.4 Archivage avec pertes

Même si l'archivage avec pertes est encore controversé, la plupart des contributions récentes, dans le domaine de la compression d'images médicales, tentent d'investir le cadre de la compression irréversible [NCOC11, GM09, Gau06, MC04, LLF03, WT01], et les travaux les plus poussées essaient de quantifier la perte acceptable pour ne pas perturber le diagnostic. Quelques radiologues avant-gardistes préconisent même l'utilisation de la compression avec pertes, notamment avec des études récentes, comme celle de Koff *et al.* [KBB<sup>+</sup>09], où ils s'attachent à définir des taux de compression pour lesquels les distorsions sont suffisamment faibles pour être tolérées lors de tests subjectifs. À l'aide de ce type d'investigations, l'Association Canadienne des Radiologistes (CAR) a également publié une norme visant à orienter les radiologues sur les taux de compression maximaux pouvant être utilisés avec les standards JPEG et JPEG-2000 pour différentes modalités d'images et régions anatomiques étudiées [CAR11, CAR10]. Ces résultats sont focalisés sur des algorithmes particuliers et sont alors peu exploitables pour d'autres techniques de compression.

La plupart de ces études sont réalisées à l'aide de protocoles d'évaluation stricts, difficiles à mettre en place et coûteux<sup>8</sup>. Ces expertises portent souvent sur la qualité du diagnostic après compression. Ainsi, pour évaluer correctement un algorithme, il faut pouvoir réunir des images (de patients malades et sains) pour toutes les techniques d'acquisition (et de reconstruction) concernées et pour les pathologies associées. Ces images doivent ensuite être évaluées par un panel de médecins spécialistes, pour différents taux de compression (et donc niveaux de dégradation).

Pour s'astreindre des problèmes de l'évaluation subjective, la piste d'une métrique objective d'évaluation de la qualité par apprentissage sur des résultats d'experts a été envisagée [DRPV06]. Une telle approche, visant à « imiter » un expert, n'a, à notre connaissance, pas été utilisée afin d'optimiser les pertes en compression. Ceci pourrait pourtant permettre d'éviter l'introduction d'artefacts, de limiter la suppression de signes pathologiques et de moins perturber l'habitude des médecins. On trouve tout de même quelques références à des travaux utilisant des modèles psychovisuels hu-

---

<sup>8</sup>Se référer à [CMLCBT08, CMZLC10] pour plus de détails sur ces méthodes d'évaluation.

mains [KJML05], plus classiques en compression d'images naturelles. Un article récent de Cavaro-Menard *et al.* [CMZLC10] fait également un état des lieux des méthodes subjectives et objectives relatives à l'évaluation de la qualité des images médicales et présente les enjeux liés à l'utilisation de l'*eye-tracking* pour l'évaluation ou pour l'optimisation de la compression.

### 1.2.5 Archivage avec pertes contrôlées

Les médecins sont très réticents au passage à un archivage avec pertes. Ceci impliquerait la possibilité qu'une information importante pour le diagnostic en cours (ou un autre ultérieur) soit altérée et introduise des faux-négatifs ou des faux-positifs. En effet, l'information utile est parfois de l'ordre de quelques pixels [CMLCBT08]. C'est pourquoi la majorité des travaux en compression d'images médicales s'attachent à conserver la totalité de l'information (compression sans perte). Une option de stockage intermédiaire consiste à contrôler de manière stricte les pertes engendrées par la compression, de façon à conserver l'information nécessaire pour ne pas perturber le diagnostic.

#### 1.2.5.1 Notion de région d'intérêt

L'approche qui semble la plus satisfaisante, en terme de qualité, consiste à compresser sans perte la ou les régions d'intérêt diagnostique (ROI : *Region Of Interest*) et à compresser avec pertes (ou même supprimer) les zones pouvant être considérées comme d'un intérêt diagnostique nul (RONI : *Region Of No Interest*). Le plus souvent la ROI se compose de la totalité des pixels composant l'objet médical observé (patient), et la RONI des pixels extérieurs (les autres). Les contraintes de cette approche sont la nécessité de devoir définir la zone d'intérêt (effectué manuellement par le radiologue ou algorithmiquement par segmentation) pour un gain de compression, certes moins spectaculaire qu'avec une compression avec pertes, mais non négligeable : de 20% à 40% sur les IRM et scanners [TL10b] et 60% sur les lames virtuelles utilisées [TL11].

#### 1.2.5.2 Compression presque sans perte

Une autre option consiste à utiliser un critère de qualité pour guider l'algorithme de compression et lui imposer des limites sur les altérations que peut subir l'image. Un indicateur sous optimal mais garantissant une certaine qualité ou une métrique robuste d'évaluation des distorsions (suivant une approche telle que celle mentionnée dans 1.2.4 par exemple) peuvent également être mis en place.

Les techniques presque sans perte actuelles cherchent majoritairement à garantir l'erreur absolue maximale autorisée sur chaque pixel (cf. 3.1.4). Cette contrainte, même si elle est sous optimale, permet d'obtenir des taux de compression plus intéressants qu'en compression sans perte pour des seuils d'erreurs très faibles (En bornant simplement l'erreur sur chaque pixel par  $\pm 1$  un gain supérieur à 1 bpp peut être obtenu) qui n'ont pas ou très peu d'impact sur le résultat visuel. Il est tout de même envisageable d'avoir recours à des campagnes d'évaluations afin de déterminer un seuil de

tolérance optimale (pour chaque type de contenu, ou de modalité d'image, par exemple) ne perturbant pas le diagnostic.

La compression presque sans perte avec prise en compte de la région d'intérêt est également envisageable.

### 1.2.6 Remarques

**Sur les résultats de nombreuses publications en compression sans perte d'images médicales :** Beaucoup de publications ne prennent pas en compte la précision d'acquisition des technologies récentes et proposent des résultats sur des images 8 bits. Ces images sont rarement natives et résultent le plus souvent d'une quantification d'images de plus haute précision (12 bits généralement). Cette quantification ne rend plus vraiment les résultats exploitables, une grosse partie du bruit présent dans l'image originale étant ainsi supprimée. Or le bruit et les artefacts de reconstruction doivent être conservés pour que la compression soit sans perte, et ont souvent un impact non négligeable sur les performances de compression. Les résultats ne sont donc pas réellement comparables à ceux obtenus dans des articles prenant en compte la précision complète, et tenant ainsi compte du bruit d'acquisition.

**Sur la réticence des médecins à accepter la mise en place de système de compression avec pertes pour l'archivage :** Bien que des erreurs de diagnostic puissent être introduites par les pertes liées à la compression, des erreurs peuvent également être causées par le bruit et les artefacts de construction spécifiques au système d'acquisition, ou tout simplement par des erreurs humaines. Les erreurs de reconstruction liées aux contraintes physiques du matériel étant inévitables, les médecins se sont habitués à travailler avec elles, et les artefacts leurs sont devenus familiers. Ainsi, si un processus de compression avec pertes devait être instauré, il devrait être acceptable d'obtenir des taux d'erreurs de diagnostic similaires à ceux actuels après un temps d'adaptation des radiologues à ces nouvelles perturbations.

**Sur la compression avec pertes :** Si des systèmes de compression avec pertes sont mis en place pour l'archivage, la standardisation du protocole de décompression devra être totalement standardisée. En effet, dans le standard JPEG-2000 [TM01]-10.5, par exemple, un certain degré de liberté est laissé pour la reconstruction des images compressées avec pertes (précision des calculs, déquantification), qui implique que deux programmes de décompression différents, deux versions d'un même programme voire deux exécutions du même programme dans certains cas (notamment si des nombres en virgule flottante sont utilisés sans la rigueur nécessaire à la reproductibilité des calculs), n'obtiendront pas nécessairement la même image après décompression. Or ceci va à l'encontre d'une authenticité de l'image en tant qu'élément de preuve.

### 1.3 Fonctionnalités intéressantes pour la télé-consultation et l'archivage

Bien que le taux de compression soit un critère d'évaluation important pour l'archivage, les cadres applicatifs peuvent requérir une certaine flexibilité de la part des outils de compression. Ces contraintes jouent un rôle important sur le choix des techniques à mettre en place et sur l'organisation du flux de données compressées. Les principales fonctionnalités que peuvent chercher à offrir les formats de fichiers sont : la scalabilité, l'accès aléatoire, et la représentation sous forme d'objets. Fortement utiles, dans un contexte où la connectivité médicale numérique se met rapidement en place depuis quelques années déjà [Cor01], ces propriétés peuvent faciliter la manipulation du contenu et rendre plus efficace la transmission des données. Leur mise en place peut se faire dans le but de permettre une réduction des temps d'accès, des coûts de transferts et/ou de calculs au décodage, tout en essayant de rester efficace pour le stockage. Elles sont notamment utiles pour éviter i) une décompression totale du fichier si ce n'est pas nécessaire (lorsque manipulées localement, par exemple), ii) un transcodage si seulement certaines portions des données sont souhaitées ou qu'une image de plus faible qualité est requise (dans un schéma client/serveur fonctionnant à l'aide de requêtes sur un réseau à faible bande passante, par exemple), ou encore iii) l'engorgement des réseaux et implicitement les temps d'attentes lorsqu'une qualité optimale n'est pas nécessaire.

#### 1.3.1 Scalabilité en qualité

Même si la compression avec pertes pour l'archivage pose encore des problèmes d'éthique, celle-ci peut tout de même être utilisée afin d'accélérer la transmission sur des réseaux à faible débit. Par exemple, si l'image doit simplement être consultée à titre pédagogique ou pour illustrer un compte-rendu post-diagnostique, l'accès à la qualité sans perte n'est pas nécessairement requis.

Certains algorithmes, dits scalables en qualité ou progressifs, permettent une compression efficace tout en proposant différents niveaux de qualité, pouvant éventuellement aller jusqu'au sans perte (*lossy to lossless*), d'une même image dans un même fichier. L'image est d'abord transmise avec une faible qualité/précision (beaucoup de pertes/distorsions), puis est progressivement raffinée (réduction des distorsions) à l'aide de nouvelles informations. Ces algorithmes autorisent ainsi la transmission d'une portion restreinte du fichier compressé, permettant un accès avec pertes pouvant s'adapter à des contraintes de qualité de reconstruction ou de temps d'accès à l'image. L'avantage principal de cette technique est de ne pas nécessiter de transcodage (décompression du fichier archivé, puis compression avec pertes de l'image) avant la transmission, ce qui permet un gain de temps et des économies en ressources énergétiques.

Cette propriété peut également être utilisée pour de l'archivage à très long terme. Dans le cas de pathologies héréditaires, où les images sont conservées indéfiniment, on peut supposer que la qualité sans perte (requis principalement pour le diagnostic et la jurisprudence) n'est plus nécessaire au delà d'un certain nombre d'années, et que le fichier peut donc être conservé avec pertes. Là encore, le transcodage n'est pas nécessaire,



il suffit simplement de supprimer une portion de fichier devenue inutile.

Equitz et Cover [EC91] démontrent qu'une telle représentation sous forme d'arbre de raffinages successifs ne peut être optimale qu'à la condition que les différentes étapes de raffinement puissent s'écrire sous la forme d'une chaîne de Markov. Ainsi, avec l'utilisation de la compression scalable il ne faut pas espérer atteindre des performances de compression supérieures à des approches non progressives.

### 1.3.2 Scalabilité en résolution

La scalabilité en résolution peut être une alternative à celle en qualité. Cette approche permet une représentation multi-échelle de l'image (généralement  $1/2^k$ ). Elle peut être utile pour réduire la quantité d'informations à transférer lorsque l'image en pleine résolution n'est pas nécessaire, ou lorsqu'une navigation rapide, utilisant des images de tailles réduites (miniatures), est suffisante pour pouvoir sélectionner la ou les images qui intéressent avant de les obtenir en pleine résolution (*quick view for fast navigation*). Elle peut être couplée à des approches avec ou sans perte.

Avec certaines techniques, la scalabilité en résolution peut également être couplée à une scalabilité en qualité de manière à réduire progressivement les distorsions pour différentes résolutions et ainsi optimiser l'information à transmettre. Elle peut également être considérée, seule, comme une représentation progressive en qualité puisqu'il est possible d'interpoler l'image de sous-résolution afin de produire une estimation de l'originale, et que les erreurs d'interpolation diminuent généralement avec la différence de résolution.

### 1.3.3 Accès aléatoire

La compression de l'information rend impossible l'accès aléatoire immédiat à une information ou une plage d'informations dont on connaît la localisation dans l'espace de représentation pré-compression. Pour les images, un procédé usuel pour permettre d'y accéder rapidement est le découpage en zones uniformes (rectangulaires par exemple dans le cas bidimensionnel) et l'ajout d'informations supplémentaires qui permettent la localisation de ces différentes zones dans le flux compressé. Ainsi un accès aléatoire grossier peut être réalisé de façon à obtenir assez rapidement les données souhaitées.

Les formats de fichiers qui cherchent à offrir un accès aléatoire sont donc conçus à l'aide d'une en-tête et/ou de marqueurs dans le flux qui apportent des repères de localisation. Lorsque des modèles adaptatifs sont utilisés pour la compression, ceux-ci doivent être réinitialisés à chaque repère de localisation afin de permettre un décodage de chaque zone indépendamment des autres. Cette réinitialisation ainsi que l'information de localisation entraînent des pertes de compression. La mise en place de l'accès aléatoire a donc un coût, et plus la précision d'accès sera fine, plus il faudra s'attendre à voir le débit augmenter. Cependant il permet d'éviter le transfert et la décompression d'informations inutiles lorsque seule une faible portion géographique de l'image est requise/souhaitée.

### 1.3.4 Découpage en objets

Le découpage de l'image en diverses zones d'informations (régions pas spécialement uniformes) peut permettre de compresser chacune d'entre elles de manière indépendante, de sorte à pouvoir gérer plus facilement les données. L'accès aléatoire, la progressivité par zone, la progressivité globale avec favorisation de certaines régions, ou encore l'attribution de qualités différentes par région lors d'une compression avec pertes peuvent être envisagées grâce à un tel découpage. Ces différentes zones sont ainsi considérées comme des objets qui ont leurs propres propriétés de compression (si plusieurs algorithmes ou bases de transformation ont été mis en concurrence par exemple) et/ou peuvent posséder des informations supplémentaires de natures diverses (méta-données). Il s'agit alors de modéliser le découpage en objets, et d'organiser les divers flux de données pour répondre aux besoins. Il faut donc prévoir un coût supplémentaire pour les spécifications de la segmentation, pour la localisation des données dans le flux final, et éventuellement pour contenir les méta-données individuelles.

## Conclusion

Le contexte médical dans lequel s'insère ce travail a été introduit dans ce chapitre. La tomographie, l'IRM et la microscopie virtuelle sont en plein essor et posent des problèmes pour le stockage et la consultation à distance. L'archivage de ces images de très grandes tailles est soumis à des contraintes législatives lourdes. Bien qu'une compression avec pertes puisse permettre de réduire les coûts d'archivage, les médecins restent réticents à une telle mesure. Des approches sans perte ou avec pertes contrôlées semblent donc à privilégier. Cependant, la compression avec pertes reste envisageable et intéressante pour les télé-consultations post-diagnostiques. Des fonctionnalités favorisant les manipulations de ces images à distance ont donc également été introduites.

La suite de ce manuscrit s'intéressera essentiellement à la compression, sans perte, presque sans perte, ainsi qu'aux algorithmes scalables en résolution ou en qualité.



## Chapitre 2

# Outils génériques pour la compression d'images

### Introduction

Afin de mieux appréhender le prochain chapitre qui dressera un état de l'art en compression d'images, ce chapitre fait un balayage rapide des principaux outils théoriques et pratiques sur lesquels s'appuient la majorité des algorithmes de compression.

Pour rester général, les éléments nécessaires à la compression de données quelconques sont tout d'abord introduits dans une première section (sec. 2.1). Les fondements de la théorie de l'information y sont rappelés, puis les principales méthodes pour le codage entropique y sont décrites. Comme les signaux et les images numériques ont des propriétés assez particulières : ils contiennent généralement une information redondante et/ou superflue, des outils qui leur sont plus spécifiques sont ensuite présentés (sec. 2.2). Ces outils concernent la décorrélation des données, qui sert à réduire la quantité d'information nécessaire à leur représentation, ainsi que les procédés d'approximation, qui permettent de supprimer une partie de l'information lorsque celle-ci est considérée comme superflue.

## 2.1 Compression de données

### 2.1.1 Théorie de l'information

La théorie de l'information définit les fondements mathématiques de la compression moderne. Elle fut introduite en 1948 par Claude Shannon [Sha48] en démontrant la limite de la compression de données numériques. Cette limite, appelée entropie, s'exprime comme le nombre moyen de symboles équiprobables nécessaires pour représenter un message provenant d'une source aléatoire d'information dont on connaît le modèle statistique. Ne sont rappelées ici que les principales définitions qui seront utiles pour la suite.

**Définitions 2.1.1 : ALPHABET, LONGUEUR ET SYMBOLES**

Un ALPHABET  $\mathcal{A}$  est un ensemble dont la LONGUEUR, ou cardinalité, est  $|\mathcal{A}| = \text{Card}(\mathcal{A}) \neq 0$ . Les éléments de cet alphabet  $\{a_1, \dots, a_{m=|\mathcal{A}|}\}$  sont ordonnés et appelés SYMBOLES.

**Définition 2.1.2 : MODÈLE**

Soit un alphabet  $\mathcal{A}$ , un MODÈLE probabiliste  $\mathbb{M}$  est une fonction

$$\begin{aligned} \mathbb{M} : \mathcal{A} &\mapsto [0, 1] \\ a_i &\rightarrow P_{\mathbb{M}}(a_i), \quad \sum_i P_{\mathbb{M}}(a_i) = 1, \end{aligned} \quad (2.1)$$

qui associe une probabilité  $P_{\mathbb{M}}(a_i)$  à chaque symbole  $a_i$  de  $\mathcal{A}$ . Cette probabilité peut être une estimation et n'est donc pas nécessairement la probabilité réelle  $P(a_i)$ .

**Définition 2.1.3 : INFORMATION**

Soit un symbole  $a_i \in \mathcal{A}$  en provenance d'une variable aléatoire  $X$  supposée suivre une loi de probabilité correspondant à un modèle  $\mathbb{M}$ . D'après le théorème de Shannon, la quantité d'INFORMATION nécessaire à exprimer  $a_i$  à l'aide d'un alphabet  $\Omega$  de longueur  $W$  est

$$\mathcal{I}_{\mathbb{M}}(X = a_i) = -\log_W P_{\mathbb{M}}(X = a_i). \quad (2.2)$$

Cette mesure correspond à la longueur de la séquence de symboles de  $\Omega$  nécessaire à exprimer de manière optimale le symbole  $a_i$ , en concordance avec le modèle  $\mathbb{M}$ .

En informatique on utilise le plus souvent  $\Omega = \{0, 1\}$  pour un codage binaire de l'information, et on exprime alors la quantité d'information en bits/symbole.

**Définitions 2.1.4 : ENTROPIE ET ENTROPIE DE SHANNON**

En utilisant le modèle de probabilité a priori  $\mathbb{M}$ , l'ENTROPIE  $H_{\mathbb{M}}(X)$  de la variable aléatoire  $X$  se définit alors comme

$$H_{\mathbb{M}}(X) = \sum_{a_i \in \mathcal{A}} P(X = a_i) \mathcal{I}_{\mathbb{M}}(X = a_i), \quad (2.3)$$

et correspond à la quantité moyenne d'information nécessaire pour coder un symbole de  $\mathcal{A}$  dont les statistiques d'apparition sont décrites par le modèle  $\mathbb{M}$ .

Dans le cas d'un modèle parfait ( $P_{\mathbb{M}}(a_i) = P(a_i), \forall a_i \in \mathcal{A}$ ), on obtient l'ENTROPIE DE SHANNON  $H(X)$  qui correspond à la quantité moyenne d'information nécessaire pour coder un symbole de  $\mathcal{A}$  de façon la plus optimale :

$$H(X) = - \sum_{a_i \in \mathcal{A}} P(X = a_i) \log_W P(X = a_i). \quad (2.4)$$

**Définition 2.1.5 : ENTROPIE CONDITIONNELLE**

L'ENTROPIE CONDITIONNELLE de  $X$  sachant  $Y$  se définit par :

$$H(X|Y) = - \sum_{x,y \in \mathcal{A}} P(X = x, Y = y) \log_W P(X = x|Y = y). \quad (2.5)$$

**Définition 2.1.6 : ENTROPIE JOINTE**

L'ENTROPIE JOINTE de la variable aléatoire discrète formée par le couple  $(X, Y)$  se déduit de (2.4) :

$$H(X, Y) = - \sum_{x,y \in \mathcal{A}} P(X = x, Y = y) \log_W P(X = x, Y = y). \quad (2.6)$$

vérifiant

$$H(X, Y) = H(X) + H(Y|X) = H(Y) + H(X|Y). \quad (2.7)$$

Ainsi

$$H(X, Y) \leq H(X) + H(Y), \quad H(X, Y) \geq H(X) \quad \text{et} \quad H(X, Y) \geq H(Y), \quad (2.8)$$

et  $H(X, Y) = H(X) + H(Y)$  quand  $X$  et  $Y$  sont indépendantes.

**Définition 2.1.7 : MESSAGE**

Un MESSAGE  $\mathcal{M} = (X_t)_{t \in \{1, \dots, T\}}$  de taille  $T$  est la réalisation d'un ensemble de variables aléatoires ordonnées (processus)  $X_t$  à valeurs dans un alphabet  $\mathcal{A}_t = \{a_{0,t}, \dots, a_{m_t,t}\}$ , suivant une loi de probabilité  $\mathcal{P}_{X_t}$  qui peut dépendre de la réalisation des autres variables.

Selon cette définition, une variable  $X_t$  d'un message  $\mathcal{M}$  de longueur  $T$  aurait pour entropie

$$H(X_t) = - \sum_{(a_{i_1}, \dots, a_{i_T}) \in \mathcal{A}_1 \times \dots \times \mathcal{A}_T} P(X_1 = a_{i_1}, X_2 = a_{i_2}, \dots, X_T = a_{i_T}) \cdot \log_W P(X_t = a_{i_t} | (X_k = a_{i_k})_{k \neq t}). \quad (2.9)$$

Clairement, pour la compression, cette expression ne convient pas à cause de la dépendance qui existe entre les variables et qui rend le modèle non causal. Pour qu'elle soit rendue causale, l'entropie de la variable  $X_t$  peut s'exprimer comme étant

$$H(X_t) = - \sum_{(a_{i_1}, \dots, a_{i_t}) \in \mathcal{A}_1 \times \dots \times \mathcal{A}_t} P(X_1 = a_{i_1}, X_2 = a_{i_2}, \dots, X_t = a_{i_t}) \cdot \log_W P(X_t = a_{i_t} | (X_k = a_{i_k})_{k < t}). \quad (2.10)$$

Classiquement, on modélise un message d'une façon beaucoup plus simple par un modèle de Markov d'ordre  $k$ . Ainsi la loi de probabilité de toutes variable aléatoire  $X_t$  ne dépend plus que de  $k$  variables qui la précèdent. On exprime alors l'entropie

conditionnelle

$$H_k(X_t) = - \sum_{(a_{i_0}, \dots, a_{i_k}) \in \mathcal{A}_t \times \dots \times \mathcal{A}_{t-k}} \mathbb{P}(X_t = a_{i_0}, X_{t-1} = a_{i_1}, \dots, X_{t-k} = a_{i_k}) \cdot \log_W \mathbb{P}(X_t = a_{i_0} | (X_{t-j} = a_{i_{t-j}})_{j \in \{1..k\}}). \quad (2.11)$$

De même, en compression, pour que le traitement puisse être rendu causal, les modélisations des lois de probabilités  $\mathcal{P}_{X_t}$  ne dépendent que d'informations connues : des données fixées au codeur et au décodeur et/ou des informations précédemment encodées/transmises. Elles peuvent éventuellement contenir des spécifications sur le(s) modèle(s) à utiliser.

### Définition 2.1.8 : ENTROPIE DE LA SOURCE

Dans la littérature, on voit parfois apparaître le DÉBIT ENTROPIQUE DE LA SOURCE (aussi appelé ENTROPIE DE LA SOURCE). Il peut se définir à l'aide de l'entropie jointe (2.6) pour une source  $\mathcal{X}$ ,

$$H(\mathcal{X}) = \lim_{N \rightarrow \infty} \frac{1}{N} H(X_0, \dots, X_{N-1}). \quad (2.12)$$

En pratique, on considère des messages  $\mathcal{M}$  finis de longueur  $N$ . L'entropie est alors

$$H(\mathcal{M}) = \frac{1}{N} H(X_0, \dots, X_{N-1}). \quad (2.13)$$

La source est, par exemple, l'image à compresser et les lois de probabilités sont calculées à partir du nombre d'occurrence de chaque symbole. Il arrive souvent que l'entropie de la source exprimée soit en fait  $H(X)$  (ou entropie d'ordre 0), en considérant que l'image est une succession de réalisations indépendantes d'une variable  $X$  dont la loi de probabilité est donnée par :

$$\mathbb{P}(a_i) = \frac{n_{a_i}}{N}, \quad (2.14)$$

où  $n_{a_i}$  est le nombre d'occurrences du symbole  $a_i$  dans le message.

### 2.1.2 Codage entropique

Un codeur entropique permet de compresser une séquence de symboles en se basant sur leur probabilité (a priori) d'apparition. Chaque symbole se voit assigner une nouvelle représentation (dépendant de sa probabilité d'apparition) de manière à ce que le codage de la source s'approche au plus de son entropie. Les codeurs entropiques qui vont être présentés permettent de compresser l'information en générant un code sans ambiguïté. Ils autorisent donc de faire une restitution sans perte. Il existe également des techniques dérivées qui peuvent dépasser la limite entropique en induisant des ambiguïtés et donc un codage avec pertes, mais elles ne seront pas abordées.

### 2.1.2.1 Codes à longueur variable

Huffman et Golomb ont proposé des techniques de construction de codes à longueur variable (VLC : *Variable Length Code*) pour la compression. Les VLC sont très utilisés car ils sont simples à mettre en place, rapides et souvent efficaces.

Un codeur VLC fonctionne de la façon suivante : pour tout symbole  $a_{\mathcal{A}_{\mathcal{I}},i}$  de l'alphabet d'entrée  $\mathcal{A}_{\mathcal{I}}$  de longueur  $N_{\mathcal{I}}$ , le codeur VLC  $\mathcal{K}_{\mathbf{M}}$ , conditionné par un modèle  $\mathbf{M}$ , associera une unique séquence de  $N_{\mathbf{M},i}$  symboles de l'alphabet de sortie  $\mathcal{A}_{\mathcal{O}}$ , appelée mot. On définit ainsi un alphabet  $\mathcal{AS}_{\mathcal{O}}$  pour lequel chaque symbole est une séquence de symboles de l'alphabet  $\mathcal{A}_{\mathcal{O}}$  :

$$\mathcal{AS}_{\mathcal{O}} = \{a_{\mathcal{AS}_{\mathcal{O}},i}\}_{i \in \{1..N_{\mathcal{I}}\}} / a_{\mathcal{AS}_{\mathcal{O}},i} \in \mathcal{A}_{\mathcal{O}}^{N_{\mathbf{M},i}}. \quad (2.15)$$

Ainsi, un tel codeur  $\mathcal{K}_{\mathbf{M}}$  peut être représenté comme une fonction qui, à un symbole dans un contexte donné (modèle  $\mathbf{M}$ ), associera un unique mot (codage « 1 → 1 »), cette association est bijective afin de permettre le décodage :

$$\begin{aligned} \mathcal{K}_{\mathbf{M}} : \quad \mathcal{A}_{\mathcal{I}} &\mapsto \mathcal{AS}_{\mathcal{O}} \\ a_{\mathcal{A}_{\mathcal{I}},i} &\rightarrow a_{\mathcal{AS}_{\mathcal{O}},i}. \end{aligned} \quad (2.16)$$

L'avantage majeur de tels codeurs est leur rapidité. Leur principal inconvénient est qu'ils sont souvent sous optimaux puisqu'ils ne proposent que des séquences de symboles  $a_{\mathcal{AS}_{\mathcal{O}},i}$  de longueurs entières.

**2.1.2.1.1 Codes de Huffman** Les codes de Huffman, mis au point en 1952 [Huf52], sont parmi les premiers à avoir émergés et sont donc également parmi les plus répandus. Le principe est d'attribuer un code court à un symbole fréquent et un code plus long à un symbole plus rare. En se plaçant dans le cas d'un alphabet de sortie binaire, l'attribution du nombre de bits pour chaque symbole se fait par le biais d'un arbre binaire, construit en fonction de la fréquence d'apparition des symboles.

A chaque nœud de l'arbre est associé la fréquence d'apparition de l'ensemble des symboles présents dans son sous arbre. L'arbre est construit de manière ascendante, à partir de ses feuilles. Chaque feuille correspond à un symbole, et à sa fréquence d'apparition. Itérativement, jusqu'à l'obtention de la racine, un nœud père est généré pour le couple des deux nœuds ayant la fréquence d'apparition la plus faible.

Le code de chacun des symboles est alors le chemin partant de la racine jusqu'à la feuille lui correspondant (bit 0 pour aller au fils gauche, et bit 1 pour aller au fils droit par exemple). Ce code est donc défini de telle sorte qu'aucun des symboles de l'alphabet  $\mathcal{AS}_{\mathcal{O}}$  ne soit le préfixe d'un autre : chaque symbole étant associé à une feuille de l'arbre le chemin qui permet d'y accéder ne peut en aucun cas être le sous-chemin d'un autre. Ainsi, le codage est non ambigu.

La création de cet arbre peut être effectuée à l'aide d'un algorithme de complexité  $O(N_{\mathcal{I}} \log_2 N_{\mathcal{I}})$ , avec  $N_{\mathcal{I}}$  la taille de l'alphabet.



Le code Huffman peut être statique, transmis ou adaptatif. Lorsqu'il est statique, le code est fixé pour le codeur et le décodeur et n'a donc pas besoin d'être transmis. Cependant, il n'est pas toujours adapté aux données.

Lorsqu'il est transmis, l'arbre doit être construit par le codeur à partir des fréquences d'apparition des symboles : un parcours supplémentaire des données est donc nécessaire (sur la totalité, ou un échantillon représentatif). Il a également un coût pour la compression puisqu'il doit être transmis.

Enfin lorsqu'il est adaptatif, en partant d'un arbre particulier les fréquences des symboles sont mises à jour au fur et à mesure de leur apparition. L'arbre doit donc être mis à jour régulièrement, ce qui demande un temps de calcul important. L'arbre de départ peut également être statique ou transmis.

#### Avantages :

- la rapidité (le plus rapide, une fois l'arbre binaire généré) : avec une table indexée par les symboles d'entrée, le temps de codage dépend uniquement de la recopie (dans le message de sortie) de la séquence associée au symbole d'entrée.
- ce codage est optimal pour les modèles où  $\forall a_{\mathcal{A}_{\mathcal{I}},i} \in \mathcal{A}_{\mathcal{I}}, -\log_W P_{\mathbb{M}}(a_{\mathcal{A}_{\mathcal{I}},i})$  est une valeur entière (i.e.  $P_{\mathbb{M}}(a_{\mathcal{A}_{\mathcal{I}},i}) = W^{-k}, k \in \mathbb{N}$ ), avec  $W = 2$  puisque l'alphabet  $\mathcal{A}_{\mathcal{O}} = \{0, 1\}$ .

#### Inconvénients :

- un modèle où  $P_{\mathbb{M}}(a_{\mathcal{A}_{\mathcal{I}},i}) = 2^{-k \in \mathbb{N}}$  est peu courant. Dans toute autre situation, l'entropie liée au codage de Huffman  $H_{\text{Huff}}$  s'éloigne de l'entropie du modèle  $H_{\mathbb{M}}$  en associant à chaque symbole une séquence de longueur entière, sous optimale, et vérifie  $H_{\mathbb{M}} \leq H_{\text{Huff}} < H_{\mathbb{M}} + 1$ . Dans le pire des cas, presque 1 bit sera donc perdu en moyenne par symbole ce qui peut être assez pénalisant lorsque  $H_{\mathbb{M}}$  est faible. Ce codage reste tout de même efficace lorsque  $P_{\mathbb{M}}(a_{\mathcal{A}_{\mathcal{I}},i}) \simeq 2^{-k \in \mathbb{N}}$
- le modèle  $\mathbb{M}$  est difficilement adaptatif. Ceci nécessite un recalcul complet de l'alphabet de sortie  $\mathcal{A}_{\mathcal{O}}$  à chaque phase d'adaptation, ce qui est coûteux en temps de calcul. Pour pallier à ce problème, on peut utiliser des informations contextuelles afin de sélectionner un modèle prédéfini et le codage Huffman associé, par exemple.

**2.1.2.1.2 Codes de Golomb** Dans le principe, un code Golomb est assez similaire à un code Huffman. La principale différence vient du fait qu'on suppose que les variables d'entrée suivent une certaine loi de probabilité qui va implicitement induire le codage. Cette propriété permet de ne pas avoir besoin de calculer la table de correspondances des symboles, et le rend ainsi facilement adaptatif [WSS96, TM01, Mal06].

Un codeur Golomb [Gol66] de paramètre  $m$  noté  $G_m$  encode une valeur entière positive  $n$  en deux parties : une représentation binaire de  $(n \bmod m)$  et une représentation unaire de  $\lceil n/m \rceil$ . Les codeurs Golomb sont optimaux pour coder des valeurs entières positives, suivant une loi de probabilité géométrique de la forme  $Q(n) = (1 - \rho)\rho^n$  avec  $0 < \rho < 1$  pour une valeur  $m = \lceil \log(1 + \rho)/\log(\rho^{-1}) \rceil$ .

Les symboles de l'alphabet  $\mathcal{AS}_{\mathcal{O}}$  sont construits par la concaténation de la représentation binaire de taille fixe de  $(n \bmod m)$  et de la représentation unaire de taille variable de  $\lfloor n/m \rfloor$ . Pour que le codage soit non ambigu, la représentation unaire est une suite de bits à 1 suivie d'un bit d'arrêt à 0.

Les codeurs Golomb-Rice [RP71, GvV75] sont un cas particulier du codeur Golomb avec  $m = 2^k$ , particulièrement adaptés pour le codage/décodage informatique. Ainsi les divisions se font par simple décalage de bits et les modulo par une opération logique utilisant un masque.

**Avantages :**

- la rapidité (légèrement plus lent que le codage Huffman) : il nécessite uniquement un modulo pour obtenir les symboles représentant la partie de poids faible des symboles d'entrée, et d'une division suivie du codage unaire (une table comme pour Huffman pourrait être utilisée, mais la propriété suivante serait perdue),
- il est facilement rendu adaptatif : seul le paramètre  $m$  (ou  $k$ ) nécessite d'être estimé,
- ce codage est optimal pour des modèles où les valeurs suivent une loi géométrique, ce qui est une bonne approximation de la répartition des valeurs à compresser dans les schémas destinés aux signaux ou aux images.

**Inconvénients :**

- comme pour les codes de Huffman, il y a une perte entropique due à l'encodage VLC (« 1 → 1 »). Cette perte peut cependant être diminuée à l'aide de systèmes adaptatifs [WSS96].

Ce codeur et ses dérivés sont très présents dans la littérature pour le codage d'erreurs de prédiction en compression sans perte d'images et de signaux audio, domaines pour lesquels il a tendance à être plus utilisé qu'Huffman puisqu'il nécessite beaucoup moins de calculs pour être rendu adaptatif. En effet l'adaptation du code Huffman nécessite le recalcul de l'arbre complet ( $O(N_{\mathcal{I}} \log_2 N_{\mathcal{I}})$ ), tandis que pour le code Golomb seul le paramètre  $m$  doit être mis à jour ( $O(1)$ ). Les valeurs à encoder ont également tendance à suivre une loi géométrique en valeur absolue, ce qui permet au code Golomb d'être efficace.

### 2.1.2.2 Codage arithmétique

Très prisé pour ses performances, le codeur arithmétique permet de compresser une séquence de symboles  $\mathcal{A}_{\mathcal{I}}^N$  en une séquence de symboles  $\mathcal{A}_{\mathcal{O}}^M$  de telle sorte que chaque symbole d'entrée ne soit pas nécessairement associé à un nombre entier de symboles de l'alphabet de sortie. Il évite ainsi le problème des codeurs VLC et est quasiment optimal au sens de  $H_{\mathbb{M}}(X)$ . Autrement dit, ce sera principalement la qualité du modèle

statistique  $M$  qui influera sur l'efficacité de la compression, et permettra de s'approcher de la limite théorique.

Un tel codeur permet de générer un message de sortie  $\mathcal{M}_O$  en un temps proportionnel à la taille du message d'entrée  $\mathcal{M}_I$  (complexité linéaire). Cependant il nécessite des traitements plus complexes que ceux d'un codeur VLC qui le rendent plus lent.

Le principe consiste à coder toute une séquence de symboles en un unique nombre décimal dont la précision permet de régénérer la séquence originale. Ce nombre correspond à la borne inférieure d'un intervalle dans lequel la séquence est plongée. On part d'un intervalle  $I_0 = [x_0 = 0, y_0 = 1)$ . Soit une séquence de symboles  $(X_t)_{t \in 1..T}$ , appartenant à l'alphabet  $\mathcal{A}_T = \{a_1 \dots a_{N_{\mathcal{A}_T}}\}$ , dont la distribution statistique est modélisée par le modèle  $M$ . Si  $I_k$  est l'intervalle obtenu après le codage du  $k$ -ième symbole,  $I_k$  est subdivisé en  $N_{\mathcal{A}_T}$  sous-intervalles de longueurs  $M(a_i) \|I_k\|$ , proportionnelles aux probabilités d'apparition de chaque symbole. L'intervalle  $I_{k+1}$  correspondra alors au  $X_{k+1}$ -ième sous-intervalle associé au symbole à coder.

Ainsi l'intervalle  $I_0$  est subdivisé et réduit symbole après symbole, en fonction du modèle, de manière non ambiguë en un intervalle  $I_N = [x_N, y_N)$ .  $x_N$  est progressivement codé (affiné) après chaque subdivision  $k$  avec une précision permettant de distinguer  $x_k$  de  $y_k$ , et rend donc également le codage non ambiguë.

#### Avantages :

- ce codage permet d'atteindre des taux très proches de l'entropie théorique,
- il est facilement adaptatif, sans avoir à modifier l'algorithme de compression lui-même, mais le modèle qu'il utilise.

#### Inconvénients :

- le codage arithmétique demande un peu plus de temps de calcul que les codeurs VLC : ces derniers sont quasiment instantanés puisqu'ils ne nécessitent que d'une lecture en mémoire du code et d'une recopie, tandis que le codeur arithmétique doit effectuer quelques opérations (au minimum 2 multiplications, 1 division, 3 additions et 2 soustractions pour chaque symbole dans l'implémentation de [BCK07]) afin de mettre à jour les bornes de l'intervalle,
- le codage est très dépendant des symboles précédemment encodés, ce qui le rend peu robuste aux erreurs de transmission, et empêche le décodage à partir d'une position aléatoire dans le flux de données.

On peut trouver de bons supports pour comprendre et programmer un tel codeur comme celui d'Amir Said [Sai04] ou celui d'Eric Bodden, Malte Clasén et Joachim Kneis [BCK07] (code source inclus). Dans les travaux de cette thèse, c'est sur l'implémentation de ce dernier que s'appuie le codage arithmétique lorsqu'il est employé.

Il existe de nombreux codeurs arithmétiques, tous basés sur le même principe, mais visant à offrir des réponses à ces inconvénients. Certains cherchent à réduire le temps de calcul en proposant une légère perte de l'optimalité du codage. D'autres essaient

d'inclure des informations redondantes afin de permettre une correction d'erreur plus aisée en cas de pertes d'informations durant la transmission.

Comme ils permettent la représentation des symboles sur un nombre fractionnaire de bits, ils sont bien adaptés à la compression de messages ayant un petit alphabet. Un bon nombre de codeurs arithmétiques binaires ont donc été développés, parmi lesquels on peut citer le Q-Coder d'IBM [PMLA88] qui utilise son propre modèle adaptatif et qui réduit le temps de calcul en supprimant les multiplications par des approximations, au détriment d'une légère perte de compression. Il est l'ancêtre du QM-Coder utilisé dans JBIG et JPEG (JPEG peut aussi utiliser un autre codeur dérivé appelé Q15 [ITU05]) et du MQ-Coder utilisé par JBIG2 et JPEG2000 [MYRP07]. On peut également citer celui d'Amir Said FastAC<sup>1</sup> (*Fast Arithmetic Coding*), CABAC (*Context-Adaptive Binary Arithmetic Coding*) utilisé par H.264/MPEG-4 AVC, ou encore le codeur arithmétique rapide utilisé par le codec video DIRAC<sup>2</sup> de la BBC Research qui est réalisé pour être entièrement disponible dans le domaine public et exempt de tout brevet.

### 2.1.2.3 Autres outils de codage

Les codes VLC et arithmétiques sont issus d'un modèle statistique explicite : les probabilités "a priori" des occurrences des symboles sont fournies au codeur. Une autre classe de codeurs entropiques se distingue également, pour laquelle le modèle statistique est implicite : les probabilités sont indirectement exploitées. Deux exemples répandus seront présentés ici.

L'un des codages les plus simples, et très utilisé, est le RLE (*Run Length Encoding*). Il est très efficace dans les situations où l'on sait que de longues séquences d'un même symbole doivent survenir. Il consiste simplement à spécifier le symbole et la longueur de la séquence. Cette technique est utilisée en compression d'image en complément des codes VLC afin de pallier à leur manque d'efficacité à compresser ces longues séquences de symboles identiques. Le plus souvent, un symbole spécial est utilisé pour préciser le changement de mode (VLC vers RLE), mais celui-ci peut également se faire de manière automatique en fonction du contexte local comme dans LOCO-I (cf. 3.1.1.1.2).

Le codage par substitution est une autre approche utilisée en compression de données. Il est souvent très efficace sur du texte ou certains types d'images dont le contenu n'est pas très riche : répétitif ou aplats de couleurs, palette de couleurs limitée, etc. Cette technique consiste à générer un dictionnaire de portions du message qui sont redondantes, et remplacer leurs occurrences par leur index dans le dictionnaire. Elle est apparue avec l'algorithme LZ77 [ZL77] et a été combinée avec un codage Huffman pour produire l'algorithme de compression DEFLATE (RFC 1951) qui n'est soumis à aucun brevet. Il est désormais utilisé dans des formats de compression très répandus comme ZIP (RFC 1950) et GZIP (RFC 1952), et en compression sans perte d'images

---

<sup>1</sup><http://www.cipr.rpi.edu/~said/FastAC.html>

<sup>2</sup><http://www.bbc.co.uk/rd/projects/dirac/index.shtml> et <http://diracvideo.org/>

par le format PNG (RFC 2083). Ce dernier fut créé afin de contrecarrer sa variante propriétaire LZW (Lempel Ziv Welch) utilisée dans les fichiers GIF (*PNG's Not Gif*) à une époque où le brevet sur LZW n'avait pas encore expiré.

**Avantages du codage par substitution :**

- il est particulièrement bien adapté à des séquences de données répétitives,
- il peut permettre de s'abstenir d'une étape de décorrélation des données (c.f. 2.2),
- contrairement aux codes VLC, la substitution d'une séquence peut permettre un coût de codage inférieur au bit par symbole.

**Inconvénients :**

- ce type d'algorithme peut requérir un espace mémoire assez conséquent pour le stockage du dictionnaire,
- il est rarement optimal.

#### 2.1.2.4 Quel codeur entropique utiliser ?

Le cadre d'application guide généralement le choix du codeur entropique. Si la compression doit favoriser la rapidité et qu'il est possible de modéliser une ou plusieurs distributions de probabilités représentatives (et pertinentes) des symboles de l'alphabet, les codes de Huffman sont souvent préférés.

L'adaptation automatique des distributions statistiques permet de rendre le modèle plus pertinent et d'améliorer la compression, au détriment d'une baisse de rapidité. Les codes Golomb-Rice adaptatifs sont réputés pour fournir un bon compromis entre rapidité et taux de compression lorsqu'il s'agit de signaux ou d'images naturelles, après une étape de décorrélation (c.f. section suivante). Si l'application est moins focalisée sur la vitesse d'exécution que sur les performances de compression, les codeurs arithmétiques adaptatifs sont privilégiés (c'est l'orientation qui a été prise pour cette thèse). En particulier, lorsque la quantité d'information pour le symbole le plus fréquent est trop faible (bien en dessous de 1 bit), même si la combinaison VLC et RLE peut être efficace, un codeur arithmétique adaptatif sera plus optimal. Celui-ci a également l'avantage de pouvoir coder efficacement l'information par plans de bits (alphabet de 2 symboles : l'entropie est alors inférieure ou égale à 1 bit par symbole). Cette technique, permet aux algorithmes de compression d'images par transformée, tel que JPEG-2000 par exemple (c.f. chapitre 3), de proposer des fonctionnalités intéressantes comme la scalabilité en qualité (c.f. 1.3.1).

## 2.2 Réduction de l'information

### 2.2.1 Signaux numériques

Beaucoup de signaux sont issus de phénomènes physiques, et plus généralement d'ondes (compression/décompression de matériaux pour les ondes sonores, ondes électromagnétiques pour les photographies, ...). Ces ondes peuvent être modélisées à l'aide de

fonctions (ou variables aléatoires) continues à valeurs réelles. Prenons pour exemple une fonction  $f(t)$  telle que  $f : \mathbb{R}^+ \mapsto \mathbb{R}$ , et considérons  $t$  comme le temps. Sur un intervalle de temps  $[t_0; t_1]$  ( $t_0 \neq t_1$ ), aussi petit qu'il soit,  $f$  possède une infinité de réalisations. De même pour un instant  $t$  donné  $f(t)$  a une précision numérique infinie, puisqu'à valeur réelle. Les supports numériques ayant une capacité limitée (nombre de bits), ces fonctions à précision infinie ne peuvent pas être stockées directement. Ainsi la numérisation n'est pas immédiate, il faut discrétiser et limiter le nombre d'instants auxquels on souhaite conserver  $f$  mais également le nombre de valeurs pouvant être prises par  $f(t)$ .

La discrétisation (ou échantillonnage) de  $[t_0, t_1]$  se fait le plus souvent à pas fixe  $T$ . On conservera alors les valeurs  $\{F(t_0), F(t_0 + T), F(t_0 + 2T), \dots\}$ , où  $F : \mathbb{R}^+ \mapsto \mathbb{R}$  est issue de  $f$  (par exemple  $F(t) = f(t)$ , ou encore  $F(t) = \frac{1}{T} \int_t^{t+T} f(u) du$ ). Cette échantillonnage est effectué habituellement de manière à respecter le théorème de Nyquist-Shannon : le signal  $F$  ne doit posséder aucune composante fréquentielle au delà de la plage  $[-\frac{1}{2}S_F; \frac{1}{2}S_F]$  où  $S_F = \frac{1}{T}$  est appelé fréquence d'échantillonnage. Cette contrainte est suffisante pour pouvoir restituer toutes valeurs prises par  $F(t)$  dans l'intervalle de temps initial  $[t_0, t_1]$ .

La discrétisation des valeurs prises par  $F(t)$  s'effectue ensuite à l'aide d'une fonction  $Q : \mathbb{R} \mapsto \mathcal{A}_Q$ , où  $\mathcal{A}_Q$  est l'alphabet fini des valeurs autorisées pour leur représentation numérique. Cette opération n'est autre qu'une quantification (cf. 2.2.3) effectuée sur un alphabet initial de taille infinie.

La représentation numérique d'un signal réel est imparfaite : des pertes d'informations sont introduites par ces discrétisations. Un compromis doit être fait entre la précision du signal et le nombre de bits nécessaires pour sa représentation. Interviennent dans ce choix des facteurs de coût de production du matériel d'acquisition et de qualité nécessaire ou « ressentie » par l'utilisateur des signaux numérisés. Dans le cas des TDM et des IRM, les discrétisations de l'espace d'acquisition agissent sur la qualité des images reconstruites et implicitement sur leur résolution et sur le nombre de bits par pixel réellement informatifs. Tandis que pour les LV, elles concernent directement la résolution et la précision d'acquisition.

## 2.2.2 Décorrélation

Les échantillons composant les signaux sont en général corrélés et possèdent donc une information redondante. Une étape de décorrélation visant à supprimer cette redondance s'avère alors très bénéfique pour l'amélioration des taux de compression. Les deux approches principales qui servent à effectuer cette décorrélation sont la prédiction et la transformation des données.

### 2.2.2.1 Prédiction

La valeur probable  $\hat{x}_i$  de l'échantillon  $x_i$  est estimée à l'aide de l'information causale. Pour effectuer la décorrélation, seule l'erreur  $\epsilon_i$  de la prédiction de  $x_i$  par  $\hat{x}_i$  est alors conservée :

$$\epsilon_i = x_i - \hat{x}_i. \quad (2.17)$$

L'information causale est généralement composée de l'ensemble des échantillons précédemment décorrélés  $x_{j \in \{0..i-1\}}$  (ainsi que de toutes valeurs pouvant être calculées à partir de ceux-ci), mais peut également inclure d'autres paramètres précédemment transmis ou fixés.

Le processus de prédiction peut alors être inversé, en utilisant la même information causale, pour reconstruire l'échantillon original :

$$x_i = \epsilon_i + \hat{x}_i. \quad (2.18)$$

Les techniques de prédiction les plus efficaces tentent d'adapter progressivement des modèles statistiques contextuelles sur les données causales de sorte à minimiser la variance, voire l'entropie, de l'information résiduelle  $\epsilon$ .

### 2.2.2.2 Transformation

La transformée  $T : s \rightarrow \hat{s}_T$  permet de changer l'espace de représentation d'un signal  $s$  en coefficients  $\hat{s}_T$ . Lorsqu'elle est inversible, il existe une transformée inverse  $T^{-1}$  telle que  $T^{-1}(T(s)) = s$ . Pour la compression, on utilise des transformées inversibles avec de bonnes propriétés de décorrélation, afin que l'information contenue dans l'espace transformé soit plus compacte que dans l'espace d'origine. Deux classes de transformées discrètes inversibles sont distinguées : les transformées redondantes et non redondantes. Si  $T$  est redondante, alors  $\text{Card}(T(s)) > \text{Card}(s)$ . En compression sans perte on préfère souvent le cas non redondant :  $\text{Card}(T(s)) = \text{Card}(s)$ .

La majorité des transformées théoriquement inversibles ne le sont pas totalement numériquement à cause des erreurs d'arrondis qui peuvent survenir lors de calculs sur des valeurs en virgule flottante. Un schéma de compression les utilisant sera donc implicitement irréversible (avec pertes). Pour pouvoir être appliquée dans un schéma réversible, les transformées réelles sont alors généralement factorisées en étapes de lifting (voir fin de section suivante) et approximées pour pouvoir être appliquées sur des valeurs entières afin d'être réversibles [BvdE92]. Dans le cas des ondelettes, la démarche est très bien décrite dans [DS98] et étendue dans [ST02]. Pour les transformées pouvant être modélisées par une matrice de projection carrée de déterminant 1, une méthode de factorisation a été proposée dans [HS01, Hao04], et reprise dans [LHX06, LXSH06] afin d'être appliquée dans le cas d'ondelettes orthonormales multibandes. Les approches de [DS98] et [Hao04] seront exploitées dans le chapitre 6.

Suit une présentation pratique de la transformée en cosinus discrète (DCT *Discret Cosine Transform*) et de la transformée en ondelettes discrète (DWT *Discrete Wavelet*

*Transform*), qui sont les plus utilisées en compression, suit ensuite un aperçu de quelques autres représentations, moins présentes dans la littérature, mais également employées en compression. La DCT et la DWT sont détaillées de façon théorique dans le livre de référence de Stéphane Mallat [Mal08]

**2.2.2.2.1 Transformée en cosinus** Issue des travaux de Joseph Fourier, la transformée en cosinus continue CCT permet de représenter une fonction continue  $f$ , définie sur un intervalle borné  $[a, b]$ , à l'aide d'une combinaison linéaire de cosinusoïdes. Elle est très connue en traitement du signal depuis que sa version discrète *DCT* a été proposée [ANR74]. Plusieurs bases de cosinus peuvent être construites à partir des séries de Fourier (appelées bases de cosinus I, II, III, IV, ...). Elles diffèrent par la représentation du signal  $\hat{f}$  qui est généré lorsque les cosinusoïdes sont étendues en dehors de l'intervalle  $[a, b]$ .

Pour un signal discret  $s$  de longueur  $N$  les DCT-II et DCT-IV se définissent comme :

$$\hat{s}_{\text{DCT-II}}(k) = \sum_{n=0}^{N-1} s(n) \cos\left(\frac{\pi(n + \frac{1}{2})k}{N}\right), \quad (2.19)$$

$$\hat{s}_{\text{DCT-IV}}(k) = \sum_{n=0}^{N-1} s(n) \cos\left(\frac{\pi(n + \frac{1}{2})(k + \frac{1}{2})}{N}\right), \quad (2.20)$$

pour  $k \in [0..N - 1]$ . Et leurs transformées inverses sont identiques à un facteur de normalisation près :

$$s(n) = \frac{2}{N} \sum_{k=0}^{N-1} \hat{s}_{\text{DCT-II}}(k) \cos\left(\frac{\pi(k + \frac{1}{2})n}{N}\right), \quad (2.21)$$

$$s(n) = \frac{2}{N} \sum_{k=0}^{N-1} \hat{s}_{\text{DCT-IV}}(k) \cos\left(\frac{\pi(k + \frac{1}{2})(n + \frac{1}{2})}{N}\right), \quad (2.22)$$

En compression d'images c'est la DCT-II qui est généralement utilisée (après une application bidimensionnelle de la transformée). Elle suppose que le signal généré par extension est  $2N$ -périodique et est pair en  $-\frac{1}{2}$  et en  $N - \frac{1}{2}$  et donc  $s(k) = s(-k - 1)$  ou encore  $s(k) = s(2N - k - 1)$ .

En compression audio (codecs MP3 ou AC-3 par exemple) c'est une version modifiée de la DCT-IV, la MDCT [PJB87], qui est plus souvent utilisée. Son utilisation pour la compression d'images a également été étudiée récemment [Mul09]. Elle est conçue pour être appliquée sur des portions du signal ayant un recouvrement. Ainsi, pour un signal  $s$ , les portions  $p_i$  de longueur  $2N$  seront telles que  $p_i$  aura ses  $N$  premiers échantillons en commun avec  $p_{i-1}$ , et ses  $N$  échantillons suivants en commun avec  $p_{i+1}$ . Elle s'exprime par :

$$\hat{p}_{i\text{MDCT}}(k) = \sum_{n=0}^{2N-1} p_i(n) \cos\left(\frac{\pi(n + \frac{1}{2} + \frac{N}{2})(k + \frac{1}{2})}{N}\right), \quad (2.23)$$



pour  $k \in [0..N - 1]$ . Et sa transformée pseudo-inverse :

$$\tilde{p}_i(n) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{p}_{i\text{MDCT}}(k) \cos\left(\frac{\pi\left(n + \frac{1}{2} + \frac{N}{2}\right)\left(k + \frac{1}{2}\right)}{N}\right), \quad (2.24)$$

pour  $n \in [0..2N - 1]$ . Elle a la propriété de fournir un nombre de coefficients transformés qui est deux fois moins important que le nombre d'échantillons. Ce nombre de coefficients réduits implique une erreur de reconstruction. Cependant cette erreur est supprimée par l'ajout de l'information obtenue par recouvrement des portions. Ce phénomène est connu sous le nom de TDAC (*time-domain aliasing cancellation*).

Contrairement à la DCT-II, la DCT-IV suppose une représentation impaire en  $N - \frac{1}{2}$ . Ainsi lorsque le bloc transformé n'est pas nul en ce point, une discontinuité est créée, et des coefficients de forte amplitude apparaissent. C'est pourquoi en compression on préfère appliquer une fenêtre avant d'utiliser la MDCT. Dans ce cas, pour que la MDCT puisse être inversible quelques opérations supplémentaires sont nécessaires.

**2.2.2.2 Transformée en ondelettes** La DCT et plus généralement les transformées de Fourier permettent de représenter un signal fini comme une somme de fonctions sinusoïdales (et donc de longueurs infinies) en supposant que celui-ci est périodique en dehors de ses bornes de représentation. Contrairement à ces approches, les ondelettes permettent une représentation localisée, temps/fréquence, du signal en le projetant sur des fonctions oscillantes à support borné, appelées ondelettes. Ces ondelettes sont issues d'une ondelette mère  $\Psi$  (ici normalisée  $\|\Psi\| = 1$  et centrée au voisinage de  $t = 0$ ) par décalage temporel (translatée de  $u \in \mathbb{Z}$  pour travailler sur des signaux discrets) et par dilatation (facteur d'échelle  $s \in \mathbb{N}$  pour obtenir une représentation discrète), et forment une famille :

$$\mathcal{W} = \left\{ \Psi_{u,s}(t) = \frac{1}{\sqrt{2^s}} \Psi\left(\frac{t-u}{2^s}\right) \right\}, \quad (2.25)$$

le facteur  $\frac{1}{\sqrt{2^s}}$  permettant de normaliser  $\|\Psi_{u,s}\| = 1$ . Pour  $t \in \mathbb{R}$ ,  $\Psi$  doit vérifier :

$$\int_{-\infty}^{+\infty} \Psi(t) dt = 0, \quad (2.26)$$

et les ondelettes discrétisées formant la famille  $\mathcal{W}_{\mathbb{Z}}$  :

$$\sum_{t \in \mathbb{Z}} \Psi_{u,s}(t) = 0. \quad (2.27)$$

La transformée en ondelettes à un instant  $u$  et à une échelle  $s$  d'un signal discret  $f$  est alors la projection de  $f$  sur  $\Psi_{u,s}$  :

$$\hat{f}_{\text{DWT}}(u, s) = \sum_{t \in \mathbb{Z}} f(t) \Psi_{u,s}(t), \quad (2.28)$$

et peut être vue comme un produit de convolution :

$$\hat{f}_{\text{DWT}}(u, s) = f \star \bar{\Psi}_s(u), \quad (2.29)$$

avec

$$\bar{\Psi}_s(t) = \frac{1}{\sqrt{2^s}} \Psi\left(\frac{-t}{2^s}\right). \quad (2.30)$$

La transformée en ondelettes permet ainsi de modéliser un filtrage multi-échelle.

Pour un signal discret  $f$  fini de longueur  $N$ , la  $\widehat{f}_{\text{DWT}}(u, s)$  sera appliquée pour  $u \in [0..N-1]$ . La DWT peut alors avoir quelques variantes en fonction de comment est modélisée l'extension du signal lorsque le filtrage nécessite des valeurs de  $f$  au delà de l'intervalle  $[0..N-1]$ . Par exemple, une symétrisation, une périodicité, ou des valeurs nulles peuvent être envisagées.

Afin de pouvoir reconstituer le signal original à partir d'une décomposition effectuée jusqu'à une échelle  $s = j$ , il est nécessaire de rajouter l'information contenue dans les échelles supérieures à  $j$ . La fonction d'échelle  $\Phi$  (aussi appelée ondelette père) est alors utilisée. Elle permet de représenter le signal  $f$  à différentes échelles et est définie par :

$$\Phi(t) = \int_1^{+\infty} \Psi_{0,s}(t) ds, \quad (2.31)$$

et permet également de générer une famille de fonctions d'échelles :

$$\mathcal{S} = \left\{ \Phi_{u,s}(t) = \frac{1}{\sqrt{2^s}} \Phi\left(\frac{t-u}{2^s}\right) \right\}. \quad (2.32)$$

La représentation du signal  $f$  pour l'ensemble des échelles supérieures à  $j$  (généralement la représentation basses fréquences de  $f$ ) peut alors s'écrire :

$$\widehat{f}_{\Phi}(u, s) = f \star \bar{\Phi}_s(u), \quad (2.33)$$

avec

$$\bar{\Phi}_s(t) = \frac{1}{\sqrt{2^s}} \Phi\left(\frac{-t}{2^s}\right). \quad (2.34)$$

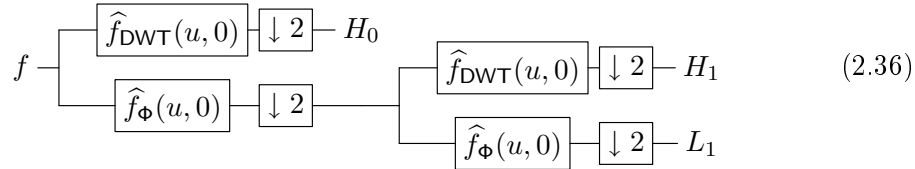
Afin d'être appliquée de façon réversible, la famille d'ondelettes discrètes  $\mathcal{W}$  doit être choisie de manière à former une frame. Il existe alors une frame duale  $\mathcal{W}^* = \{\Psi_{u,s}^*\}$  telle que :

$$f(t) = \sum_{u=0}^{N-1} \widehat{f}_{\Phi}(u, j) \Phi_j^*(t) + \sum_{s=0}^j \sum_{u=0}^{N-1} \widehat{f}_{\text{DWT}}(u, s) \Psi_{u,s}^*(t) \quad (2.35)$$

On parle d'ondelettes bi-orthogonales. Les  $\Phi$  et  $\Psi$  sont appelés filtres d'analyse, tandis que les  $\Phi^*$  et  $\Psi^*$  sont appelés filtres de synthèse. Si  $\mathcal{W}^* = \mathcal{W}$ , la famille d'ondelettes  $\mathcal{W}$  est orthogonale.

Jusqu'ici, la décomposition est redondante : les  $\widehat{f}_{\Phi}(u, 0)$  ainsi que chacun des  $j$  niveaux de décomposition  $\widehat{f}_{\text{DWT}}(u, j)$  nécessitent tous  $N$  coefficients. La décomposition peut cependant être modélisée comme une analyse en sous-bandes par cascade de bancs de filtres, et au même titre que celle-ci un sous échantillonnage peut être appliqué. La

bande  $\widehat{f}_{\text{DWT}}(u, 0)$ , associée aux hautes fréquences du signal selon  $\Psi$ , et la bande  $\widehat{f}_{\Phi}(u, 0)$ , associée aux basses fréquences, sont sous échantillonnées et la décomposition peut être ré-appliquée sur les basses fréquences :

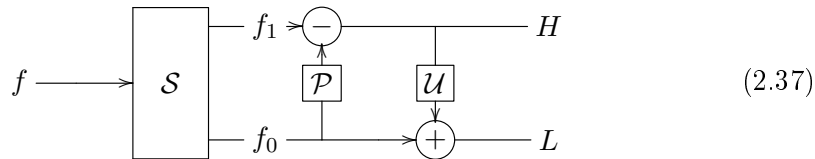


où  $\downarrow 2$  est un sous-échantillonnage uniforme d'un facteur 2. Les  $H_{i \in \{0..j\}}$  et  $L_j$  sont équivalents aux  $\widehat{f}_{\text{DWT}}(u, i)$  et  $\widehat{f}_{\Phi}(u, j)$  sous échantillonnés.

Il n'est pas toujours évident de calculer analytiquement les filtres de synthèse permettant de reconstruire le signal d'origine. Cependant, le *lifting scheme*, dont les concepts fondamentaux et des références à d'autres articles sont disponibles dans [Swe96a], permet de concevoir très facilement des filtres d'analyse et leur filtre dual. Le principe repose sur un partitionnement  $\mathcal{S}$  (comme *split*) des échantillons du signal  $f$  en plusieurs sous-ensembles entre lesquels sont successivement appliquées des étapes de lifting. Chaque étape de lifting consiste à prendre l'un des sous ensembles et à modifier chacun de ses échantillons par addition ou soustraction du résultat d'une opération n'impliquant que des échantillons des autres sous-ensembles.

### Exemple | 2.2.1 : SCHÉMA DE LIFTING SIMPLE

Pour illustrer, deux sous ensembles  $f_0$  et  $f_1$  sont pris.  $f_0$  contient les échantillons pairs et  $f_1$  les échantillons impairs.  $f_0$  est utilisé pour effectuer la prédiction  $\mathcal{P}$  (comme *predict*) des échantillons de  $f_1$  et leur soustraire cette valeur.  $f_1$  contient alors les erreurs de prédiction qui peuvent être considérés comme les hautes fréquences  $H$  sous-échantillonnées de  $f$ . Cette erreur de prédiction est ensuite utilisée pour effectuer une mise à jour  $\mathcal{U}$  (comme *update*) des valeurs de  $f_0$  afin de leurs supprimer ces hautes fréquences identifiées dans  $H$ .  $f_0$  peut être alors considéré comme les basses fréquences sous-échantillonné  $L$  de  $f$  :



Le schéma peut ensuite être réitéré sur les basses fréquences pour obtenir une décomposition dyadique, et sur les hautes fréquences pour obtenir une décomposition en paquets d'ondelettes.

La transformée inverse, et donc les filtres de synthèse, peuvent être obtenus à l'aide d'un schéma de lifting dual pour lequel les étapes de lifting sont appliquées dans l'ordre inverse, les additions annulées par des soustractions et les soustractions par des addi-

tions.  $f$  peut être récupéré en fusionnant la partition des sous-ensembles obtenus. Des schémas de lifting d'ondelettes, qui sont souvent très simples, sont couramment employés par les algorithmes de compression pour leur vitesse d'exécution et pour leur faible coût mémoire. En effet la transformée peut ainsi être effectuée dans le même espace mémoire que celui du signal initial. Le schéma très simple de l'exemple 2.2.1 sera utilisé dans le chapitre 5 afin de construire des bases de paquets d'ondelettes optimisées.

Les ondelettes ont une propriété intéressante pour la compression : si  $\Psi$  a ses  $P$  premiers moments nuls (régularité d'ordre  $p$ ), alors, la projection de tout polynôme d'ordre inférieur à  $P$  sur  $\Psi$  sera nulle :

$$\int_{-\infty}^{+\infty} x^p \Psi(x) dx = 0 \quad \forall p \in \mathbb{Z}/P \quad (2.38)$$

Ainsi plus  $\Psi$  sera régulière, plus la transformée générera des coefficients nuls (et réduira ainsi l'entropie).

Malheureusement, le nombre de moments nuls est limité par le support de l'ondelette. Plus la régularité sera élevée et plus le nombre d'oscillations, nécessaires à l'annulation des moments, le sera également. Daubechies a montré que pour obtenir une ondelette orthogonale à  $P$  moments nuls, le filtre associé à l'ondelette nécessite au moins  $2P$  coefficients. Or, plus la taille du support est élevée, plus l'ondelette risque de capter des discontinuités éloignées (polynômes d'ordre important). Dans ce cas, la projection aura des valeurs non nulles. Les ondelettes de Daubechies [Dau88] ont ainsi été créées pour posséder ces  $P$  moments nuls et avoir un support le plus compact possible (et donc  $2P$ ), et ensuite étendues afin d'apporter des propriétés complémentaires intéressantes [Dau93, CD93]. Parmi ces extensions, les symlettes, qui sont définies pour s'approcher le plus possible de filtres à phase linéaire, seront utilisées dans le chapitre 6.

Un compromis entre le nombre de moment nuls et la taille du support doit donc être fait. En compression sans perte d'images, on utilise souvent les « ondelettes spline 5/3 » de Le Gall [LGT88] (ce qui signifie que le filtre passe bas possède 5 coefficients et que le passe haut en possède 3) qui peut être implémenté par un schéma de lifting par spline (partitionnement pair/impair, prédiction des échantillons impairs comme appartenant aux droites passant par leurs deux voisins pairs (i.e. leur moyenne), mise à jour par ajout aux échantillons pairs de la moitié de la moyenne des erreurs commises sur leurs deux voisins impairs). Tandis qu'en compression avec pertes on préfère souvent les « ondelettes CDF 9/7 » de Cohen, Daubechies et Feauveau [CDF90].

Les ouvrages de référence d'Ingrid Daubechies [Dau92] et de Stéphane Mallat [Mal08] permettront au lecteur intéressé d'approfondir la théorie et les applications de cette transformée.

**2.2.2.3 Autres transformées** Bien que la DCT et la DWT soient les plus utilisées en compression, il en existe beaucoup d'autres, dont des techniques dérivées : la DPWT (*Discret Packet Wavelet Transform*), par exemple, consiste à réappliquer des

décompositions sur les bandes hautes fréquences de la DWT de façon à mieux décorréler les coefficients d'ondelettes.

La bien connue transformée en "S" ou "ondelette de Haar" a également eu une utilisation accrue en compression sans perte, et se voit encore utilisée aujourd'hui pour son support compact et donc la faible dépendance entre ses coefficients [WQ05]. Elle fut améliorée en compression sans perte d'images par une étape de prédiction permettant de mieux décorréler ses coefficients : la transformée "S+P" [SP93, SP96a]. Sa cousine, la WHT (*Walsh-Hadamard Transform*) a récemment été remise au goût du jours en compression sans perte comme avec le codeur LAR [DBM06] ou dans le standard JPEG-XR de Microsoft [TSS<sup>+</sup>08].

Partant de la constatation que les ondelettes avaient du mal à décorréler l'information au niveau des contours, une panoplie de décompositions orientées a été développée et vise à mieux s'adapter aux orientations locales pour mieux décorréler l'information. Même si elles sont peu (ou pas) utilisées en compression sans perte, elle fournissent des résultats intéressants en débruitage ou en compression avec pertes par exemple. Parmi ces décompositions on peut citer les contourlettes [DV05], les bandelettes [LPM05], ... et, bien sûr, l'une des plus théoriquement fondée et validée : la transformée en curvelettes [CD04, CDDY06].

Une autre transformation assez connue et qui a pu être utilisée en compression sans perte et presque sans perte d'images est la pyramide laplacienne [BA83, AABL97, ABA01]. Celle-ci est construite récursivement par la création d'une image basses fréquences sous-échantillonnée, et une image hautes fréquences pleine résolution, correspondant à l'erreur entre l'image basses fréquences interpolée pour fournir une image pleine résolution, et l'image originale. Ainsi, elle est redondante d'un facteur  $4/3$ , et n'est donc plus vraiment utilisée en compression.

On peut également citer la transformée de Radon discrète, connue sous le nom de transformée "Mojette" [Gué09, GN05] pour laquelle il y a eu quelques essais en compression sans perte d'images [KA08, DBN<sup>+</sup>04]. Cette transformée peut être appliquée de manière non redondante et réversible, même pour un nombre très limité de projections. Dans [KA08], l'image est projetée selon trois orientations d'angles très proches. Les résultats de projection ainsi obtenus pour chacune des orientations sont alors très corrélés entre eux et peuvent bénéficier d'une étape de réduction de l'information. Les auteurs utilisent une réorganisation des « bacs » d'accumulation des projections en sous-bandes, et exploitent un schéma prédiction intra-bande et inter-bande pour effectuer la décorrélation des données. En terme de taux de compression, ces premiers résultats n'apportent rien de plus que l'utilisation d'autres transformées, cependant elle peut permettre de faciliter l'encryption des données. Dans [DBN<sup>+</sup>04], la mojette est utilisée de manière redondante de sorte à mieux protéger les données pour la transmission.

Enfin, la KLT (*Karhunen-Loève Transform*), équivalente à la PCA (*Principal Component Analysis*) ou la POD (*Proper Orthogonal Decomposition*), est réputée pour son optimalité théorique en terme de débit-distorsions pour la compression de sources suivant une loi gaussienne. Pour les sources suivant une loi quelconque, cette transformée orthonormale permet d'obtenir la meilleure décorrélation possible, et minimise l'erreur quadratique moyenne (MSE) qu'il est possible d'obtenir avec un nombre de coefficients

restreints. Cependant, pour que ces optimalités soient vérifiées, il faut connaître les vecteurs propres engendrés par la décomposition (et donc les transmettre). Afin d'éviter de transmettre la base des vecteurs propres, une autre option est de fixer une base optimale pour une classe d'images par exemple (mais celle-ci sera généralement sous optimale pour une image particulière). La DCT est, dans le cas de processus de Markov gaussiens, réputée pour en être une bonne approximation des vecteurs propres de sa KLT, et c'est pourquoi elle est couramment employée en compression d'image. Le lecteur pourra se référer au chapitre 4 de [TM01] ainsi qu'aux articles [PTMO07, BSS09]. La KLT sera utilisée dans le chapitre 6 dans un schéma de compression basé sur un apprentissage multi-bases (pour une classe d'images spécifiques) afin de mieux décorréler l'information dans le domaine image (sur des blocs de pixels) ou dans le domaine transformé en ondelettes (sur des blocs de coefficients).

Il existe encore une panoplie de transformées, certaines efficaces pour la compression, d'autres moins, d'autres encore inconnues du domaine... Leur utilisation dépend principalement de leurs propriétés, de celles du signal et de celles souhaitées dans le schéma de compression (avec ou sans perte, représentation progressive, bien adaptée au système visuel humain, types d'artefacts générés en compression avec pertes, ...).

### 2.2.3 Quantification

Afin d'augmenter les performances de compression des signaux numériques, une solution consiste à imposer une réduction sur le nombre de symboles de l'alphabet en entrée du processus de codage entropique. Cette réduction peut s'effectuer par quantification scalaire : tout symbole de l'alphabet initial (ensemble des valeurs numériques possibles pour la représentation dans le domaine image ou transformé, ou encore pour les résidus de prédiction) est associé à un nouveau symbole dans un alphabet plus petit. Il est également possible d'utiliser une quantification vectorielle : une séquence de symboles en entrée du système est associée à l'index de la séquence la plus similaire dans un dictionnaire de séquences représentatives.

Dans ces deux situations, une opération surjective est effectuée et les symboles ou séquences originaux ne pourront pas être récupérés. Il faut alors chercher à optimiser les pertes d'informations.

#### 2.2.3.1 Quantification scalaire

Soient  $\mathcal{A}_{\mathcal{I}}$  l'alphabet initialement utilisé pour représenter le signal et  $\mathcal{A}_{\mathcal{Q}}$  l'alphabet après quantification, tels que  $\text{Card}(\mathcal{A}_{\mathcal{Q}}) < \text{Card}(\mathcal{A}_{\mathcal{I}})$ , l'opérateur de quantification peut s'écrire :

$$Q : \begin{array}{l} \mathcal{A}_{\mathcal{I}} \mapsto \mathcal{A}_{\mathcal{Q}} \\ x \rightarrow \tilde{x} \end{array}, \quad (2.39)$$

et l'opération de déquantification pseudo inverse :

$$Q^+ : \begin{array}{ccc} \mathcal{A}_Q & \mapsto & \mathcal{A}_I \\ \tilde{x} & \rightarrow & \hat{x} \end{array} . \quad (2.40)$$

Par la suite, la valeur approximée  $\tilde{x} = Q^+(Q(x))$  sera notée  $\tilde{Q}(x)$ .

### Exemple | 2.2.2 : UN QUANTIFICATEUR SCALAIRE SIMPLE

Si l'alphabet initial  $\mathcal{A}_I$  est l'ensemble des valeurs entières pouvant être représentées sur  $N$  bits,  $\text{Card}(\mathcal{A}_I) = 2^N$ . Un exemple simple de quantification scalaire est la réduction de la précision des échantillons du signal. On peut ainsi décider de supprimer un certain nombre de bits ( $k$ ) afin de réduire la taille de l'alphabet. L'alphabet des valeurs quantifiées  $\mathcal{A}_Q$  sera alors l'ensemble des valeurs entières pouvant être représentée sur  $N - k$  bits soit  $\text{Card}(\mathcal{A}_Q) = 2^{N-k}$ . La manière la plus simple d'effectuer cette réduction est un décalage de bits supprimant ceux de poids faible. L'opération pseudo-inverse peut consister, par exemple, à remplacer les bits supprimés par des bits nuls.

L'optimisation des erreurs de quantification peut s'effectuer à l'aide d'algorithmes comme Lloyd-Max [Max60, Llo82], par exemple, qui cherche la solution optimale au problème de minimisation de l'espérance de l'erreur quadratique due à la quantification scalaire :

$$\mathbb{E} \left[ \left( X - \tilde{Q}(X) \right)^2 \right]. \quad (2.41)$$

Cependant, utiliser ce type d'approche nécessite que le décodeur connaisse la loi de probabilité de  $X$ . Les paramètres de sa distribution (ou un dictionnaire de symboles) doivent donc être transmis, fixés, ou estimés de manière équivalente au codeur et au décodeur. Similairement, l'opération de déquantification  $\tilde{Q}$  peut aussi être optimisée.

Pour la compression d'images, il est souvent préférable d'appliquer la quantification après la décorrélation. Celle-ci est souvent propice à l'utilisation d'un simple quantificateur uniforme, qui consiste à proposer des valeurs quantifiées  $\{\tilde{Q}(\mathcal{A}_Q)\}$  uniformément réparties dans  $\mathcal{A}_I$ . Un second quantificateur très utilisé en compression est le quantificateur uniforme avec zone morte qui possède une zone de quantification plus importante autour de l'origine (afin de supprimer une partie du bruit) et reste uniforme partout ailleurs.

Notez que la représentation finie des valeurs réelles en virgule flottante, correspond également à une quantification scalaire. Cette discrétisation de l'ensemble des valeurs est souvent la cause d'accumulations d'erreurs de calculs, et de pertes d'informations lorsque des transformations numériques à valeurs réelles sont employées. La transformée de Fourier discrète ou celle en cosinus sont de bons exemples.

### 2.2.3.2 Quantification vectorielle

Bien qu'intéressante pour des problèmes tels que la compression, la quantification scalaire n'est pas optimale et peut être améliorée en considérant un alphabet initiale  $\mathcal{A}_{\mathcal{V}_T}$  dont chaque symbole est une combinaison (ou vecteur) de symboles de l'alphabet  $\mathcal{A}_T$ . La quantification vectorielle (QV) permet de réduire les distorsions (en terme d'erreur quadratique) introduites par une simple quantification scalaire uniforme.

Les systèmes de compression utilisant ce type de quantification sont souvent asymétriques : le codeur doit estimer les paramètres optimaux de quantification et les transmettre. Un dictionnaire optimal est ainsi construit. Chacun de ses éléments est un représentant de l'alphabet  $\mathcal{A}_{\mathcal{V}_T}$  qui sera utilisé par la suite pour reconstruire le signal. Les valeurs quantifiées correspondent alors à l'index du représentant du dictionnaire le plus similaire selon une métrique de distorsion donnée (cf. section suivante).

Il existe de nombreuses méthodes pour construire le dictionnaire. L'un des algorithmes les plus utilisés est celui de Linde, Buzo et Gray (LBG) [LBG80, GG91], qui est similaire à la méthode des  $k$ -means pour le clustering de données. LBG est une généralisation en dimension finie de l'algorithme Lloyd-Max. Il part d'un dictionnaire sous-optimal et cherche à l'améliorer. En partant d'une partition initiale en  $k$  clusters, le centroid de chacun des clusters est calculé et correspond à un élément du dictionnaire. Les individus d'apprentissage sont associés au centroid le plus proche (au sens de la norme  $L_2$ ) pour former  $k$  nouveaux clusters et le processus est réitéré jusqu'à la convergence (qui est assurée). La complexité pour trouver un dictionnaire optimal est exponentielle avec la dimension des vecteurs, et sa convergence dépend du dictionnaire initial. Il existe donc des techniques sous optimales, visant à offrir de bonnes performances pour un temps de calcul plus faible. En particulier, la quantification vectorielle algébrique (QVA) évite de devoir stocker un dictionnaire et son utilisation requière très peu de calculs, la technique la plus rapide étant de fixer une partition uniforme. Pour plus de détails sur la QVA, le lecteur est invité à consulter les travaux du Prof. J. M. Moureaux [Mou07].

Dans le chapitre 6, un algorithme similaire au LBG sera utilisé pour l'optimisation de plusieurs bases orthonormées dans le but de rendre l'énergie des coefficients transformés plus compacte et favoriser la scalabilité en qualité d'un type d'images particulier. La différence avec LBG étant que le représentant d'un cluster n'est plus un un vecteur mais une base (KLT du cluster) et que l'association au cluster ne s'effectue non plus par minimisation de la norme  $L_2$ , mais par maximisation de la compacité de l'énergie des coefficients obtenus après la transformée par KLT.

### 2.2.4 Optimisation débit-distorsion

Comme mentionné précédemment la numérisation des signaux pose le problème de trouver le débit binaire nécessaire à obtenir une distorsion donnée et inversement. De



même, lorsqu'une compression avec pertes est effectuée, les principaux objectifs peuvent être de compresser au maximum tout en respectant une certaine qualité, ou encore d'introduire le minimum de distorsions pour un taux de compression fixé.

### 2.2.4.1 Théorie débit-distorsion

La théorie débit-distorsion (voir chapitre 3 de [TM01] par exemple) cherche à apporter des solutions aux deux problèmes précédents. Pour un signal  $f$  donné, soit  $\mathbf{R}(Q)$  le débit pour une configuration des paramètres de compression  $Q$  (généralement le quantificateur) et  $\mathbf{D}(Q)$  la distorsion, alors la fonction débit-distorsion  $D(R_m)$  est définie de manière théorique comme la solution au problème de minimisation de la distorsion pour un débit maximal  $R_m$ , et la fonction distorsion-débit  $R(D_m)$  est définie comme celle du plus petit débit pour une distorsion maximale  $D_m$ . Ces problèmes d'optimisation sous contrainte peuvent s'écrire :

$$D(R_m) = \min_{Q \in \mathcal{Q}, \mathbf{R}(Q) \leq R_m} \mathbf{D}(Q), \quad (2.42)$$

$$R(D_m) = \min_{Q \in \mathcal{Q}, \mathbf{D}(Q) \leq D_m} \mathbf{R}(Q). \quad (2.43)$$

avec  $\mathcal{Q}$  l'ensemble des configurations possibles des paramètres de compression.

Si  $R$  et  $D$  sont des fonctions convexes, ces problèmes sont souvent résolus par une minimisation lagrangienne sans contrainte :

$$J_D(Q, \lambda) = \mathbf{D}(Q) + \lambda \mathbf{R}(Q), \quad (2.44)$$

$$J_R(Q, \lambda) = \mathbf{R}(Q) + \lambda \mathbf{D}(Q), \quad (2.45)$$

avec  $\lambda > 0$ .

Ainsi, une métrique de qualité et/ou de distorsion  $\mathbf{D}(Q)$  adaptée au contexte applicatif ou aux contraintes de calculs est utilisée afin d'optimiser le débit versus les distorsions.

### 2.2.4.2 Mesures de distorsion

De nombreuses métriques peuvent être utilisées pour mesurer les distorsions. La majorité de celles-ci se fondent sur la norme de Minkowski :

$$\text{Err} = \left( \sum_k |f(k) - \tilde{f}(k)|^\beta \right)^{1/\beta}. \quad (2.46)$$

La plus utilisée pour l'optimisation débit-distorsion est l'erreur quadratique moyenne (MSE : *Mean Squared Error*). Pour un signal de longueur  $N$ , elle s'exprime comme

$$\text{MSE} = \frac{1}{N} \sum_{k=0}^{N-1} |f(k) - \tilde{f}(k)|^2. \quad (2.47)$$

Les valeurs de la MSE étant souvent très élevées à bas débit, le PSNR (*Peak-to-peak Signal-to-Noise Ratio*) qui s'exprime en décibel (dB) est souvent préféré pour comparer des courbes de distorsions :

$$\text{PSNR}_{(\text{dB})} = 10 \log_{10} \frac{(2^p - 1)^2}{\text{MSE}}, \quad (2.48)$$

où  $p$  est le nombre de bits de précision des échantillons. Dans le cas de la compression sans perte, le PSNR est infini, et il décroît selon l'erreur. De plus le fait d'utiliser une valeur logarithmique permet d'observer le rapport de proportionnalité sur l'erreur quadratique entre plusieurs courbes de manière plus compacte :

$$\frac{\text{MSE}_B}{\text{MSE}_A} = 10^{\frac{\text{PSNR}_A - \text{PSNR}_B}{10}}. \quad (2.49)$$

La minimisation de l'erreur quadratique moyenne revient à minimiser les variations d'énergies entre le signal original et celui reconstruit. Cependant, cette variation énergétique n'est pas toujours adaptée, principalement lorsque le résultat de la compression est destiné à être apprécié par des humains (images, sons, ...) qui n'ont pas une sensibilité globale (moyenne des erreurs sur la totalité de l'image, du signal), mais plus localisée (variations spatiales, fréquentielles, temporelles...). Ainsi des métriques visant à simuler le comportement du cerveau humain face aux distorsions peuvent être utilisées pour optimiser la compression. Des modèles psycho-acoustiques sont employés pour la compression sonore, tels que celui utilisé par la norme internationale MP3 (MPEG-1/2 Audio Layer 3 : ISO/CEI 13818-3), et des modèles psycho-visuels (HVS *Human Vision System*) peuvent servir pour l'image et la vidéo. Pour la vision, les métriques d'évaluation de la qualité les plus connues sont la métrique JND (*Just Noticeable Difference*) de Lubin (Sarnoff Corporation) [Lub93, Lub95, Lub97, LF97], et la VDP (*Visible Difference Predictor*) de Daly [Dal93, Dal94].

Ces modèles psycho-visuels utilisent souvent les trois principales variations auxquelles l'œil humain est sensible : la sensibilité à l'intensité lumineuse, à la fréquence spatiale (contrastes), et au contenu (effets de masquage fréquentiel). Le masquage, également utilisé dans les modèles psycho-acoustiques, est un phénomène survenant en présence de plusieurs fréquences d'intensité différentes : localement, plus des oscillations de forte amplitude sont présentes moins l'humain perçoit celles de faible amplitude. Ceci peut survenir dans des zones texturées, ou proches de contours contrastés par exemple.

Il existe également d'autres modèles moins basés sur des études de la perception mais tout de même efficaces (en particulier sur des images naturelles) telles que celles proposées par Wang : SSIM (*Structural SIMilarity*) [WBSS04] et MS-SSIM (*MultiScale SSIM*) [WSB03] une extension multi-échelle plus robuste.

Ces métriques demandent souvent une complexité calculatoire assez importante, qui les rend difficilement utilisables avec des systèmes de compression rapides. Wang utilise tout de même SSIM dans [WLS07] pour effectuer l'optimisation d'un codage progressif et obtient des résultats intéressants.

### 2.2.4.3 Quantification scalaire par plans de bits

Les techniques d'optimisation qui viennent d'être mentionnées sont le plus souvent utilisées afin de maximiser les performances d'une compression avec pertes pour un débit ou une qualité ciblée. Cependant, lors d'une compression scalable en qualité pouvant éventuellement aller jusqu'au sans perte, ces approches sont difficilement envisageables à cause de leur coût calculatoire.

Comme indiqué dans 2.2.3.1, la quantification scalaire uniforme est souvent très efficace dans le domaine transformé. Le quantificateur de l'exemple 2.2.2 est uniforme sur  $\mathbb{Z}$ . Il peut être rendu symétrique autour de 0 en effectuant la quantification de  $x$  par décalage des bits de sa valeur absolue, et en réservant un bit supplémentaire pour transmettre son signe lorsque  $\tilde{x}$  est non nul. Il est ainsi mieux adapté à la distribution des coefficients transformés, généralement centrée en 0. Cette solution sous-optimale, mais très simple, est largement utilisée en compression par transformée. Elle permet d'obtenir facilement des quantificateurs uniformes imbriqués en effectuant un codage entropique plan de bit par plan de bit (en commençant par les bits de poids les plus forts) qui est souvent très efficace. Ainsi une scalabilité en qualité peut être aisément mise en place. Afin d'optimiser le rapport débit-distorsions, des techniques avancées telles que celles utilisées par les codeurs SPITH [SP96b] et EBCOT [Tau00] (voir 3.1.2.2) ou [ALM11] peuvent permettre d'organiser le flux compressé afin de transmettre en priorité les bits associés à un ensemble de coefficients importants. Enfin, pour minimiser les distorsions, l'opération de déquantification peut également tenter de s'adapter aux distributions statistiques locales des coefficients transformés [ALMSS09, ALM09].

## Conclusion

Les principaux outils nécessaires à la compression irréversible (avec pertes) ou réversible (sans perte) de signaux ou d'images viennent d'être présentés. Les algorithmes qui seront les plus détaillés dans le prochain chapitre, ainsi que ceux développés dans cette thèse s'appuieront essentiellement soit sur un schéma prédictif soit sur l'utilisation de la transformée en ondelettes (également couplée à une transformée par KLT pour la dernière contribution). Même si certains des algorithmes de compression, qui seront présentés, utilisent un modèle de codage implicite, notamment EZW et SPIHT qui s'appuient sur des propriétés de la transformée en ondelettes des images naturelles (c.f. 3.1.2.2), ceux-ci sont souvent sous-optimaux et se retrouvent couplés à un codage arithmétique afin d'obtenir de meilleures performances. Désormais, ce dernier est le codage entropique majoritairement utilisé, afin de maximiser les performances de compression. Même si il a l'inconvénient d'être plus lent que les VLC, avec la puissance de calcul des ordinateurs d'aujourd'hui et les optimisations existantes, il n'est plus très coûteux en temps de calcul. Le codage Golomb-Rice reste tout de même privilégié pour des applications de compression sans perte se voulant être très rapides ou pour une plus faible consommation sur du matériel embarqué. Concernant la quantification, celle-ci sera surtout impliquée dans des schémas scalables en qualité et donc par l'intermédiaire de quantificateurs scalaires imbriqués relatifs à un codage par plan de bit.

## Chapitre 3

# Algorithmes de référence en compression d'images

### Introduction

Les algorithmes les plus reconnus en compression d'images sont résumés dans ce chapitre. Cette vue d'ensemble s'oriente principalement autour des techniques pour la compression sans perte, presque sans perte et/ou scalable.

Dans une première section (sec. 3.1) sont présentées les références usuelles et les standards pour les images fixes (2D) qui ont principalement été définis pour la compression d'images naturelles. La seconde section (sec. 3.2) s'intéresse plus particulièrement à la compression des images produites en milieu hospitalier. Il y est constaté que les algorithmes précédents et en particulier les standards, bien que non spécifiques à l'imagerie médicale, sont généralement employés pour leur compression 2D. Plus typiques du milieu médical, des extensions de ces techniques pour les images volumiques sont ensuite présentées. Cette section s'achève sur un aperçu des algorithmes retenus pour la standardisation des échanges de résultats d'examens médicaux.

En complément, le lecteur est invité à consulter [NACM07, NACM08] et en particulier les chapitres [Bas08, CMNADB08, OZONA08, CMLCBT08] qui font respectivement un état de l'art sur la compression, la compression d'images médicales 2D, d'images médicales volumiques, l'évaluation subjective et objective des images médicales dégradées par la compression, et enfin le chapitre [GC08] qui traite de la place des standards dans ce contexte.

### 3.1 Références usuelles et standards de compression pour les images fixes

Cette section permet d'appréhender différentes méthodes de compression parmi les plus efficaces et la façon dont elles mettent en application les concepts présentés dans

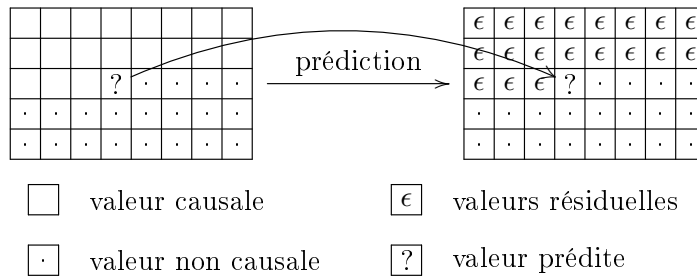


FIG. 3.1 – modèle DPCM

le chapitre précédent. Les techniques de compression sont souvent classées en fonction du type d'approche de décorrélation employé. Les algorithmes usuels de compression prédictifs et par transformées, ainsi qu'une solution combinant les deux processus de décorrélation seront donc présentés séparément. Pour finir, les méthodes permettant d'effectuer une compression presque sans perte seront décrites.

L'étude étant principalement orientée vers la compression sans perte et presque sans perte, les transformées redondantes telles que la décomposition pyramidale laplacienne, la transformée en curvelettes ou encore les décompositions parcimonieuses dans des dictionnaires redondants ne seront pas abordées ici. Cette redondance est souvent défavorable en compression sans ou presque sans perte à cause du bruit présent dans les données. Ces techniques deviennent plus efficaces en compression avec pertes, une fois que la quantification a supprimé une bonne partie du bruit.

### 3.1.1 Compression prédictive

L'approche prédictive se prête particulièrement bien à la compression sans perte et presque sans perte d'images. Dans ces deux situations, elle est souvent plus efficace qu'avec des approches par transformées, mais les fichiers compressés produits offrent souvent moins de propriétés intéressantes pour une transmission interactive. Certains modèles prédictifs sont très simples et peuvent donc être exécutés de manière très rapide. D'autres sont beaucoup plus complexes, mais permettent d'obtenir des taux de compression meilleurs, plus utiles pour de l'archivage.

#### 3.1.1.1 Modèle DPCM

Les trois algorithmes prédictifs les plus connus en compression d'images sont celui utilisé pour la compression sans perte dans le standard JPEG, son successeur LOCO-I qui fut choisi pour normaliser JPEG-LS, et enfin CALIC qui est souvent employé comme algorithme de référence. Ces codeurs sont appliqués en une seule passe sur l'image, ligne après ligne et pixel par pixel. Cette approche est souvent généralisée sous le nom de codage DPCM (*Differential Pulse Code Modulation*). Le schéma de fonctionnement peut se décrire à un niveau pixelique et être appliqué séquentiellement (FIG. 3.1).

**3.1.1.1.1 JPEG sans perte** « JPEG sans perte » est le nom généralement attribué au mode sans perte défini dans le standard JPEG (ISO/IEC International Standard 10918-1 – ITU-T Recommendation T.81). Il n'est pas basé sur la DCT, contrairement au mode de compression avec pertes, et utilise un modèle DPCM dans lequel, pour rester causal, chaque pixel est prédit à l'aide de ses voisins précédemment encodés.

Si  $x_i$  est le pixel à prédire, JPEG utilise le motif suivant :

$c$	$b$
$a$	$x_i$

Le standard propose 7 schémas pour effectuer la prédiction. Le schéma retenu étant utilisé sur la totalité de l'image et spécifié dans l'en-tête du fichier.

n°	prédiction
0	pas de prédiction
1	$\hat{x}_i \triangleq a$
2	$\hat{x}_i \triangleq b$
3	$\hat{x}_i \triangleq c$
4	$\hat{x}_i \triangleq a + b - c$
5	$\hat{x}_i \triangleq a + ((b - c)/2)$
6	$\hat{x}_i \triangleq b + ((a - c)/2)$
7	$\hat{x}_i \triangleq (a + b)/2$

Après cette prédiction, un codage entropique de type Huffman ou arithmétique est utilisé pour effectuer la compression (pour plus de détails, se référer à l'annexe H du standard).

Ce modèle de compression très simple n'offre pas des taux de compression fantastiques. La demande croissante pour un standard de compression sans perte efficace a conduit à la normalisation de JPEG-LS.

**3.1.1.1.2 JPEG-LS (LOCO-I)** Le standard JPEG-LS (ISO/IEC International Standard 14495-1 – ITU-T Recommendation T.87), principalement normalisé pour pallier aux faibles performances du JPEG sans perte, s'appuie sur l'algorithme LOCO-I (*Low Complexity Lossless Compression for Images*) [WSS96, WSS00]. Cet algorithme fut retenu pour avoir l'un des meilleurs compromis entre rapidité et performances de compression. Il utilise une modélisation contextuelle simple pour effectuer une prédiction et un codage entropique adaptatif basé sur des VLC. Il reste ainsi rapide et efficace tout en s'adaptant aux variations statistiques de l'image, et peut permettre d'effectuer une compression presque sans perte (cf. 3.1.4).

Comme JPEG sans perte, l'approche DPCM s'effectue en une seule passe. Pour chaque pixel, ses voisins précédemment encodés sont tout d'abord utilisés pour effectuer une détection rudimentaire de contours. Une prédiction non linéaire s'appuyant sur cette détection est ensuite appliquée.

Si  $x_i$  est le symbole à encoder, JPEG-LS utilise le motif suivant :

$c$	$b$	$d$
$a$	$x_i$	

La valeur de prédiction est :  $\hat{x}_i \triangleq \begin{cases} \min(a, b) & \text{si } c \geq \max(a, b), \\ \max(a, b) & \text{si } c \leq \min(a, b), \\ a + b - c & \text{sinon.} \end{cases}$

Un modèle contextuel est ensuite sélectionné à l'aide des valeurs quantifiées des gradients entre les pixels voisins :

$$\begin{aligned} g_1 &= d - b \\ g_2 &= b - c \\ g_3 &= c - a \end{aligned}$$

Dans l'article où LOCO-I fut présenté [WSS96], l'implémentation utilisait 4 valeurs de gradients quantifiés qui formaient 1094 contextes pour des images en niveaux de gris sur 8 bits. Dans JPEG-LS, ces 3 valeurs permettent d'obtenir au maximum 365 contextes.

Ces modèles contextuels sont utilisés afin d'estimer de manière adaptative la distribution de probabilités des résidus  $\epsilon_i$ , conditionnellement à la configuration du voisinage de prédiction. Le contexte sélectionné pour  $x_i$  est utilisé pour corriger le biais de prédiction (erreur moyenne ayant tendance à se produire dans une même configuration), et pour estimer efficacement les paramètres du codage entropique.

Les résidus de prédiction  $\epsilon = x_i - \hat{x}_i \bmod \alpha$ , avec  $\alpha$  la dimension de la plage de niveaux de gris de l'image, appartiennent à l'intervalle  $[-\lfloor \alpha/2 \rfloor .. \lfloor \alpha/2 \rfloor]$ . Le codage s'effectue en supposant que pour chaque contexte les valeurs de  $\epsilon$  suivent une distribution Laplacienne centrée en 0. Afin de se rapprocher d'une entrée suivant une distribution géométrique à valeurs entières, les  $\epsilon$  sont associés aux valeurs  $0 \leq M(\epsilon) \leq \alpha - 1$  telles que :

$$M(\epsilon) = \begin{cases} 2\epsilon & \epsilon \geq 0, \\ 2|\epsilon| - 1 & \epsilon < 0. \end{cases}$$

Il est ainsi possible de les encoder à l'aide d'un codeur Golomb-Rice adapté. Le paramètre  $k$  du codeur (cf. section 2.1.2.1.2) est estimé et mis à jour régulièrement pour chacun des contextes afin d'améliorer l'adaptation aux variations locales des images. Afin de corriger un biais introduit par  $M(\epsilon)$  qui a tendance à générer un codage plus court pour les valeurs négatives que pour les valeurs positives,  $M(-1 - \epsilon)$  est encodé lorsque le centre de la distribution des bits est plus proche de -1 que de 0.

Enfin, pour contourner le problème de redondance du codage VLC, qui nécessite au minimum un bit par symbole, un codage RLE est utilisé dans les zones uniformes. Plus précisément, lorsqu'une zone uniforme est détectée ( $a = b = c$ ), la longueur  $L$  (qui peut être égale à 0) de la séquence de pixels égaux à  $a$  est codée en s'appuyant sur une compression Golomb-Rice également.

**3.1.1.1.3 CALIC** CALIC (*Context Based, Adaptive, Lossless Image Coding*) [WM97] sélectionne tout d'abord un prédicteur à l'aide de quelques pixels déjà connus. Il effectue ensuite sa prédiction et la rend adaptative en utilisant une modélisation des erreurs précédemment commises (correction contextuelle des biais de prédiction). Enfin, il effectue la compression à l'aide d'un codeur arithmétique adaptatif. CALIC propose également un mode de codage arithmétique ternaire pour les zones de l'image où il ne détecte au plus que deux niveaux de gris distincts : 1 symbole est utilisé pour chacun des deux niveaux de gris, et 1 symbole permet de signaler la présence d'un niveau de gris différent et donc la nécessité de repasser en mode prédictif.

Si  $x_i$  est le symbole à encoder, CALIC utilise le motif suivant :

		$nn$	$nne$
	$nw$	$n$	$ne$
$ww$	$w$	$x_i$	

avec  $n$  (*north*),  $w$  (*west*),  $ne$  (*northeast*),  $nw$  (*northwest*),  $nn$  (*north-north*),  $ww$  (*west-west*) et  $nne$  (*north-northeast*).

Le gradient local en  $x_i$  est estimé horizontalement par  $d_h = |w - ww| + |n - nw| + |n - ne|$ , et verticalement par  $d_v = |w - nw| + |n - nn| + |ne - nne|$ , et est utilisé pour détecter l'orientation et l'amplitude des contours afin d'effectuer une prédiction de  $x_i$  adéquat (GAP : *Gradient-Adjusted Prediction*) :

$$\hat{x}_i \triangleq \begin{cases} w & \text{si } d_h - d_v > \alpha, & \% \text{ contour horizontal très marqué} \\ (X + w)/2 & \text{si } \alpha \geq d_h - d_v > \beta, & \% \text{ contour horizontal} \\ (3X + w)/4 & \text{si } \beta \geq d_h - d_v > \gamma, & \% \text{ contour horizontal peu marqué} \\ n & \text{si } d_h - d_v < -\alpha, & \% \text{ contour vertical très marqué} \\ (X + n)/2 & \text{si } -\alpha \leq d_h - d_v < -\beta, & \% \text{ contour vertical} \\ (3X + n)/4 & \text{si } -\beta \leq d_h - d_v < -\gamma, & \% \text{ contour vertical peu marqué} \\ X & \text{sinon.} \end{cases} \quad (3.1)$$

avec  $X = (w + n)/2 + (ne - nw)/4$ , et  $\alpha > \beta > \gamma$  trois constantes fixées respectivement à 80, 32 et 8 dans la publication [WM97] pour des images 8 bits. Pour les images avec une précision plus importante, les valeurs de  $d_h$  et  $d_v$  sont réajustées pour pouvoir être utilisées dans cette équation.

Cette prédiction  $\hat{x}_i$  est ensuite corrigée grâce à une modélisation contextuelle adaptative du biais de prédiction. Le contexte de correction est sélectionné en combinant la classification du motif textuel, formé par les pixels voisins ( $n$ ,  $w$ ,  $nw$ ,  $ne$ ,  $nn$  et  $ww$ ), à l'erreur  $\epsilon_{i-1}$ , commise sur le pixel précédent, et à l'activité locale  $d_h$  et  $d_v$ . La correction correspond alors à :  $\hat{x}_i \leftarrow \hat{x}_i + f(n, w, nw, ne, nn, ww, \epsilon_{i-1}, d_h, d_v)$ .

L'erreur  $\epsilon_i = x_i - \hat{x}_i$  est ensuite compressée par un codage arithmétique dont les paramètres sont estimés de manière adaptative, conditionnellement au contexte de correction.



Ce schéma est relativement similaire à JPEG-LS, cependant le prédicteur et sa sélection sont plus complexes, la correction des biais également, et enfin le codage arithmétique est beaucoup plus gourmand que le codage Golomb-Rice mis en place par LOCO-I. De par cette conception, CALIC demande plus de calculs que LOCO-I qui fut préféré pour la normalisation de JPEG-LS pour son rapport taux de compression/rapidité. CALIC reste cependant une référence, offrant des taux de compression meilleurs.

**3.1.1.1.4 Autres algorithmes DPCM** De nombreuses études ont été effectuées autour de ce modèle de prédiction DPCM. Les algorithmes les plus performants utilisent une modélisation statistique contextuelle, comme LOCO-I et CALIC. Bien que ce dernier soit toujours une référence, quelques algorithmes sont apparus depuis et permettent une compression plus efficace. La plupart de ces algorithmes utilisent la méthode des moindres carrés pour construire des prédicteurs adaptatifs [LO01, MT01, YDD03, KL05] mais nécessitent plus de calculs.

D'autres algorithmes DPCM s'effectuent en plusieurs passes et sont donc beaucoup plus gourmands en temps de calcul. L'algorithme TMW [MT97, MT98], par exemple, effectue une première passe pour analyser l'image et générer les paramètres d'un modèle statistique. Ce modèle est ensuite utilisé pour effectuer un codage prédictif plus optimal lors d'une seconde passe. Les paramètres du modèle sont d'abord transmis, suivis par l'encodage de l'erreur de prédiction. Cette technique permet d'obtenir des résultats meilleurs que CALIC (entre -0.1 bpp et -0.2 bpp sur les résultats présentés). FMP [AAB02] obtient des taux de compression similaires à l'aide d'une méthode complexe basée sur la logique floue. Celle-ci nécessite une première étape d'apprentissage qui permet d'optimiser un nombre fixe de prédicteurs  $\vec{\phi}_m$ . Pour chaque pixel devant être estimé, un nouveau prédicteur  $\vec{\phi}(n)$  est conçu à l'aide d'une combinaison linéaire des  $\vec{\phi}_m$  dont les coefficients sont estimés par la méthode des moindres carrés sur le voisinage causal. Pour cette algorithme, c'est une matrice d'apprentissage (prédicteurs retenus) qui est transmise, suivie par les données prédites. Enfin, MRP<sup>1</sup> [MMI00, MOU05] cherche à optimiser des prédicteurs pour minimiser l'entropie, au lieu de minimiser l'erreur quadratique moyenne. Cet algorithme est vraiment très coûteux, mais obtient quasiment les meilleurs taux de compression : sur l'image de Lena  $512 \times 512$ , il<sup>2</sup> nécessite plus de 2'30 minutes<sup>3</sup> avec les paramètres par défaut pour obtenir une compression à 4.300 bpp et 14'30 minutes avec l'option '-o' (qui permet d'aller plus loin dans l'optimisation) pour une compression à 4.280 bpp, contre 0.02 secondes avec JPEG-LS<sup>4</sup> pour une compression à 4.607 bpp.

Enfin, il existe aussi des approches DPCM qui cherchent à fournir une représentation scalable en qualité. Elles s'appuient sur une compression DPCM avec pertes de l'image (couche de base), suivie par le codage successif de une ou plusieurs couches de

<sup>1</sup><http://itohws03.ee.noda.sut.ac.jp/~matsuda/mrp/>

<sup>2</sup>version 0.5 du logiciel MRP<sup>1</sup>

<sup>3</sup>résultats obtenus sur un Intel(R) Core(TM)2 Duo CPU E6750 @ 2.66GHz

<sup>4</sup><http://www.hpl.hp.com/loco/>

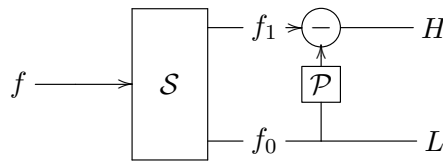


FIG. 3.2 – schéma de lifting pour la prédiction hiérarchique

raffinement de l'information, permettant d'améliorer la qualité de restitution de l'image. Ces algorithmes s'effectuent nécessairement en plusieurs passes (au moins une passe par couche de qualité). La restitution peut tenter de minimiser l'erreur quadratique moyenne pour chaque couche [RR01], ou proposer plusieurs couches de qualités presque sans perte jusqu'au sans perte [AMSS02, AMSS05].

### 3.1.1.2 Modèle prédictif hiérarchique

La plupart des variantes des codeurs prédictifs DPCM en une passe ont pour gros avantage d'être efficaces et de nécessiter peu de mémoire pour effectuer les calculs de prédiction. Seules les quelques lignes précédentes de l'image et les modèles des contextes nécessaires à la prédiction et au codage ont besoin d'être stockés. Les algorithmes peu complexes (comme LOCO-I) seront ainsi préférés pour des systèmes embarqués, ou pour une transmission sans perte rapide. Les algorithmes DPCM peu coûteux en une passe ne sont pas scalables et ceux plus coûteux en plusieurs passes peuvent être rendus scalables en qualité, mais aucun d'entre eux n'est scalable en résolution (à cause de l'approche DPCM).

Afin d'obtenir cette fonctionnalité, l'approche prédictive hiérarchique par interpolation (HIP : *hierarchical interpolated prediction*) peut être utilisée. Les HIP utilisent une grille pour sous échantillonner l'image et ainsi obtenir une représentation basse résolution. Celle-ci sert à prédire, généralement par interpolation, les pixels manquants. Le procédé est ensuite réappliqué sur la basse résolution. L'image est ainsi décomposée sous une forme pyramidale qui permet une compression progressive de sa résolution. Cette décomposition est assez proche d'une transformée en ondelettes par lifting mais ne comporte qu'une étape prédictive comme illustré dans la FIG. 3.2. Les algorithmes l'utilisant restent donc des codeurs prédictifs.

HINT (Hierarchical INTerpolation) [RVvDP88] est l'un des premiers algorithmes à fonctionner ainsi. Il est conçu sur une grille quinconce, avec une prédiction utilisant des filtres d'interpolation linéaires non séparables. L'une de ses extensions, IHINT [AAB97], conserve une grille rectangulaire afin d'effectuer une prédiction entrelacée horizontale puis verticale pour chaque niveau de résolution. Cette approche est similaire à une décomposition en ondelettes dyadique (voir FIG. 3.4) mais reste prédictive. L'étape horizontale est illustrée dans la FIG. 3.3. L'étape suivante consiste à transposer la prédiction sur la bande basse résolution « $L$ » et sur la bande des résidus afin d'obtenir quatre sous-bandes : trois contenant des résidus de prédiction horizontaux, verticaux et

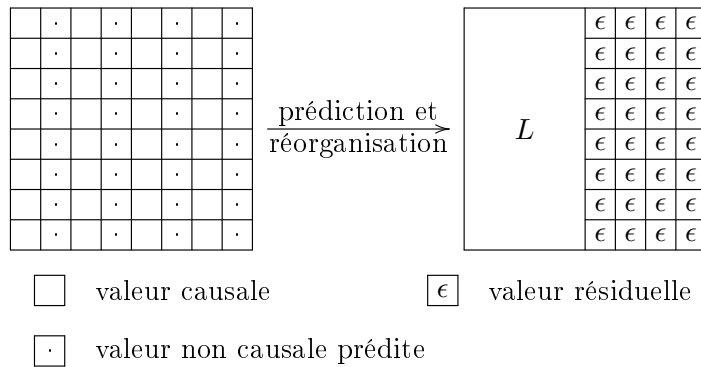


FIG. 3.3 – exemple de modèle de prédiction hiérarchique

diagonaux, et une contenant l'image sous échantillonnée, sur laquelle on peut ensuite réappliquer les deux étapes précédentes.

### 3.1.2 Compression par transformée

La compression par transformée se prête mieux à la compression d'image scalable ou avec pertes. Les transformées en cosinus et en ondelettes, précédemment introduites (2.2.2.2), sont les deux transformées les plus utilisées. Comme la majorité des transformées basées sur le domaine spectral, leur usage résulte de l'observation que les images naturelles tendent à avoir la majorité de leur énergie concentrée autour des basses fréquences. Ainsi, les approches spectrales réduisent l'entropie de la source en compactant une majorité de l'information sur les coefficients associés à ces basses fréquences. Elles sont principalement appliquées aux images de manière séparable : transformation des lignes de l'image, puis transformation verticale des coefficients obtenus par l'application horizontale précédente.

#### 3.1.2.1 DCT

La Transformée en Cosinus Discret (DCT-II) est l'une des transformées les plus répandues en compression d'images car elle est exploitée dans ce domaine depuis les années 80. Elle est surtout employée en compression avec pertes, et notamment par le standard JPEG adopté en 1992 (ISO/IEC International Standard 10918-1 – ITU-T Recommendation T.81). Cette transformée a su montrer ses faiblesses au fil du temps. Le principal reproche va aux artefacts générés lors d'une compression avec pertes. De plus, les fonctions sinusoïdales étant à valeurs réelles, le calcul de la DCT génère des pertes d'informations numériques. Il est tout de même possible d'utiliser la factorisation en étapes de lifting [HS01, Hao04, WDJ06, Abh07] pour l'approximer et pouvoir ainsi l'utiliser un schéma de compression sans perte. Wang *et al.* [WWJ<sup>+</sup>08] s'en sert de façon efficace afin d'effectuer un compression scalable *lossy to lossless* d'images naturelles, et obtient des résultats équivalents à JPEG-2000 (c.f. 3.1.2.2).

Puisque l'énergie des images est le plus souvent concentrée sur les basses fréquences, les algorithmes de compression avec pertes ont tendance à quantifier plus fortement les hautes fréquences, de manière à obtenir un nombre de coefficients nuls plus importants et ainsi à réduire l'entropie. L'utilisation d'un modèle psycho-visuel peut également servir à supprimer les composantes fréquentielles non ou peu perceptibles (masquage).

Dans le standard JPEG, comme dans beaucoup de codec se basant sur la DCT, l'image est découpée en blocs et la DCT est appliquée sur chacun d'entre eux. Les coefficients sont ensuite quantifiés, avant d'être compressés. Un parcours en «zig-zag» des coefficients d'un bloc permet de les ordonner des plus basses fréquences aux plus hautes. Un codage RLE sert ensuite à réduire les longues séquences de coefficients identiques, et un marqueur de fin de bloc à indiquer que tous les coefficients suivants sont nuls. Enfin les données sont compressées à l'aide d'un codage Huffman. Le standard permet également l'utilisation d'un codage arithmétique.

### 3.1.2.2 Ondelettes

Ces dernières années, la transformée phare en compression d'images est la transformée en ondelettes discrète (DWT *Discrete Wavelet Transform*). Elle est notamment utilisée par le standard JPEG-2000 (ISO/IEC International Standard 15444-1 – ITU-T Recommendation T.800). Elle est réputée pour sa simplicité d'utilisation (en particulier grâce au *lifting scheme*), ses bonnes propriétés de décorrélation, sa représentation multi-résolution, etc. Elle a connu un nombre considérable d'applications en compression avec et sans perte d'images. Les ondelettes aidant, les algorithmes s'appuyant sur cette transformée proposent généralement un codage scalable en résolution et/ou en qualité.

La DWT a également inspiré un bon nombre d'autres transformées utilisées en compression ou en débruitage d'images (les paquets d'ondelettes, les curvelettes, les bandelettes, les contourlettes, ...) visant à apporter des propriétés complémentaires (meilleure décorrélation des coefficients hautes fréquences, réduction de l'amplitude des coefficients aux abords des discontinuités/contours, ...).

La DWT bidimensionnelle est souvent effectuée à l'aide de filtres 2D séparables qui permettent une implémentation rapide. Ainsi, pour chaque niveau de décomposition, elle est appliquée sur les lignes de l'image, générant une représentation basses fréquences et des coefficients hautes fréquences horizontaux. Elle est ensuite de nouveau appliquée sur les colonnes des deux sous-ensembles (sous-bandes) ainsi obtenus. Cette approche est appelée décomposition dyadique ou décomposition en bandes par octave. Elle peut être modélisée par le schéma de la FIG. 3.4 où HDWT et VDWT sont respectivement la décomposition en ondelettes horizontale et verticale,  $LL_k$  la représentation basse résolution de l'image lors de l'étape précédente (ou l'image complète pour  $LL_0$ ),  $L_{k+1}$  celle de basse résolution et  $H_{k+1}$  les coefficients hautes fréquences après filtrage horizontal.  $LL_{k+1}$  correspond alors à la version basse résolution de  $LL_k$ ,  $LH_{k+1}$  aux coefficients hautes fréquences verticaux,  $HL_{k+1}$  à ceux horizontaux et  $HH_{k+1}$  aux diagonaux.

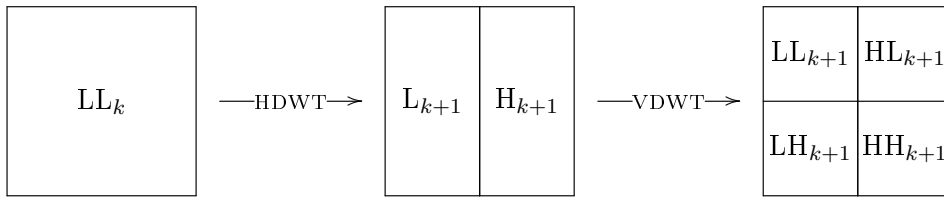


FIG. 3.4 – Décomposition dyadique

Cette décomposition est la plus courante mais il en existe d'autres utilisant des filtres pouvant être basés sur des grilles d'échantillonnage différentes (quinconces, hexagonales, ...) et/ou non séparables. Elles sont la plupart du temps implémentées par schémas de lifting.

Introduit par Shapiro, EZW [Sha93] (*Embedded Zerotree Wavelet*) est l'un des premiers codeurs basés ondelettes à se démarquer. Il fait parti des références incontournables de la compression moderne et marque le début de toute une génération d'algorithmes qui se sont inspirés de son concept comme, en particulier, SPIHT [SP96b] (Set Partitioning in Hierarchical Trees) de Said et Pearlman. Ce dernier, plus performant que EZW, est encore très utilisé comme algorithme de référence pour l'évaluation d'autres méthodes et son approche sert souvent de base à d'autres algorithmes. Ces deux techniques effectuent un codage progressif de manière à optimiser le rapport débit/distorsion de l'image reconstruite quel que soit le point d'arrêt dans le flux de données. EZW et SPIHT sous leur forme originelle tendent ainsi à minimiser l'erreur quadratique moyenne.

Suit une description de ces deux algorithmes et des principaux qui ont suivi. Ils peuvent être classés en deux grandes familles d'approches (inter- et intra-bande) et une troisième plus petite (approches mixtes). Elles se distinguent par les dépendances exploitées pour effectuer la compression des coefficients de la DWT. Ces algorithmes sont principalement conçus pour la décomposition dyadique, mais peuvent être utilisés (ou étendus) avec d'autres décompositions en sous-bandes (DWT quinconce, DCT par blocs, ...).

**3.1.2.2.1 Codage inter-bande** Cette première génération d'algorithmes essaie d'améliorer la compression en utilisant la redondance d'information entre les coefficients d'ondelettes au travers des différentes échelles. Ils exploitent également la propriété croisée des images à contenir plus d'informations dans les basses fréquences, et celle des ondelettes à compacter cette énergie des basses fréquences. Ainsi l'énergie des coefficients hautes fréquences d'une certaine région spatiale a tendance à décroître avec l'échelle à laquelle est associée la sous-bande.

**EZW** EZW [Sha93] effectue une succession d'approximations par quantifications imbriquées afin de permettre un codage progressif (cette quantification est simplifiable

en ne considérant que les différents plans de bits). L'information est organisée dans une Zerotree spécifiant la carte de signifiante des coefficients. Le Zerotree représente l'arbre hiérarchique entre un coefficient d'ondelette et les coefficients correspondant à la même zone de l'image dans la sous-bande de même orientation fréquentielle du niveau de résolution supérieur. Un nœud du Zerotree permet d'indiquer si tous les coefficients qui appartiennent au sous-arbre sont inférieurs au seuil de quantification, et le cas échéant, évite de les coder. Afin d'effectuer un codage progressif, ce Zerotree est d'abord généré pour un certain seuil de quantification (plan de bits de poids fort), puis est progressivement raffiné pour chaque seuil de quantification plus fin (plans de bits suivants). Le codage se fait alors plan de bit par plan de bit, à l'aide d'un parcours en largeur de l'arbre (i.e. résolution par résolution). Cette représentation effectue une compression implicite de l'information transformée.

**SPIHT** Tout comme EZW, SPIHT [SP96b, SP96a] exploite les similarités spatiales de répartition énergétique des coefficients d'ondelettes entre les différentes échelles. Cependant l'organisation en arbre hiérarchique est utilisée pour effectuer un tri partiel des coefficients de manière à coder en priorité ceux de plus forte énergie. Le flux de sortie contient alors l'information générée par l'algorithme de tri du codeur, nécessaire au décodeur pour pouvoir réorganiser (dé-trier) les coefficients, ainsi que les valeurs des coefficients (bits et signe).

Comme EZW, l'algorithme suppose une décroissance énergétique au travers des résolutions de l'arbre hiérarchique. Ainsi des coefficients non significatifs dans les basses fréquences ont de fortes chances d'avoir des enfants non significatifs dans les hautes fréquences, et ceux possédant une énergie importante ont généralement des enfants possédant moins d'énergie. Ces hypothèses sont utilisées par SPIHT afin de réduire le débit binaire du tri partiel. Celui-ci réorganise les coefficients d'une branche en fonction de leur bit de poids le plus fort (réduction de l'erreur quadratique) à l'aide d'un parcours en profondeur des nœuds possédant un descendant significatif dans l'arbre hiérarchique (contrairement à EZW qui effectue un parcours en largeur), et ce plan de bit par plan de bit.

L'algorithme sera détaillé davantage dans le chapitre 6.

**Successeurs de SPIHT** SPIHT a connu un grand succès dans le domaine de la recherche, notamment en imagerie médicale. Il a été étendu :

- pour la compression d'images médicales volumiques [XWYP98], et amélioré afin de faciliter l'accès aléatoire [CKP04] ;
- pour le codage de région d'intérêt [AHS02, AHS04] ;
- pour la télémédecine afin de pouvoir transmettre des images à différentes résolutions (ce que ne permet pas SPIHT) et qualités [HCL03] ;
- pour la compression d'images médicales à l'aide d'un modèle psychovisuel humain [PR07] ;
- et pour la compression presque sans perte [YCP05, YP06].

**PROGRESS** PROGRESS [CPS05] est un algorithme de Cho, Pearlman et Said qui vise à être efficace et très rapide. Pour cela, les propriétés inter-bandes sont encore exploitées mais dans un contexte de scalabilité en résolution uniquement. Le codage par plan de bits, qui devient rapidement couteux en temps de calcul, est alors éliminé. Afin de ne pas utiliser de codeur entropique, l'arbre hiérarchique est exploité pour spécifier le nombre de bits nécessaires à la représentation des coefficients enfants (par blocs de 4) en fonction de l'amplitude de leur coefficient parent. Un fois ce nombre de bits connu, les 4 coefficients enfants sont transmis. Ce processus est itéré sous-bande par sous-bande, résolution par résolution. Les résultats en terme de compression sont assez similaires à SPIHT pour une rapidité de décompression très supérieure.

**3.1.2.2.2 Codage intra-bande** Cette seconde génération ne prend pas en compte la redondance inter-bande et tente d'optimiser la compression des différentes sous-bandes indépendamment.

**EBCOT (JPEG-2000)** Le standard JPEG-2000 [TM01] (ISO/IEC International Standard 15444-1 – ITU-T Recommendations T.800) et son extension (Part 2 : ISO/IEC International Standard 15444-2 – ITU-T Recommendations T.801) sont conçus pour répondre à plusieurs demandes : permettre l'accès aléatoire, la scalabilité (spatiale et/ou en qualité) et la possibilité de privilégier une région d'intérêt, en compression avec et sans perte. Il repose sur l'algorithme EBCOT de Taubman [Tau00]. Il n'est pas nécessairement optimal si on se focalise uniquement sur les taux de compression, en particulier à cause de toutes les fonctionnalités qu'il permet de proposer. Il offre des taux légèrement inférieurs à JPEG-LS en compression sans perte, et un rapport débit/distorsion bien plus intéressant que JPEG. Cependant sa complexité calculatoire est assez importante.

La scalabilité en qualité est assez différente de celle de SPIHT et EZW. Elle s'effectue en couches de qualité. Pour chacune des couches une optimisation est faite afin de sélectionner l'information la plus importante. Si le flux de données est tronqué entre deux couches de qualité, l'image décodée sera certes améliorée en comparaison à l'image obtenue pour la couche de qualité inférieure, mais le rapport débit/distorsion sera moins optimal qu'au niveau de chacune des couches. Et donc, le codage est moins scalable que SPIHT et EZW qui cherchent à être optimaux quel que soit le point de troncature dans le flux. L'ajout de chaque couche de qualité demande plus de calculs au codeur pour l'optimisation et requiert l'insertion d'informations supplémentaires dans le flux de données.

EBCOT repose sur la compression indépendante d'un découpage en petits blocs (*Code Block*) de chacune des bandes de coefficients obtenues lors de la transformée. Typiquement la taille d'un code bloc est de  $64 \times 64$  coefficients, mais elle peut prendre toutes valeurs de  $4 \cdot 2^{E_1} \times 4 \cdot 2^{E_2}$  tant que  $E_1 + E_2 \leq 8$ , soit au minimum 16 coefficients et au maximum 4096.

La compression d'un bloc se fait plan de bits par plan de bits en 3 passes. La passe de signifiante, celle de raffinement et celle de nettoyage. La passe de signifiante consiste à

prédire les coefficients qui devraient devenir significatifs, à transmettre si oui ou non ils le sont et le cas échéant transmettre également leur signe. Un coefficient étant considéré comme significatif si son bit de poids fort est dans le plan de bits courant. La seconde passe sert à transmettre les bits du plan courant appartenant à des coefficients déjà significatifs lors du traitement du plan précédent. Enfin l'étape de nettoyage permet de transmettre l'information de localisation des coefficients prédits comme restant non significatifs, mais le devenant, ainsi que leur signe. Les deux premières étapes sont facultativement compressées à l'aide d'un codeur arithmétique adaptatif contextuel, tandis que la troisième l'est toujours.

Durant la compression de chacun des blocs, des informations concernant la qualité de reconstruction sont retenues et des points de troncature sont insérés dans le flux binaire. Ainsi lorsqu'un codage progressif est effectué, pour chaque couche de qualité souhaitée, une optimisation débit/distorsion permet de sélectionner dans chaque bloc compressé les portions de son flux à transmettre.

Lorsqu'une région d'intérêt doit être prise en considération, les bits des coefficients d'ondelettes lui appartenant sont tout simplement décalés vers les bits de poids fort avant le codage, de manière à être compressés en priorité.

Pour le lecteur intéressé, l'article de magazine [SCE01] effectue un survol de l'architecture et des fonctionnalités de JPEG-2000. Le tutoriel de Adams [Ada05], disponible avec l'implémentation libre Jasper<sup>5</sup>, décrit plus les techniques d'implémentation des algorithmes. Enfin, le livre de référence de Taubman et Marcellin [TM01] approfondit la théorie, le fonctionnement et le standard. Ce livre fournit également le code source de la version 2 de Kakadu<sup>6</sup>; les versions plus récentes du code source sont payantes, mais une version exécutable reste à disposition pour effectuer des tests.

**SWEET** SWEET [And97] est un algorithme de codage intra sous-bande rapide. Il effectue un découpage en quadtree plan de bit par plan de bit afin de localiser les coefficients devenant significatifs : lorsqu'un bit significatif est détecté sur un certain niveau, l'arbre est subdivisé, et l'algorithme est appliqué récursivement sur chacun des sous-blocs. Ainsi, SWEET effectue la construction du quadtree à l'aide d'un parcours en profondeur. Lorsqu'un pixel significatif est identifié, tous les bits nécessaires à sa représentation sont transmis. L'algorithme est appliqué sur l'ensemble des plans de bits de chaque sous-bande, de sorte à avoir une représentation scalable en résolution. Ce fonctionnement est très proche de celui de PROGRESS, mais n'exploite pas l'information inter-bande.

**SPECK** SPECK [PINS04] de Pearlman, Islam, Nagaraj et Said, reprend l'idée de partition d'ensemble de valeurs utilisée par SPIHT afin de transmettre les coefficients importants en priorité. Cependant SPECK l'applique uniquement à des blocs de coefficients internes à chaque sous-bande, sans utiliser d'information hiérarchique. Comme

---

<sup>5</sup><http://www.ece.uvic.ca/~mdadams/jasper/>

<sup>6</sup><http://www.kakadusoftware.com/>



SWEET, il partitionne chacune d'entre elle à l'aide d'un quadtree pour d'identifier les pixels significatifs. Cependant, il effectue une transmission par plan de bits de l'ensemble des sous-bandes et reste ainsi progressif en précision tout comme EZW ou SPIHT. Pour chaque plan de bit, un ordonnancement des blocs non significatifs de l'ensemble des quadtrees est effectué selon leur taille (les plus petits blocs d'abord). Ce traitement favorise ainsi le codage des coefficients proches de ceux précédemment identifiés comme significatif (et ayant donc une probabilité plus importante de le devenir à leur tour). Ceci permet de favoriser les coefficients de forte amplitude et ainsi de réduire les distorsions pour un débit donné.

**3.1.2.2.3 Codage mixte** Cette troisième génération tente d'exploiter les propriétés des deux précédentes, à savoir la localisation spatiale et hiérarchique des coefficients de forte amplitude, mais n'a apporté qu'un faible gain de compression. On peut citer les algorithmes EZBC (*Embedded ZeroBlocks Coding*) [Hsi01, HW02] et WBTC (*Wavelet zero-Block-zero-Tree Coding*) [MK06].

### 3.1.3 Compression mixte

Les techniques prédictives sont souvent celles qui permettent d'obtenir les meilleurs taux de compression, mais elles proposent rarement des solutions au codage progressif qui permet de fournir rapidement un résumé de l'information. Sur ce point les codeurs les plus efficaces utilisent presque tous les transformées en ondelettes ainsi qu'un codage par plan de bits, afin de fournir une progressivité sur la qualité de l'image reconstruite. Les approches intra-bande sont actuellement les plus réputées, mais des codeurs inter-bandes très efficaces tels que SPIHT sont toujours d'actualité.

En utilisant une quantification uniforme, les approches prédictives présentées peuvent permettre de manière assez simple d'utiliser une compression presque sans perte (c.f. 3.1.4) efficace à haut débit (pour un faible pas de quantification). Cependant, pour une compression avec pertes à bas débit, cette approche est peu efficace. La combinaison de l'approche prédictive, avec celle par transformée peut permettre d'obtenir de bons résultats en compression avec pertes non scalable en qualité. Bien que ce sujet soit légèrement hors de propos pour cette étude, il semble tout de même intéressant de mentionner cette technique.

Les approches DPCM traitent les pixels l'un après l'autre, et utilisent les pixels voisins précédemment traités pour pouvoir prédire le suivant. Ainsi, en se plaçant dans une situation avec pertes, pour pouvoir avoir un schéma reproductible au décodeur, le traitement provoquant la perte ne peut être effectué que sur le pixel en cours de prédiction afin que le traitement pseudo-inverse puisse être réalisé et permette d'obtenir une estimation du pixel qui pourra ensuite être utilisée pour prédire le suivant.

Une extension de l'approche DPCM consiste à appliquer la prédiction sur une plage de données (un bloc par exemple) : toutes les valeurs d'un ensemble sont alors prédites simultanément. Cette décorrélation est certainement moins efficace qu'une décorrélation DPCM, cependant le bloc étant prédit comme un tout, il est encore possible d'effectuer

une transformation pour décorrélérer les données résiduelles du bloc avant d'effectuer une quantification. Le lecteur intéressé pourra aller voir du côté de la compression intra du codeur H.264/MPEG-4 AVC [Ric03] réutilisée pour proposer un codec image dans [ZVR08] par exemple.

En se plaçant dans le domaine spectral, la prédiction par bloc permet en partie de diminuer l'amplitude des coefficients basses fréquences. La prédiction pourrait également être considérée et appliquée directement dans un espace transformé, telle qu'une décomposition en ondelettes, par exemple, afin d'éviter des effets de blocs.

### 3.1.4 Compression presque sans perte

Le terme « presque sans perte », concerne majoritairement les approches de compression dont les pertes sont contrôlées par le PAE (*peak of absolute error*) correspondant à une contrainte sur l'erreur absolue maximale autorisée quel que soit le pixel :

$$\text{PAE} = \max_{k \in [1..N] \times [1..M]} \|I(k) - \tilde{I}(k)\| = \|I - \tilde{I}\|_{\infty}, \quad (3.2)$$

avec  $I(N \times M)$  l'image originale et  $\tilde{I}(N \times M)$  celle obtenue après décompression.

Pour que la compression presque sans perte ait du sens, les valeurs du PAE sont choisies de manière à conserver la qualité visuelle de l'image originale, tout en améliorant la compression. Contrairement aux approches utilisant l'erreur quadratique moyenne (MSE) comme critère de compression (comme la majorité des algorithmes de compression avec pertes), le PAE permet de garantir la qualité pour tous pixel, et ce, quel que soit son voisinage (proche d'un contour, dans une zone texturée ou homogène). Ce critère peut ainsi permettre de mieux répondre aux exigences médicales.

La façon la plus naïve d'effectuer l'estimation presque sans perte est d'appliquer une quantification directement dans le domaine image, puis d'appliquer une compression sans perte de l'image quantifiée. Cependant, elle se révèle être moins efficace que les approches qui vont être présentées, et peut introduire une gêne, lors de l'affichage, due à la réduction du nombre de niveaux de gris présents.

Pour les techniques strictement prédictives, la méthode majoritairement suivie est l'emploi d'une quantification scalaire uniforme, sur le résidu de prédiction, d'un pas  $\delta$  égal au PAE :

$$\check{\epsilon} = Q_{\delta}(\epsilon) = \text{sign}(\epsilon) \left\lfloor \frac{|\epsilon| + \delta}{2\delta + 1} \right\rfloor. \quad (3.3)$$

La prédiction doit alors s'effectuer à l'aide des valeurs causales reconstruites presque sans perte après déquantification de leur résidu :

$$\tilde{\epsilon} = Q_{\delta}^{+}(\check{\epsilon}) = \check{\epsilon}(2\delta + 1). \quad (3.4)$$

Les extensions presque sans perte de CALIC [WMS95], de TMW [MT98], et bien d'autres comme [CR94, KM95, KM98], utilisent cette approche qui peut également être étendue en prédiction hiérarchique en effectuant une compression presque sans perte

de la résolution la plus fine et en quantifiant les résidus de prédiction de chacune des bandes nécessaires à la prédiction des suivantes.

Des méthodes beaucoup plus complexes basées sur une quantification par treillis [KM95, KM98, WB00] peuvent également permettre d'optimiser le débit et/ou les distorsions, au sens de la MSE, tout en respectant la contrainte presque sans perte. Cependant les résultats sont décevants si on tient compte du temps d'exécution supplémentaire introduit [WB00]. Ainsi, toujours afin de réduire la MSE, Wu et Bao [WB00] proposent d'utiliser une modélisation contextuelle des biais de prédiction induits par la quantification et de transmettre régulièrement dans le flux de données une information supplémentaire permettant de modéliser ce biais. La prédiction étant ainsi meilleure, cette méthode permet à la fois de réduire le coût de codage, mais également la MSE.

JPEG-LS a également suivi une approche légèrement différente qui promeut autant que possible l'utilisation du codage RLE, tant que celui-ci satisfait le PAE. Dans le cas contraire, (3.3) est utilisé pour quantifier l'erreur résiduelle. Cela permet une compression efficace sans nécessiter l'utilisation d'un codage arithmétique, mais peut générer des artefacts dérangent (trainées du au RLE) lorsque le PAE devient important.

Il est beaucoup plus difficile de s'adapter efficacement au PAE dans le domaine transformé. Ainsi la manière la plus simple est d'effectuer une compression avec pertes de la transformée, puis de compresser presque sans perte l'erreur entre l'image originale et l'image reconstruite. Ce modèle est utilisé dans [WB97, Kri03a, Kri03b, YCP05, YP06]. Cependant pour améliorer l'efficacité il faut optimiser le pas de quantification à appliquer sur les ondelettes, de façon à minimiser le débit nécessaire pour les deux représentations. [Kri03b] utilise JPEG-2000 pour les coefficients d'ondelettes et fait cette optimisation manuellement offline, tandis que Yea [YP06], qui utilise SPIHT, effectue une estimation du coût pour trouver le point de coupure optimal dans le flux compressé.

[Abh03] propose une approche différente effectuant la quantification dans un schéma de lifting lors du premier niveau de décomposition en ondelettes. Celle-ci est peu efficace, et les artefacts sont très proches de ceux obtenus lorsque la quantification est effectuée directement sur l'image.

L'utilisation de la compression irréversible par transformée couplée au codage du peut permettre d'avoir une représentation progressive en qualité, jusqu'au presque sans perte. [Kri03b] va un peu plus loin en permettant d'imbriquer plusieurs niveaux de qualité presque sans perte par raffinements successifs : PAE=0 (qualité sans perte), PAE=1, PAE=4 et PAE=13, dans un même flux de données. Enfin l'étude [AMCS06] cherche à proposer une représentation progressive en qualité optimisée pour minimiser le PAE lors du codage des coefficients d'ondelettes.

## 3.2 Études et standardisation pour la communication des images médicales

Cette section s'intéresse aux études plus spécifiques à la communication d'images médicales ainsi qu'aux solutions mises en place dans le standard DICOM.

### 3.2.1 Images bidimensionnelles

Parmi les modalités bidimensionnelles d'imagerie médicale, la radiographie est la plus utilisée. Les premières images étaient acquises directement sur film et furent par la suite informatisées par numérisation de la copie physique. Elle peuvent désormais être acquises à l'aide de capteurs électroniques permettant une quantification numérique immédiate (CR : *Computed Radiography*). Leurs quantités importantes et leurs dimensions (pouvant par exemple dépasser  $2000 \times 3000$  sur 16 bits pour une radio des poumons) les rendent également de bonnes candidates à la compression.

Les CR ainsi que les coupes de scanner, d'IRM et de PET (tomographie par émission de positrons : contrairement à la tomographie classique, les radiations captées et utilisées pour reconstruire une image sont envoyées par un produit radioactif injecté au patient) sont principalement considérées dans les études qui ont été menées sur la compression des images médicales bidimensionnelles [DvAPL97, KOK<sup>+</sup>98, Clu00]. Ces travaux comparent les résultats d'algorithmes existants en compression généraliste tels que les commandes unix *pack* (Huffman adaptatif), *compress* (basé sur LZW), *gzip* (basé sur LZ77) ou encore le codeur arithmétique adaptatif STAT<sup>7</sup> (de F. Bellard); et en compression d'images tels que le JPEG sans perte<sup>8</sup>, JPEG-LS<sup>8</sup>, JPEG-2000<sup>8</sup>, CALIC<sup>8</sup>, SPIHT<sup>8</sup>, TMW<sup>8</sup>, S+P [SP93, SP96a], etc. Parmi ces codeurs, on trouve une variété d'algorithmes prédictifs et par transformées, avec codage entropique adaptatif (arithmétique, Golomb-Rice) ou non adaptatif (tables Huffman fixes).

Les résultats de ces études mettent en évidence que les techniques de compression d'images restent plus appropriées que les méthodes généralistes. CALIC ressort comme le codeur le plus performant sur l'ensemble des images. Certains algorithmes peuvent cependant être plus intéressants selon les modalités : TMW est légèrement plus efficace que CALIC sur les images tomographiques (mais reste plus lent), SPIHT est mieux adapté pour les IRM, et JPEG-LS pour les scintigraphies. Leurs taux de compressions sont compris en moyenne entre 2:1 et 5:1.

Comme l'efficacité des algorithmes sur les images médicales semble cohérente avec celle sur les images naturelles, les standards présentés dans la section précédente sont majoritairement utilisés pour la compression sans perte de ces images 2D, de même que pour la compression coupe par coupe des images volumiques. On trouve peu de propositions d'algorithmes sans perte cherchant à s'adapter aux types de contenus de ces images afin d'améliorer les résultats.

Pour les images médicales bidimensionnelles, les recherches se sont donc plutôt penchées sur la compression avec pertes. Au début des années 90, il avait été envisagé d'utiliser la compression des coefficients de la transformée en cosinus discret de la totalité de l'image (*Full-Frame DCT*) après une quantification adaptée, de sorte à supprimer les effets de blocs indésirables d'une compression de type JPEG. Des références et d'autres approches assez anciennes sont plus détaillées dans [CMNADB08]. Des travaux plus récents, tels que [KBB<sup>+</sup>09] ont cherché à quantifier les taux de compressions

---

<sup>7</sup><http://bellard.org/>

<sup>8</sup>cf. 3.1

acceptables pour les standards actuels et ont contribué à l'élaboration de la norme de la CAR (Association Canadienne de Radiologie) [CAR10, CAR11]. Celle-ci essaie d'orienter les radiologues, lorsqu'un archivage irréversible est envisagé, en fonction de la technique d'acquisition utilisée, et de la zone corporelle étudiée. Certaines techniques plus spécifiques aux modalités à traiter ont également été proposées en mammographie [PPT<sup>+</sup>03] et en échographie [GSP05]. La première utilise une modélisation de la région d'intérêt des mammographies et une compression adaptée, la seconde tient compte du type de bruit présent dans les échographies et tente de réduire son influence à l'aide d'une quantification adaptée. Ces efforts concernant la compression irréversible des images médicales sont également présents pour l'imagerie volumique. L'utilisation de la quantification vectorielle algébrique sur les coefficients d'une transformée en ondelettes volumique [Gau06, GM09] a par exemple été proposée dans le but de réduire les distorsions de manière à moins pénaliser le diagnostic.

### 3.2.1.1 Accès aléatoire

Pour de très grandes images, comme les lames virtuelles par exemple, il peut être intéressant de pouvoir accéder rapidement à une portion spatiale particulière de l'image, sans nécessiter une transmission ou un décodage de l'image complète. Avec des approches prédictives adaptatives, l'unique solution est de découper l'image avant compression en imagelettes de tailles raisonnables : suffisamment grandes pour ne pas trop diminuer les performances de compression, suffisamment petites pour éviter de transmettre trop d'informations.

Avec des codeurs ondelettes comme EBCOT, puisque le découpage est déjà effectué dans l'ensemble des sous-bandes de coefficients d'ondelettes, il est possible de récupérer l'ensemble des paquets d'informations correspondant à la localisation spatiale souhaitée.

### 3.2.1.2 Régions d'intérêt

La région d'intérêt, qu'il est plus facile de définir sur des images médicales que sur des images naturelles, permet en particulier de différencier le fond (inutile au diagnostic) de la forme (données intéressantes). Ainsi le fond peut être compressé avec beaucoup de pertes (voire supprimé) tandis que la forme est compressée sans perte ou avec de faibles dégradations. Dans [PPT<sup>+</sup>03] le tissu, le contour et le fond des mammographies sont ainsi compressés de manière indépendante. Dans un cadre général, il est possible de spécifier les régions d'intérêt manuellement ou de façon automatique par segmentation/classification.

La transformée en ondelettes a été adaptée pour être appliquée sur des régions de l'image de formes arbitraires comme dans les travaux de Li et Li [LL00] (SA-DWT : *Shape-Adaptive DWT*) et étendue aux ondelettes entières (ISA-DWT : *Integer SA-DWT*) dans [AHS02] par Abu-Hajar et Sankar qui l'utilisent avec un codeur dérivé de SPIHT pour obtenir de bonnes performances sur des images naturelles [AHS04]. Cette approche peut facilement être utilisée en imagerie médicale.

### 3.2.1.3 Progressivité

La majorité des algorithmes scalables employés sur des images médicales s'appuient sur la transformée en ondelettes (SPIHT, EBCOT, ...). Dans [GEVK00] Grüter *et al.* proposent également l'utilisation de la ROPD (*Rank-Order Polynomial Decomposition*) pour la compression progressive de ces images. Elle s'appuie sur une décomposition en sous-bandes morphologiques [ELK95], à l'aide de prédicteurs polynomiaux non linéaires.

Le codeur LAR (*Locally Adaptive Resolution*) [DBBR07] initialement conçu pour effectuer une compression avec pertes scalable a également été étendu pour effectuer une compression *lossy-to-lossless* de manière efficace [BDR05, DBM06, PBDB08]. Il repose sur une représentation de l'image sous la forme d'une partition en quadtree de zones relativement homogènes auxquelles est associé un niveau de gris représentatif. Une image grossière est construite par interpolation dans les zones homogènes, et est utilisée pour prédire l'originale. L'erreur d'approximation est considérée comme la texture et est codée de manière multi-résolution en s'appuyant sur la décomposition en quadtree précédente. Ce modèle de représentation simple peut être utilisé et étendu pour satisfaire divers critères, tels que la progressivité ou le codage de ROI, et diverses techniques de décorrélation de l'information textuelle peuvent être adoptées selon le contexte d'utilisation.

### 3.2.2 Images volumiques

Si pour les images bidimensionnelles on trouve peu d'algorithmes spécialisés, en imagerie volumique la littérature est un peu plus riche. Certaines études s'appuient tout de même sur des techniques initialement destinées aux vidéos, qui peuvent être également considérées comme des images volumiques. L'approche la plus triviale pour leur compression est l'encodage de chacune des coupes indépendamment des autres à l'aide d'algorithmes bidimensionnels. Cette technique est très clairement sous-optimale, en terme de taux de compression, puisqu'elle ne prend pas en considération les corrélations pouvant exister entre des coupes successives. Cependant, elle est couramment employée en compression de vidéos lors de l'acquisition, afin de réduire la complexité des algorithmes, de conserver une bonne qualité ou de faciliter l'accès aléatoire et le montage en studios par exemple. En imagerie médicale c'est une approche qui peut être employée lorsque le contexte d'utilisation nécessite de favoriser l'accès aléatoire à des coupes éparées.

Les approches bidimensionnelles ont également connu de nombreuses extensions volumiques et quelques travaux se sont déjà penchés sur la problématique d'une compression efficace permettant un accès aléatoire rapide (localement ou via un réseau). Les résultats fournis par ces travaux sont prometteurs. La notion de régions d'intérêt et de codage d'objets peut également être importante et se révéler bénéfique : fixer des qualités adaptées aux besoins, éviter de transférer/stocker des régions inutiles. Il existe déjà quelques techniques intéressantes pour leur compression, ainsi que pour modéliser leur localisation.

Afin de supprimer la redondance d'information entre les images successives la prédiction et la transformation sont encore applicables. Les principales techniques de décorrélation volumique sont présentées en commençant par les approches prédictives, très utilisées en vidéo, puis en enchaînant sur celles par transformées, plus utilisées en imagerie médicale et satellitaire et commençant également à émerger en compression vidéo très scalable. Les applications de ces outils en milieu médical, pour faciliter l'accès aléatoire, la représentation de la région d'intérêt ou utiliser une modélisation objet seront également discutés. Le lecteur pourra également se référer aux états de l'art [PvAdRD01] et [SMB<sup>+</sup>03] qui sont assez complémentaires, le premier étant plus orienté sur le codage prédictif, et le second sur le codage par transformées. Le chapitre [OZONA08] fait également un état de l'art des deux approches et traite également les séquences vidéos médicales (2D+t). Les termes « coupe » et « image » seront indifféremment employés par la suite.

### 3.2.2.1 Codage prédictif

Le domaine de la compression vidéo, qui peut être considérée comme une modalité volumique, possède une vaste littérature dans laquelle le codage prédictif est très apprécié. Certaines techniques ont été améliorées durant de nombreuses années et sont ainsi devenues très efficaces. Elles sont parfois utilisées en compression avec pertes d'images médicales volumiques.

Les codecs vidéos "grand public" sont conçus de façon à offrir un décodage qui puisse être effectué en temps réel. Ils essaient de fournir un accès aléatoire assez fin pour faciliter la navigation et permettre un positionnement temporel dans la vidéo qui soit rapide. Les plus récents cherchent également à être très progressifs afin d'offrir la possibilité d'effectuer du streaming pour différents débits et/ou résolutions spatiales et/ou résolutions temporelles. Pour l'accès aléatoire, la technique du GOP (*Group Of Pictures*) est très employée. Elle consiste à compresser les séquences par paquets d'images de longueurs fixes (ex : 8 ou 16 images). Pour de tels codecs, le codage prend souvent beaucoup plus de temps de façon à mieux décorréler l'information et à réduire les calculs lors du décodage pour permettre une reconstruction et un affichage temps réel.

Trois approches classiques de prédiction volumique peuvent être dissociées. La première consiste à prédire la coupe à compresser à partir d'autres coupes déjà connues (les précédentes par exemple). L'image de l'erreur de prédiction volumique est ensuite décorrélée à l'aide de techniques 2D. Ceci s'effectue souvent à l'aide de transformées (DCT-2D ou DWT-2D). Les deux approches suivantes sont des extensions des techniques bidimensionnelles. Elles utilisent l'information volumique disponible en chacun des pixels à compresser. L'extension volumique séquentielle (DPCM), qui se résume en un parcours coupe par coupe, ligne par ligne, et une prédiction pixel par pixel est dissocié de l'extension volumique hiérarchique (HINT) qui utilise une représentation volumique basse résolution (par simple sous-échantillonnage) pour prédire les pixels à une résolution supérieure.

**3.2.2.1.1 Prédiction de coupe** Le standard de compression vidéo le plus récent, H.264/MPEG-4 AVC (*Advanced Video Coding* : ITU-T Recommendation H.264, ISO/IEC MPEG-4 AVC 14496-10) qui est une évolution des standards H.261, H.262 et H.263 précédents, relève des travaux en commun du groupe H de l'ITU, spécialisé dans l'audiovisuel et les systèmes multimédias, et du groupe de travail MPEG (*Moving Picture Experts Group*) d'ISO/IEC. Sans rentrer dans les détails (le lecteur pourra se référer à [Ric03]), il offre de nombreuses fonctionnalités, dont la scalabilité temporelle, l'accès aléatoire et la décomposition en objets.

L'accès aléatoire est mis en place avec l'aide des GOPs, et la scalabilité temporelle (permettant d'encoder la vidéo à diverses fréquences d'acquisition temporelle dans le même flux de données) est permise grâce à une mise en place astucieuse de divers prédicteurs d'image. On distingue en particulier les images de type I (pour *Intra*), P (pour *Predicted*) et B (pour *Between* ou *Bi-predicted*). Comme l'indique leur nom, les images I sont compressées en intra : une approche bidimensionnelle ne nécessitant pas d'informations sur les autres images est utilisée. Les images P sont compressées après une prédiction à l'aide d'image(s) de références précédentes, et les images B après une prédiction à l'aide d'image(s) de références précédentes et/ou suivantes déjà encodées.

L'image devant être prédite est découpée en blocs de taille fixe. Pour chacun d'entre eux, les images de référence sont utilisées pour rechercher un bloc le plus similaire possible, qui sera employé pour la prédiction. Cette technique permet d'effectuer une compensation du mouvement dans les vidéos, et la localisation du bloc le plus ressemblant est codée à l'aide d'un vecteur souvent appelé vecteur (de compensation) de mouvement. Cette localisation peut être sub-pixelique, au quel cas les blocs prédicteurs sont générés à l'aide d'une interpolation par filtrage. L'image des résidus de prédiction est ensuite compressée avec pertes à l'aide d'une TCD-2D.

La succession de coupes des images volumiques médicales ne respectent pas l'hypothèse d'objet en mouvement par translation. Les contours/surfaces des différents objets volumiques (organes...) représentés coupe-à-coupe ont certes tendance à se déplacer mais également à se déformer/dilater/contracter. Cependant, ces déformations seront localement faibles si la résolution en ( $z$ ) est assez fine (comme les TDM coupes fines par exemple). Des prédicteurs utilisant une estimation de mouvement permettent donc de décorrélérer l'information mais seront moins efficaces à plus faible résolution (IRM par exemple).

C'est pourquoi Nosratina *et al.* [NMOL96], et Srikanth et Ramakrishnan [SR05] effectuent la compression d'IRM à l'aide de modèles de prédiction par compensation de déformations. Ces modèles utilisent les déformations d'un maillage pour prédire des images successives. Bien que ces articles ne proposent pas de résultats pouvant être aisément comparés, on peut supposer que, pour les images ayant une résolution en ( $z$ ) assez faible, une telle approche puisse obtenir de meilleurs résultats qu'avec une compensation de mouvement classique, par blocs.

**3.2.2.1.2 Prédiction séquentielle** Proche de la prédiction de coupe, dos Santos et Scharcanski proposent dans [dSS08] une compensation de mouvement en deux étapes



pour la compression sans perte et presque sans perte d'angiographies (représentation temporelle). Un *bloc matching* est effectué pour trouver le bloc le plus ressemblant dans une image de référence (image précédente), et est utilisé pour générer un prédicteur linéaire adaptatif du pixel courant. Les résultats expérimentaux de cette technique, de complexité calculatoire importante, n'améliorent que faiblement la compression effectuée frame par frame par JPEG-LS (de l'ordre de 0.08 bits/pixel pour des images 8 bpp, et 0.13 bpp pour des images 12 bpp).

Comme l'estimation du mouvement est très coûteuse en temps de calcul, De Rycke et Philips [dRP99, PvAdRD01] utilisent un codeur intra (JPEG-LS) pour la majorité de l'image et une compensation de mouvement uniquement sur les pixels pouvant être potentiellement mal prédits par le prédicteur 2D. Ce changement de mode est effectué automatiquement, avec l'aide de l'image précédente, et ne nécessite donc d'aucune information supplémentaire. En pratique, moins de 1% des pixels sont prédits par la compensation de mouvement, ce qui permet une réduction importante de la complexité.

Diez-Garcia *et al.* [DGSWAL05] utilisent un prédicteur volumique adaptatif en une passe utilisant un schéma similaire à la prédiction DPCM de LOCO ou CALIC étendu sur une troisième dimension. Ainsi, les voxels utilisés pour la prédiction appartiennent à la coupe précédente et à la portion DPCM déjà prédite de l'image. Le prédicteur est optimisé au sens des moindres carrés, sur des données sélectionnées à l'aide d'une fenêtre « d'entraînement ». Cette approche n'offre pas vraiment de meilleures performances de compression que des codeurs utilisant une transformée volumique et ne permet pas d'offrir la progressivité. Cependant, la méthode adaptative proposée bénéficie de paramètres pouvant être ajustés (taille de la fenêtre d'apprentissage, et un seuil permettant d'éviter de recalculer le prédicteur) afin d'accélérer les traitements au détriment des performances de compression. Similairement à JPEG-LS, cette technique est alors plus adaptée pour une compression rapide facilitant les transferts.

**3.2.2.1.3 Prédiction hiérarchique** Roos et Viergever [RV93] utilisent une approche DPCM volumique simple et une extension 3D de HINT et les comparent aux approches 2D. Leur conclusion est qu'une extension volumique n'apporte qu'un faible gain de codage. Ces résultats restent en concordance avec l'article de Aiazzi *et al.* [AABA96] qui présente GRINT (*Generalized Recursive Interpolation*) une généralisation de HINT, dont ils comparent l'entropie d'ordre zéro après décorrélation 2D et 3D. Sur les résultats présentés, seule une faible amélioration (-0.05 bpp) est notée avec le passage du 2D au 3D. Cependant HINT-3D et GRINT peuvent permettre d'effectuer un codage progressif du volume. Ces résultats peuvent être remis en cause sur des images plus récentes pour lesquelles la corrélation inter-coupe peut se retrouver accentuée par une distance d'acquisition plus fine et/ou par des données moins bruitées.

### 3.2.2.2 Codage par transformée

Comme pour les transformées bidimensionnelles, une simple extension mathématique des transformées monodimensionnelles suffit à obtenir des transformées volumiques [BPBCG93]. Les différences de performances viennent le plus souvent des techniques utilisées pour compresser et organiser les coefficients de la transformée. L'architecture des codeurs est donc sujette à discussions. Ils sont souvent des extensions volumiques des codeurs ayant fait leurs preuves sur les images 2D. Ils reprennent les principes algorithmiques et les étendent sur des volumes de pixels. Ainsi une décomposition en quadtree devient une décomposition en octree ou un codage par "petits" blocs (EBCOT) devient un codage par "petits" cubes. L'un des états de l'art incontournable est celui de Schelkens *et al.* [SMB<sup>+</sup>03], qui détaille et compare les résultats de quelques algorithmes volumiques.

Pour l'archivage médical, un gain de compression en comparaison aux mêmes algorithmes 2D est généralement notable mais il n'est pas fantastique. La plupart de ces codeurs permettent une représentation progressive de la totalité du volume. Cette fonctionnalité est intéressante lorsque plusieurs qualités différentes sont nécessaires, ou lorsqu'une représentation basse qualité du volume est suffisante pour pouvoir localiser les coupes et/ou objets réellement utiles et devant être transférés sans perte. Malheureusement ces codeurs n'offrent pas souvent la possibilité de récupérer une information localisée.

JPEG-2000 permet de s'adapter à de nombreux besoins mais était initialement conçu pour les images 2D uniquement, et la transformée en ondelettes était assez peu paramétrable. La seconde partie (ISO/IEC International Standard 15444-2 – ITU-T Recommendations T.801 : *JPEG-2000 Part 2 : Extensions*) a rapidement été ajoutée au standard (approuvée : 08/2002). Elle permet, parmi beaucoup d'autres améliorations, de considérer des images multi-composantes (images volumiques ou multi-spectrales). Cette version autorise la décorrélation inter-composantes 2D à l'aide de transformées linéaires matricielles (KLT par exemple), ou d'ondelettes (préférées pour les images volumiques). Le codeur EBCOT initial est conservé, les images n'étant pas réellement considérées comme volumiques. Ainsi une autre partie du standard (ISO/IEC International Standard 15444-10 – ITU-T Recommendations T.809 *JPEG-2000 Part 10 : Extensions for three-dimensional data*), également appelée JP3D, a également été approuvée (08/2007). Cette extension est plus spécifique aux images volumiques, et étend le codeur EBCOT pour exploiter l'information volumique, de manière à être encore plus efficace sur ce type d'images.

Des techniques plus complexes d'extension de la transformée volumique ont également fait leur apparition. Il existe notamment une demande croissante pour des codecs très scalables en vidéo, permettant de représenter, dans un même flux binaire, différentes résolutions d'image, différentes résolutions temporelles et ce pour différents débits. Une solution pouvant être adoptée consiste en une compensation du mouvement par blocs, temporellement décorrés à l'aide d'ondelettes. WAVIX [VGP02, BFG05],

et MC-EZBC<sup>9</sup> [HW01, WC02](*Motion-Compensated EZBC*) sont deux exemples suivant cette approche. WAVIX utilise JPEG-2000 pour compresser chacune des images de chacune des sous-bandes décorréées temporellement ( $\approx$  3D-EBCOT). MC-EZBC a un fonctionnement relativement similaire mais effectue le codage à l'aide de 3D-EZBC.

### 3.2.2.3 Accès aléatoire

Certaines techniques de codage prédictif peuvent être très compétitives pour leurs taux de compression, rapidité et coût mémoire pour le codage/décodage. Cependant l'approche prédictive adaptative devient beaucoup plus critique pour l'accès à une coupe particulière. En effet, pour obtenir toute l'information nécessaire à sa décompression (contexte/prédicteur), il faut préalablement transférer et décoder la totalité des informations dont elle dépend. De même des algorithmes comme 3D SPIHT obligent un décodage complet du volume pour pouvoir accéder à une coupe particulière sans aucune perte.

Une solution pour réduire ce problème est de travailler par groupes de coupes (GOS : *Group Of Slices*), comme en vidéo avec les groupes d'images (GOP : *Group Of Pictures*) de longueurs réduites (par la suite on ne différenciera pas GOS et GOP, de même l'axe transaxial (z) des volumes sera confondu avec l'axe temporel des vidéos). Ainsi on obtient une coupe particulière en ne décodant que l'information nécessaire contenue dans le GOS, au détriment d'une baisse d'efficacité. Les GOPs d'un codage prédictif offrent souvent des images codées en intra uniquement (la première image par exemple), tandis que des GOPs résultant d'une transformée en ondelettes offrent des représentations progressives : en résolution temporelle et spatiale, et en qualité. Un résumé du contenu du volume est ainsi disponible pour un faible coût et permet un accès aléatoire plus rapide à une sélection des coupes (ou plages de coupes) à consulter dans une qualité supérieure. Cette approche peut donc permettre de réduire les transferts réseaux et accélérer la visualisation lors d'une interrogation à distance, en comparaison à une compression totalement volumique.

Dans l'optique d'améliorer l'efficacité de SPIHT-3D, Cho, Kim et Pearlman [CKP04] proposent l'utilisation d'une décomposition en arbre asymétrique pour le tri partiel des coefficients (AT-SPIHT : *Asymmetric Tree SPIHT*). Ils montrent qu'il reste efficace même sur des GOPs de petite taille. L'accès aléatoire est ainsi favorisé.

Menegaz et Thiran [MT03] effectuent une compression volumique à l'aide d'ondelettes, et organisent le codage de façon à pouvoir favoriser un accès aléatoire par coupe. Cependant la technique proposée n'est pas toujours rentable lorsqu'une seule coupe doit être récupérée. En effet, le nombre de coefficients nécessaire pour reconstruire d'une coupe particulière dépend du nombre de niveaux de décompositions et de la taille du support des ondelettes utilisées pour la décorrélation. Ainsi la compression intra de la coupe en question peu devenir rapidement plus rentable.

<sup>9</sup>codes source disponibles : <http://www.cipr.rpi.edu/research/mcezbc/>

Une solution intéressante à ce problème est proposée par Wu et Qiu [WQ05] qui ont également cherché un algorithme rapide permettant un accès aléatoire efficace. Leur proposition, M3DW (*Modified 3D dyadic Wavelet*), utilise une décomposition dyadique volumique non conventionnelle. Ils estiment que les coefficients d'ondelettes hautes fréquences ne sont pas suffisamment corrélés entre des coupes successives. La décomposition utilisée est alors une décomposition dyadique 2D classique sur un niveau, suivie d'une décomposition inter-coupe en ondelettes de Haar uniquement sur les basses fréquences. Ce processus peut ensuite être réitéré sur le volume basse résolution. La compression des coefficients est ensuite effectuée avec un codeur Golomb-Rice adaptatif pour rester rapide.

Le choix de l'ondelette de Haar pour la décorrélation inter-coupe n'est pas anodin. Celle-ci, ayant le support le plus compact, permet de réduire au maximum le nombre de coefficients nécessaires au décodage d'une seule coupe. Avec cette technique, le nombre de valeurs non compressées qu'il est nécessaire de transmettre pour la reconstitution d'une coupe est de l'ordre de  $4/3$  supérieur à un codage intra. Cependant les résultats sur la base 8 bits du CIPR (cf. 1.1.3) montrent que le gain de codage apporté par cette technique dépasse généralement les 33% en comparaison à JPEG-LS. Ainsi le débit binaire compressé nécessaire au décodage d'une seule coupe est légèrement inférieur à celui de JPEG-LS en intra. Les taux de compression sont également meilleurs que 3D SPIHT (-0.12 bpp) lui-même plus performant que CALIC (-0.36 bpp) et permettent un gain moyen supérieur à 0.8 bpp en comparaison à JPEG-2000. Enfin le décodage d'un volume complet est plus rapide qu'avec JPEG-LS (réputé pour sa rapidité). Il en résulte que cette technique rapide est à la fois efficace pour le stockage et pour la transmission de coupes aléatoires.

#### 3.2.2.4 Région d'intérêt et objets

Menegaz soumet un algorithme volumique de compression d'une décomposition de l'image en objets d'intérêts permettant un décodage progressif de chacun d'entre eux, indépendamment [Men00, MT02]. Son approche repose sur une décomposition en ondelettes classique sur la totalité du volume. Pour chaque région d'intérêt, tous les coefficients nécessaires à sa reconstruction sont conservés et encodés (dans l'article deux codeurs 3D sont considérés EZW-3D et MLZC). Ainsi, les coefficients d'ondelettes contribuant à plusieurs objets (proches des contours) sont codés de façon redondante. Une alternative à cette redondance serait l'utilisation de la transformée en ondelettes adaptée à une forme arbitraire (cf. section 3.2.1.2).

Menegaz suppose que les délimitations des objets sont déjà connues du codeur et du décodeur, et ne considèrent pas le codage de celles-ci. Elle suggère tout de même l'utilisation du modèle « Hybrid-Ellipsoid » de Vaerman [Vae99, VMT99] afin de modéliser la surface des objets. Durant sa thèse sur les images médicales 2D [Che07], Chen propose également d'utiliser un découpage polygonal pour la modélisation de région d'intérêt afin de réduire son coût en comparaison à la compression d'un masque binaire [CTC06].

Le modèle précédent doit connaître la région d'intérêt avant compression. Dans [SAN10] Sanchez *et al.* proposent une approche progressive originale pour favoriser la transmission d'une ROI volumique. Ils s'appuient sur un codeur EBCOT-3D modifié et une optimisation de l'organisation des paquets de données compressées s'adaptant à la ROI désirée, et qui fournit également un aperçu à faible qualité du voisinage de la ROI. L'approche est plutôt destinée pour la télé-médecine interactive, dans un contexte où le client peut choisir sa région d'intérêt.

### 3.2.3 Standard DICOM

DICOM (*Digital Imaging and COmmunication in Medicine*) est un standard international pour l'échange d'images et de données médicales. Il inclut notamment la spécification de formats de données pour la communication d'images par le biais d'un réseau ou d'un support physique, ainsi que des informations relatives au contexte d'acquisition (matériel, paramètres d'acquisition, résolution physique, informations sur le patient concerné, ...). Le standard couvre également les protocoles d'impression des images, de visualisation ou de sécurisation des échanges.

Afin d'éviter la normalisation d'algorithmes pour la compression des images et pour garantir la pérennité des données, DICOM s'appuie essentiellement sur des normes internationales existantes, non spécifiques au contenu médical. Ainsi, le contenu des images peut être intégré sans compression (RAW), compressé à l'aide d'un simple codage RLE, de l'algorithme DEFLATE (même compression que gzip sur unix), de JPEG, JPEG sans perte, JPEG-LS, JPEG-2000, MPEG-2, ou encore MPEG-4.

Avec les algorithmes RAW, RLE, DEFLATE, JPEG, JPEG sans perte, JPEG-LS et JPEG-2000, seules des images 2D sont gérées par le standard. Ainsi, pour les séquences d'images (images volumiques), si l'un de ces formats est utilisé, chaque coupe doit être transmise séparément. Un message/fichier est donc généré pour chacune d'entre elles, et chaque message inclut un ensemble de données nécessaires à son utilisation : informations patient, paramètres d'acquisition, ... , soit une certaine redondance.

Pour pouvoir transmettre directement des séquences d'images, et uniquement des séquences d'images couleurs  $3 \times 8$ -bits compressées avec pertes, seule l'utilisation de MPEG-2 était autorisée par le standard avant l'ajout de l'extension de JPEG-2000 (Part 2) à DICOM en 2005. Celui-ci permet d'appliquer une décorrélation multi-composantes qui peut être employée pour décorréler les coupes successives. Désormais, des séquences sur plus de 8-bits, peuvent être transmises avec ou sans pertes, puisque cette extension le permet, et ce, dans un seul message. La compression des images volumiques telles que les TDM ou les IRM peut ainsi être améliorée.

Depuis, MPEG-4 offrant de meilleures performances pour la compression vidéo que MPEG-2, a également été ajouté (le texte final date d'avril 2011).

Le lecteur intéressé par plus de détails est invité à lire le chapitre [GC08] qui donne

un aperçu général du standard, le livre [Pia08] qui approfondi le contenu du standard, et enfin le standard lui-même. Celui-ci est disponible sur le site officiel<sup>10</sup>, et référencé (ainsi que les derniers ajouts et corrections) de manière plus explicite sur le site de David Clunie<sup>11</sup>.

## Conclusion

Dans ce chapitre ont été présentés et détaillés les principaux algorithmes de compression sans perte, presque sans perte et/ou scalables. Ces propriétés sont celles qui sont principalement visées par les contributions qui seront exposées dans la prochaine partie. Comme le confirme le standard de communication pour les images médicales de par les techniques de compression qu'il intègre, il existe peu d'algorithmes spécifiques à la compression d'images médicales 2D. Cependant les images volumiques étant assez spécifiques à ce milieu, quelques solutions pour leur compression visent davantage à répondre aux besoins pratiques de leur utilisation.

Parmi les algorithmes qui viennent d'être vus, les incontournables CALIC, et SPIHT ainsi que les standards JPEG-LS et JPEG-2000 seront pris comme références. Le modèle prédictif hiérarchique sera également étendu dans le prochain chapitre et IHINT servira de comparaison. L'approche prédictive pour la compression sans perte, ainsi que son extension en compression presque sans perte, seront utiles pour le chapitre 4 et également en fin du chapitre (5). Le modèle de compression par transformée concernera les deux derniers chapitres.

---

<sup>10</sup><http://medical.nema.org/>

<sup>11</sup><http://www.dclunie.com/dicom-status/status.html>



## Deuxième partie

# Techniques avancées pour la compression d'images médicales





## Chapitre 4

# Prédiction hiérarchique, orientée et adaptative

### Introduction

Comme il a pu être mentionné précédemment, les meilleurs résultats en compression sans perte et presque sans perte sont généralement obtenus à l'aide d'approches DPCM avancées. Elles utilisent des méthodes adaptatives qui s'appuient sur des modèles contextuels de l'information causale. Les algorithmes LOCO-I [WSS00], utilisé par le standard JPEG-LS, et CALIC [WM97] sont souvent pris comme références, y compris en imagerie médical [Clu00]. Cependant, de tels codeurs ne proposent pas une représentation scalable alors que celle-ci peut permettre de faciliter la consultation d'images à distance.

Dans le contexte de la compression sans perte, le codage scalable en résolution ou en débit est permis par les approches par transformées réversibles telles que la RDCT [WWJ<sup>+</sup>08] ou plus souvent les transformées en ondelettes entières (IWT), utilisées par exemple dans le mode sans perte du standard JPEG-2000 [TM01], ainsi que par les approches prédictives hiérarchiques par interpolation (HIP) telles que HINT [RVvDP88] ou IHINT [AAB97]. Afin de proposer une scalabilité en résolution ou en qualité, les coefficients transformés ou les données résiduelles peuvent être compressés à l'aide de codeurs par plan de bits tels que EBCOT [Tau00] ou SPIHT [SP96b], ou par des codeurs entropiques spécialisés utilisant un dictionnaire n-aire afin d'être plus optimaux pour une scalabilité en résolution uniquement.

Les schémas de lifting employés pour les IWT sont assez similaires à la prédiction par interpolation, mais permettent également de concentrer l'énergie dans les bandes basses résolutions. Ces bases d'ondelettes entières permettent d'obtenir une représentation multi-résolution anti-aliasée et ont un meilleur compromis débit/PSNR que les HIP lorsqu'elles sont utilisées avec un codeur scalable en qualité. Cependant il est plus difficile de les exploiter dans un contexte presque sans perte.

Dans ce chapitre est proposé un nouvel algorithme pour la compression sans perte

ou presque sans perte efficace d'images médicales 2D. Il utilise une prédiction hiérarchique orientée (HOP) qui combine l'approche DPCM avec l'approche HIP. Ainsi, cette approche originale ne permet d'effectuer qu'une scalabilité en résolution, mais exploite l'information des pixels déjà connus dans une sous-bande afin d'améliorer la décorrélation en comparaison aux IWT ou aux HIP.

Comme beaucoup d'approches DPCM, HOP s'appuie sur le contexte de prédiction afin de sélectionner un prédicteur linéaire orienté dont les coefficients sont fixés (prédicteurs statiques) puis effectue une correction des biais de prédiction pour s'adapter aux données. Il sera vu que ces prédicteurs statiques ne sont pas toujours efficaces sur les images les plus douces : images généralement moins bruitées, pour lesquelles les démarcations entre les divers éléments sont plus diffuses. C'est pourquoi une extension originale utilisant un support plus étendu pour la prédiction adaptative sera également étudiée. Afin de mieux tirer parti de ce support, la prédiction est optimisée au sens des moindres carrés.

Dans la première section de ce chapitre (sec. 4.1) est décrite l'approche de décorrélation hiérarchique. Après une brève description générale de l'algorithme, les modèles pour la prédiction statique et pour la prédiction optimisée sont détaillés, puis le modèle de correction des biais de prédiction analysé. Enfin, le processus utilisé pour permettre une extension presque sans perte est expliqué. La seconde section (sec. 4.2) détaille la méthode de compression utilisée pour les données décorrélées après avoir décrit un outil utilisé pour réduire le nombre de symboles nécessaires à représenter ces valeurs résiduelles. L'approche de compression hiérarchique est expliquée, puis le codeur entropique mis en place pour cette représentation est détaillé. La dernière section (sec. 4.3) commente les résultats obtenus pour la compression sans perte et presque sans perte d'images tomographiques et d'IRM.

Les contributions principales sont : i) l'approche prédictive hiérarchique orientée et adaptative et son extension pour les images douces (4.1.1 et 4.1.2), ii) une correction contextuelle des biais de prédiction à l'aide d'une nouvelle approche séquentielle (4.1.3), iii) la méthode de codage entropique utilisée (sec. 4.2).

## 4.1 Prédiction hiérarchique

Dans cette section est rappelé le fonctionnement de l'algorithme IHINT qui sera pris comme référence pour les schémas de prédiction hiérarchique classiques, et permet d'introduire l'approche hiérarchique de HOP. Les prédicteurs hiérarchiques de HOP sont ensuite détaillés.

### 4.1.1 Décomposition hiérarchique

Un niveau de prédiction de IHINT [AAB97] peut se résumer en deux étapes de prédiction, illustrées dans la FIG. 4.1. Si  $L$  est l'ensemble des pixels horizontalement pairs et  $H$  l'ensemble des pixels horizontalement impairs, la première étape (HStep)

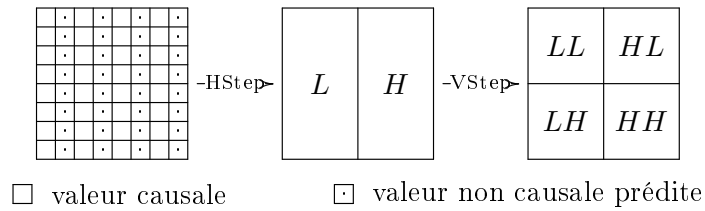


FIG. 4.1 – Un niveau de la prédiction hiérarchique de l’algorithme IHINT

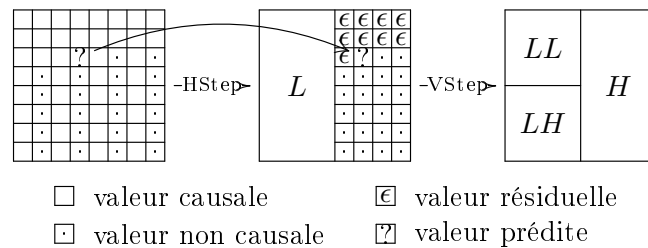


FIG. 4.2 – Un niveau de la prédiction hiérarchique de l’algorithme HOP

consiste à prédire les valeurs des pixels de  $H$  grâce à une interpolation horizontale des valeurs de  $L$  à l’aide d’un filtre à réponse impulsionnelle finie.  $H$  contient ensuite les résidus de cette prédiction. La seconde étape (VStep) consiste à transposer HStep pour l’appliquer verticalement, indépendamment sur  $L$  afin d’obtenir  $LL$  et  $LH$ , et sur  $H$  pour obtenir  $HL$  et  $HH$ .

L’ensemble  $LL$  contient alors une version sous échantillonnée, et donc de plus basse résolution, de l’image. Afin de décomposer une image hiérarchiquement, ce schéma peut être appliqué récursivement sur  $LL$ . Les valeurs ou résidus des pixels de chacun des sous-ensembles sont généralement réorganisés pour obtenir une représentation dyadique, afin de faciliter leur manipulation.

Pour les comparaisons qui suivront dans ce chapitre, la prédiction IHINT des pixels impairs est faite par la moyenne de ses deux voisins (pairs). Elle correspond au filtre passe haut des ondelettes 5/3 de LeGall [LGT88] également connues sous le nom d’ondelettes CDF 5/3 [CDF90], très fréquemment utilisées en compression sans perte à l’aide d’ondelettes, et en particulier par le mode sans perte de JPEG-2000. Ce filtre a été retenu pour sa similarité avec les prédicteurs statiques de HOP (voir l’équation (4.3) dans 4.1.2.1). De plus il est bien connu que cette interpolation linéaire permet d’effectuer une prédiction efficace.

Comme IHINT, un niveau de prédiction de HOP s’effectue en deux étapes (FIG. 4.2). La première (HStep) consiste à prédire les pixels horizontalement impairs à l’aide de l’ensemble des pixels déjà « connus » : l’approche proposée n’utilise pas seulement

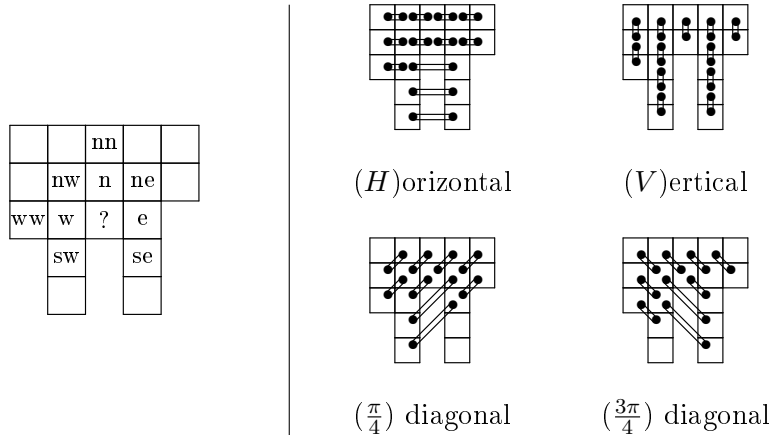


FIG. 4.3 – Motif de prédiction contextuel (à gauche) et liens entre les pixels utilisés pour l'estimation du gradient (à droite)

les pixels pairs, elle peut également tirer avantage de tous ceux précédemment prédits dans le sous ensemble  $H$  (qui sont désormais des valeurs causales) comme pour une prédiction DPCM. Contrairement à IHINT, la seconde étape (VStep) ne s'applique que sur le sous-ensemble  $L$ , les valeurs de  $H$  étant suffisamment décorréliées, et permet aussi d'obtenir une représentation sous échantillonnée  $LL$  de l'image, pouvant également être décomposée récursivement.

## 4.1.2 Prédicteurs hiérarchiques orientés

### 4.1.2.1 Prédicteurs statiques

HOP est principalement conçu pour des images bruitées, contenant des objets structurés avec des contours bien marqués (images natives contrastées). Inspiré par le prédicteur efficace de CALIC, GAP (*gradient adjusted predictor*), une estimation de l'orientation est effectuée à l'aide du motif de la FIG. 4.3. Tout d'abord, la moyenne des valeurs absolues des différences locales pour chacune des orientations appartenant à l'ensemble  $\Omega = \{H, V, \frac{\pi}{4}, \frac{3\pi}{4}\}$  est calculée :

$$d_\omega = \frac{1}{\text{Card } D_\omega} \sum_{(x_i, x_j) \in D_\omega} \frac{|I(x_i) - I(x_j)|}{\|x_i - x_j\|}, \quad (4.1)$$

avec  $I$  l'image,  $\omega \in \Omega$  et  $D_\omega$  défini par l'ensemble des couples de pixels reliés dans la FIG. 4.3. Ensuite, en utilisant un seuil de bruit  $T_{\text{Noise}}$  (défini plus tard dans 4.1.2.3),

l'orientation du prédicteur  $o \in O = \{N, H, V, \frac{\pi}{4}, \frac{3\pi}{4}\}$  est sélectionnée :

$$o = \begin{cases} N, & \text{si } \max\left(\left|d_{\frac{\pi}{4}} - d_{\frac{3\pi}{4}}\right|, |d_V - d_H|\right) \leq T_{\text{Noise}}, \\ \operatorname{argmin}_{x \in \{\frac{\pi}{4}, \frac{3\pi}{4}\}} d_x, & \text{sinon, si } \left|d_{\frac{\pi}{4}} - d_{\frac{3\pi}{4}}\right| > |d_V - d_H|, \\ \operatorname{argmin}_{x \in \{H; V\}} d_x, & \text{sinon.} \end{cases} \quad (4.2)$$

Elle permet de choisir parmi cinq prédicteurs : ( $N$ )on-orienté, ( $V$ )ertical, ( $H$ )orizontal, ( $\frac{\pi}{4}$ ) diagonal, et ( $\frac{3\pi}{4}$ ) diagonal. La valeur prédite est alors,

$$\hat{x} = \begin{cases} n & \text{si } o = V, \\ (w + e)/2 & \text{si } o = H, \\ (sw + ne)/2 & \text{si } o = \pi/4, \\ (nw + se)/2 & \text{si } o = 3\pi/4, \end{cases} \quad (4.3)$$

afin d'effectuer une prédiction orientée le long des contours, et

$$\hat{x} = \frac{n + w + e + \frac{1}{\sqrt{2}}(nw + ne + sw + se)}{3 + \frac{4}{\sqrt{2}}}, \quad \text{si } o = N, \quad (4.4)$$

pour qu'elle soit estimée à l'aide d'un filtre plus robuste dans les régions homogènes qui contiennent majoritairement du bruit.

L'activité locale pour l'orientation du prédicteur,

$$A = \begin{cases} (d_H + d_V + d_{\frac{\pi}{4}} + d_{\frac{3\pi}{4}})/4 & \text{si } o = N, \\ d_o & \text{sinon,} \end{cases} \quad (4.5)$$

sera utilisée par la suite pour le choix de prédicteurs adaptatifs (4.1.2.2) et pour la correction des biais de prédiction (4.1.3).

#### 4.1.2.2 Prédicteurs adaptatifs optimisés au sens des moindres carrés

La prédiction de HOP n'est pas vraiment efficace sur les images douces parce que les petits supports de prédiction utilisés dans (4.3) ne sont pas très adaptés pour la décorrélation d'une information trop diffuse (en particulier le prédicteur vertical). Puisqu'un prédicteur d'ordre supérieur devrait permettre d'obtenir une meilleure estimation sur ces images (à l'aide d'un support de prédiction plus étendu), deux nouvelles approches de prédiction sont introduites : HOP-LSE et HOP-LSE<sup>+</sup>. Afin de construire  $\hat{x}$ , elles utilisent une combinaison linéaire sur un ensemble, légèrement étendu en comparaison aux prédicteurs statiques qui viennent d'être présentés, de pixels causaux pour effectuer la prédiction (voir FIG. 4.4). Dans le but d'exploiter efficacement ces nouveaux supports, les prédicteurs sont construits de manière dynamique à l'aide d'une estimation au sens des moindres carrés qui permet de mieux les adapter aux spécificités de chaque image.

Pour HOP-LSE<sup>+</sup>, l'optimisation est toujours effectuée, tandis que pour HOP-LSE elle n'est utilisée que si  $o \neq N$ . Dans le cas contraire, le prédicteur statique non-orienté

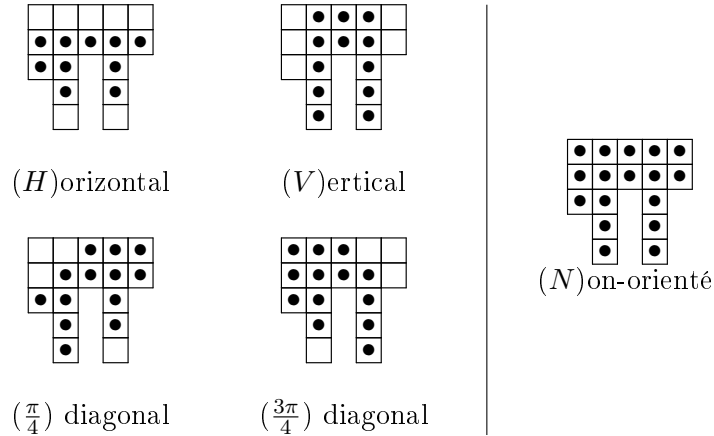


FIG. 4.4 – Ensembles de pixels causaux  $\sigma_o$  utilisés pour HOP-LSE (à gauche), et pour HOP-LSE<sup>+</sup> (à gauche et à droite).

de l'équation (4.4) est utilisé. Cela permet de réduire la complexité en effectuant le calcul des coefficients de prédiction uniquement dans les zones non homogènes. Sur l'ensemble des bases d'images CT et IRM utilisées, cela ne représente que 30% des pixels et permet alors, avec des implémentations non optimisées, de diviser le temps d'exécution par un facteur 3 environ en comparaison à HOP-LSE<sup>+</sup>.

Afin de calculer différents prédicteurs en fonction de l'activité locale,  $A$  (4.5) est quantifié logarithmiquement

$$a = Q^{\log_2(A)}, \quad (4.6)$$

où  $Q^{\log_2(A)}$  donne l'index du bit de poids fort de  $A$ . Un contexte de prédiction, qui dépend à la fois de l'activité et de l'orientation locale  $\Pi = (a, o)$  peut alors être sélectionné.

Soit  $x_{\Pi,t+1}$  la valeur du pixel à prédire, où  $\Pi$  est le contexte sélectionné pour la prédiction et  $t$  est le nombre de fois que  $\Pi$  a été précédemment sélectionné comme contexte de prédiction. Soit  $\psi_{\Pi,t+1}$  le vecteur colonne qui contient les valeurs des pixels causaux appartenant à l'ensemble  $\sigma_o$ . La prédiction sera :

$$\hat{x} = \langle \lambda_{\Pi,t}, \psi_{\Pi,t+1} \rangle, \quad (4.7)$$

où  $\lambda_{\Pi,t}$  est le vecteur contenant les coefficients de la combinaison linéaire de prédiction. Ce vecteur est obtenu par la solution du problème de minimisation de l'erreur quadratique moyenne de prédiction des  $(x_{\Pi,k})_{k \leq t}$ , et se calcule à l'aide de la forme bien connue :

$$\lambda_{\Pi,t} = \left( (\Psi_{\Pi,t})^\top \Psi_{\Pi,t} \right)^{-1} (\Psi_{\Pi,t})^\top \chi_{\Pi,t}, \quad (4.8)$$

où  $\Psi_{\Pi,t} = [\psi_{\Pi,1}, \dots, \psi_{\Pi,t}]^\top$ ,  $\chi_{\Pi,t} = [x_{\Pi,1}, \dots, x_{\Pi,t}]^\top$ .

Parce que cette solution nécessite que  $\Psi^\top \Psi$  soit non singulière et parce qu'il est nécessaire d'avoir un nombre suffisant de  $\psi$  et  $x$  pour être efficace, tant que  $t \leq T$  ( $T$  a

été fixé à 64 pour les expérimentations), et tant que  $\Psi^\top \Psi$  est proche de la singularité, les prédicteurs statiques associés à  $o$ , décrits précédemment, sont utilisés.

Afin de ne pas conserver une grande matrice  $\Psi_{\Pi,t}$  et un vecteur  $\chi_{\Pi,t}$  de tailles incrémentielles, pour chacun des contextes, l'implémentation est effectuée en mettant à jour itérativement la matrice définie positive  $AC_{\Pi,t} = (\Psi_{\Pi,t})^\top \Psi_{\Pi,t}$  (auto-corrélation des pixels de prédiction non centrés) :

$$AC_{\Pi,t+1} = AC_{\Pi,t} + \psi_{\Pi,t+1} (\psi_{\Pi,t+1})^\top \quad (4.9)$$

ainsi que le vecteur  $CC_{\Pi,t} = (\Psi_{\Pi,t})^\top \chi_{\Pi,t}$  (cross-corrélation entre les pixels de prédiction et les valeurs prédites non centrés) :

$$CC_{\Pi,t+1} = CC_{\Pi,t} + \tilde{x}_{\Pi,t+1} \psi_{\Pi,t+1}, \quad (4.10)$$

avec  $\tilde{x}_{\Pi,t+1} = x_{\Pi,t+1}$  en mode sans perte et  $\tilde{x}_{\Pi,t+1}$  la valeur reconstruite de  $x_{\Pi,t+1}$  si une quantification est utilisée (voir 4.1.4).

### 4.1.2.3 Seuil $T_{\text{Noise}}$

Plutôt que d'effectuer une optimisation du seuil  $T_{\text{Noise}}$ , qui pourrait s'avérer coûteuse en temps, celui-ci est initialisé à partir d'une estimation du bruit prise comme étant le seuil de quantification dans la formule de Donoho [Don95]. Ce seuil est calculé sur les hautes fréquences de la transformée orthonormale de Haar de l'image à compresser. Pour HOP et HOP-LSE<sup>+</sup>, même si cette estimation n'est pas toujours optimale pour l'efficacité de compression, elle permet d'améliorer légèrement les résultats, jusqu'à environ 1% pour HOP sur les images bruitées et pour HOP-LSE<sup>+</sup> sur les images douces. Pour les CT bruitées, aucune amélioration n'est apportée avec HOP-LSE<sup>+</sup> à cause des artefacts de reconstruction (pouvant être assimilés à du bruit) qui varient avec la localisation spatiale et dont les propriétés sont donc difficiles à capturer avec le processus d'optimisation retenu. La TAB. 4.1 donne les résultats obtenus avec et sans le seuil. Les résultats pour HOP-LSE ne sont pas présentés, car HOP-LSE n'a pas de sens sans le seuil  $T_{\text{Noise}}$  : il serait alors équivalent à HOP-LSE<sup>+</sup>, le processus d'optimisation des prédicteurs serait appliqué pour tous les pixels et le prédicteur statique non-orienté caractérisant HOP-LSE<sup>+</sup> ne serait jamais utilisé.

Pour éviter son calcul sur chaque image, le seuil  $T_{\text{Noise}}$  pourrait être optimisé hors-ligne en fonction du matériel et des paramètres d'acquisition par exemple.

## 4.1.3 Correction des biais de prédiction

### 4.1.3.1 Bref rappel

Afin d'éviter les erreurs systématiques, aussi appelées biais, de prédiction qui arrivent généralement dans une configuration particulière, la technique usuelle consiste à sélectionner un contexte permettant de corriger la prédiction à l'aide de la moyenne  $\mu^\epsilon(C(\hat{x}))$  des erreurs précédemment commises dans le même contexte :

$$\hat{x} \leftarrow \hat{x} + \mu^\epsilon(C(\hat{x})). \quad (4.11)$$



TAB. 4.1 – Impact du seuil  $T_{\text{Noise}}$  sur la compression (bpp)

utilisation de $T_{\text{Noise}}$	HOP		HOP-LSE <sup>+</sup>	
	non	oui	non	oui
MeDEISA CT	5.232	5.201	4.591	4.543
VHP-Male CT	4.889	4.861	4.894	4.897
VHP-Male MRI	4.908	4.855	4.749	4.741

$C(\hat{x})$  peut être choisi à partir de n'importe quel calcul causal. Par exemple, LOCO-I quantifie le gradient entre les voisins causaux du pixel à prédire, et CALIC utilise également la valeur de prédiction  $\hat{x}$  afin de définir un modèle textuel. Dans [US08], Ulaça et Stasinski étendent cette approche en cumulant plusieurs corrections à l'aide d'une moyenne pondérée  $\alpha_k \mu_k^\epsilon(C_k(\hat{x}))$  des erreurs survenues pour différents modèles de sélection de contextes :

$$\hat{x} \leftarrow \hat{x} + \sum_k \alpha_k \mu_k^\epsilon(C_k(\hat{x})), \quad (4.12)$$

et montrent des améliorations de la compression.

#### 4.1.3.2 Correction séquentielle de l'erreur contextuelle (SCEC)

Il est proposé d'étendre (4.12) en utilisant une correction séquentielle de l'erreur contextuelle (SCEC). Pour  $k = 1$  jusqu'à  $K$ , elle applique séquentiellement le schéma de correction :

$$\hat{x}_k \leftarrow \hat{x}_{k-1} + \alpha_k \mu_k^\epsilon(C_k(\hat{x}_{k-1})), \quad (4.13)$$

avec  $\hat{x}_0 = \hat{x}$ . La valeur de correction finale est alors  $\hat{x} \leftarrow \lfloor \hat{x}_K \rfloor$ , où  $\lfloor \cdot \rfloor$  est l'arrondi à l'entier le plus proche. L'erreur de prédiction  $\epsilon = x - \hat{x}$  est ensuite utilisée pour mettre à jour chaque  $\mu_k^\epsilon(C_k(\hat{x}_{k-1}))$ . Lorsque la sélection de contexte ne dépend pas de la valeur de prédiction intermédiaire  $\hat{x}_{k-1}$ , (4.13) est équivalent à (4.12), mais lorsqu'elle en dépend, la prédiction est successivement affinée, et permet de sélectionner des contextes de plus en plus fiables. Pour les expérimentations, les coefficients  $\alpha_k$  sont simplement fixés à  $1/K$ .

La sélection de contexte mise en place pour HOP est maintenant décrite. Puisque les approches orientées ne sont pas toujours efficaces dans les régions texturées, les  $C_k$ , dépendant de  $\hat{x}_{k-1}$ , sont définis en retenant le procédé de capture de la texture utilisé par CALIC. L'information textuelle locale  $\{t_0, \dots, t_4\} = \{n, w, e, 2n - nn, 2w - ww\}$  est binarisée afin de définir un nombre sur 5 bits  $B = b_4 b_3 \dots b_0$ , avec  $b_i = 1$  si  $t_i \geq \hat{x}_{k-1}$ , et  $b_i = 0$  sinon.  $B$  est combiné avec l'activité locale quantifiée  $a$  (4.6), et avec le numéro de l'orientation utilisée : 0 pour les zones homogènes, 1 pour horizontale, 2 pour verticale, 3 pour  $\frac{\pi}{4}$  diagonale, et 4 pour  $\frac{3\pi}{4}$  diagonale.

### 4.1.3.3 Validation de la SCEC

Quelques résultats, obtenus avec la SCEC quand  $K$  varie, sont reportés dans la TAB. 4.2. L'analyse est faite pour HOP, HOP-LSE et HOP-LSE<sup>+</sup> en mode sans perte (PAE=0) et presque sans perte (PAE=4, c.f. 4.1.4). La discussion sur les performances des prédicteurs est laissée pour la section 4.3.

Avec les prédicteurs statiques (HOP), la SCEC permet toujours d'améliorer la compression de façon asymptotique, avec un mieux pour les images douces. Pour exemple, sur les coupes CT filtrées de MeDEISA le débit de la compression sans perte est réduit (en comparaison à l'absence de correction :  $K=0$ ) de 3.1% en utilisant seulement un contexte de correction ( $K=1$ ) et de 3.55% pour une séquence de cinq corrections ( $K=5$ ), alors que sur les IRM natives bruitées de VHP-Male l'amélioration est de 0.64% quand  $K=1$  et 0.91% lorsque  $K=5$ . Sur les images CT natives de VHP-Male, pour lesquelles le bruit est davantage dépendant de la localisation spatiale, le débit n'est réduit que de 0.51% ( $K=1$ ) à 0.76% ( $K=5$ ). En mode presque sans perte, les bénéfices sur la compression sont meilleurs encore, avec un débit réduit jusqu'à ( $K=4$ ) 6.25% sur les coupes tomographiques de MeDEISA, et 1.35% sur les CT et 1.87% sur les IRM de VHP-Male. Pour ce qui est des distorsions en mode NLS, seul le premier contexte de correction semble améliorer le PSNR. Pour  $K > 1$ , le débit mais également le PSNR sont réduits, conservant un compromis débit/distorsion à peu près similaire.

Avec les prédicteurs dynamiques (HOP-LSE et HOP-LSE<sup>+</sup>), les mêmes observations peuvent être faites sur les tomographies. Cependant sur les IRM, lorsque  $K = 1$ , la correction des biais semble échouer. Ceci peut s'expliquer par le fait que les dimensions des coupes sont plus petites ( $256 \times 256$ ) et par la nature très contrastée des IRM. Lors de la décomposition, les plus petites résolutions sont aliasées à cause du sous-échantillonnage : les pixels sont alors moins corrélés et l'optimisation au sens des moindres carrés est alors plus difficile. La prédiction devient donc moins bonne et l'annulation des biais qui s'effectue sur des résultats très aléatoires n'est pas correcte à cause du manque de données pour estimer correctement les statistiques. Ceci pourrait être amélioré en désactivant la correction pour chaque contexte tant que celui-ci n'a pas été suffisamment sélectionné et mis à jour pour être stable. Cependant, vu que  $K > 1$  permet de contourner cet inconvénient (en moyennant les erreurs de différents contextes) et d'améliorer les résultats, aucun seuil de ce type n'a été utilisé.

Les résultats montrent que la SCEC permet de réduire la taille des images compressées, aussi bien en sans perte qu'en presque sans perte. En mode NLS, la SCEC a peu d'impact sur le compromis débit distorsion avec les prédicteurs statiques, mais avec ceux optimisés dynamiquement, il est préférable que  $K$  soit nul ou strictement supérieur à 1. Pour toutes les autres expérimentations,  $K$  est alors toujours fixé à 3.

		PAE=0 (BPP)						
		$K$	0	1	2	3	4	5
HOP	MeDEISA CT		<u>5.201</u>	5.040	5.021	5.019	<b>5.017</b>	<b>5.017</b>
	VHP-Male CT		<u>4.861</u>	4.836	4.828	4.826	4.825	<b>4.824</b>
	VHP-Male MRI		<u>4.855</u>	4.824	4.815	4.813	4.812	<b>4.811</b>
HOP-LSE	MeDEISA CT		<u>4.767</u>	4.695	4.681	4.677	4.676	<b>4.675</b>
	VHP-Male CT		<u>4.854</u>	4.835	4.824	4.821	4.819	<b>4.818</b>
	VHP-Male MRI		<u>4.784</u>	4.777	4.768	4.765	<b>4.764</b>	<b>4.764</b>
HOP-LSE+	MeDEISA CT		<u>4.543</u>	4.525	4.511	4.508	4.507	<b>4.506</b>
	VHP-Male CT		<u>4.897</u>	4.864	4.844	4.837	4.833	<b>4.830</b>
	VHP-Male MRI		4.741	<u>4.744</u>	4.734	4.731	4.730	<b>4.729</b>

		PAE=4 (BPP/PSNR)					
		$K$	0	1	2	3	4
HOP	MeDEISA CT		<u>2.654</u> / <u>64.756</u>	2.503 / <b>64.856</b>	2.492 / 64.854	2.489 / 64.850	<b>2.488</b> / 64.847
	VHP-Male CT		<u>2.437</u> / <u>64.953</u>	2.413 / <b>65.033</b>	2.407 / 65.023	2.405 / 65.016	<b>2.404</b> / 65.011
	VHP-Male MRI		<u>1.921</u> / <u>64.478</u>	1.898 / 64.479	1.889 / 64.479	1.887 / 64.479	<b>1.885</b> / <b>64.480</b>
HOP-LSE	MeDEISA CT		<u>2.288</u> / <u>64.836</u>	2.228 / <b>64.873</b>	2.218 / 64.868	2.215 / 64.866	<b>2.214</b> / 64.862
	VHP-Male CT		<u>2.413</u> / <u>65.004</u>	2.399 / <b>65.044</b>	2.392 / 65.035	2.390 / 65.029	<b>2.389</b> / 65.025
	VHP-Male MRI		1.855 / <b>64.485</b>	<u>1.857</u> / 64.483	1.848 / 64.482	1.845 / 64.483	<b>1.844</b> / <u>64.482</u>
HOP-LSE+	MeDEISA CT		<u>2.149</u> / 64.850	2.128 / <b>64.879</b>	2.119 / 64.875	2.116 / 64.873	<b>2.115</b> / 64.869
	VHP-Male CT		<u>2.398</u> / <u>65.033</u>	2.384 / <b>65.054</b>	2.376 / 64.045	2.374 / 64.038	<b>2.373</b> / 65.035
	VHP-Male MRI		1.838 / <b>64.480</b>	<u>1.841</u> / 64.477	1.831 / <u>64.476</u>	1.828 / <u>64.476</u>	<b>1.827</b> / 64.477

Les meilleurs résultats sont en gras, et les moins bons sont soulignés.

TAB. 4.2 – Résultats de compression pour différentes longueurs ( $K$ ) de la SCEC.

#### 4.1.4 Extensions presque sans perte

Les extensions presque sans perte de IHINT et de HOP sont détaillées ici. (c.f. 3.1.4 pour les notions élémentaires)

##### 4.1.4.1 IHINT presque sans perte

Soit  $P(S_e, S_o)$  permettant d'obtenir les résidus de l'ensemble  $S_o$  lorsque celui-ci est prédit à l'aide de  $S_e$ .

Pour étendre IHINT en presque sans perte (c.f. 4.1.1 et FIG. 4.1), après avoir organisé les pixels de l'image pour former la pyramide dyadique, l'approche commence par  $\tilde{Q}_\delta$  approximer la bande  $LL$  en  $\tilde{LL}$  du plus petit niveau de décomposition. Ensuite, une descente pyramidale est effectuée, en commençant par cette plus petite résolution :  $\tilde{LL}$  est utilisée pour prédire  $LH$  et  $HL$  afin d'obtenir leurs résidus  $\tilde{Q}_\delta$  approximatés  $\tilde{\epsilon}_{LH}$  et  $\tilde{\epsilon}_{HL}$ . La prédiction de  $\tilde{\epsilon}_{LH}$  est inversée pour construire  $\tilde{LH}$ , et permet de calculer  $\tilde{\epsilon}_{HH} = \tilde{Q}_\delta \left( P \left( \tilde{\epsilon}_{HL}, P \left( \tilde{LH}, HH \right) \right) \right)$ . En inversant les étapes de prédiction,  $\tilde{HH}$  et  $\tilde{HL}$  sont récupérés. Enfin, les données de  $\tilde{LL}$ ,  $\tilde{LH}$ ,  $\tilde{HL}$  et  $\tilde{HH}$  sont réorganisées pour former l'image NLS de la résolution supérieure qui peut ensuite servir de sous-bande  $\tilde{LL}$  pour le niveau de résolution suivant. Ce schéma est itéré jusqu'à l'obtention de l'image NLS pleine résolution.

Lorsque l'image NLS pleine résolution est ainsi obtenue, la prédiction hiérarchique (qui reste exactement la même que pendant la phase de descente de la pyramide) est appliquée dans l'ordre inverse (de la plus haute à la plus basse résolution, comme pour la compression sans perte). Les valeurs résiduelles et l'image de plus basse résolution sont à nouveau  $Q_\delta$  quantifiées, permettant ainsi d'obtenir exactement les mêmes  $\epsilon_\delta$  que lors de la descente pyramidale.

##### 4.1.4.2 HOP presque sans perte

L'extension NLS de HOP fusionne les lignes directrices des approches hiérarchiques et DPCM : l'approximation  $\tilde{Q}_\delta$  de l'image de plus basse résolution est utilisée pour prédire celles des résolutions suivantes, la prédiction étant toujours effectuée à partir de valeurs reconstruites. Pour chacun des niveaux croissants de résolution, l'étape VStep (voir FIG. 4.2) sert à construire la bande totalement NLS  $\tilde{L}$ , qui permet ensuite d'effectuer la prédiction NLS de la bande  $H$ .

Comme pour IHINT, lorsque l'image pleine résolution est obtenue, la décomposition est ré-appliquée, dans le même ordre que pour la compression sans perte, et les valeurs résiduelles ainsi que la plus basse résolution sont à nouveau  $Q_\delta$  quantifiées.

Pour IHINT comme pour HOP, la décomposition presque sans perte, requise avant la compression, nécessite l'approximation NLS (descente pyramidale) suivie par la décomposition (reconstruction de la pyramide). La complexité temporelle est donc à peu près

deux fois celle de la décomposition sans perte qui ne nécessite que la construction pyramidale. Une autre approche pour l'implémentation utilisant un stockage temporaire des résidus quantifiés obtenus pendant la descente pyramidale permettrait une complexité temporelle similaire au mode presque sans perte, mais avec un coût mémoire supplémentaire égal à la taille de l'image.

## 4.2 Compression

### 4.2.1 Réorganisation des valeurs résiduelles

Connaissant  $I_{\min} = \min(I)$  et  $I_{\max} = \max(I)$  (avec  $I$  l'image), la valeur de prédiction  $\hat{x}$  peut être forcée à rester dans l'intervalle  $[I_{\min}, I_{\max}]$  : si  $\hat{x} < I_{\min}$  respectivement  $\hat{x} > I_{\max}$ , elle est forcée à valoir  $\hat{x} \leftarrow I_{\min}$  resp.  $\hat{x} \leftarrow I_{\max}$ .

Le résidu de la prédiction (éventuellement quantifié)  $\check{\epsilon}_\delta = Q_\delta(x - \hat{x})$  est alors borné par  $\check{R}_{\min} = Q_\delta(I_{\min} - \hat{x})$  et  $\check{R}_{\max} = Q_\delta(I_{\max} - \hat{x})$ . Il peut donc se voir ré-attribuer une valeur à l'aide d'une fonction bijective :

$$\text{Remap}_{\hat{x}}(\check{\epsilon}_\delta) : \begin{matrix} [\check{R}_{\min} : \check{R}_{\max}] \\ \check{\epsilon}_\delta \end{matrix} \mapsto \begin{matrix} [0 : \check{R}_{\max} - \check{R}_{\min}] \\ \bar{\check{\epsilon}}_\delta \end{matrix} \quad (4.14)$$

Cette technique est souvent utilisée par les codeurs prédictifs afin de réduire la taille de l'alphabet d'un facteur 2, en comparaison aux valeurs résiduelles possibles sur l'ensemble de l'image  $[Q_\delta(I_{\min} - I_{\max}) : Q_\delta(I_{\max} - I_{\min})]$ . L'entrelacement des valeurs résiduelles initialement positives et négatives permet d'obtenir une PDF qui tend à être strictement décroissante sur l'ensemble des résidus (et peu permettre de légèrement réduire l'entropie) lorsque  $\text{Remap}_{\hat{x}}(\check{\epsilon}_\delta)$  est défini comme (voir également FIG. 4.5) :

$$\text{Remap}_{\hat{x}}(\check{\epsilon}_\delta) = \begin{cases} \check{\epsilon}_\delta - \check{R}_{\min} & \text{si } s \geq 0 \text{ et } \check{\epsilon}_\delta + \check{R}_{\min} > 0 \\ \check{R}_{\max} - \check{\epsilon}_\delta & \text{si } s < 0 \text{ et } \check{\epsilon}_\delta + \check{R}_{\max} < 0 \\ 2|\check{\epsilon}_\delta| & \text{sinon si } s \cdot \check{\epsilon}_\delta \geq 0 \\ 2|\check{\epsilon}_\delta| - 1 & \text{sinon si } s \cdot \check{\epsilon}_\delta < 0 \end{cases} \quad (4.15)$$

avec  $s = \underset{-1,1}{\text{sign}}(\check{R}_{\max} + \check{R}_{\min})$ .

Dans les résultats présentés, la méthode de réorganisation des résidus  $\text{Remap}_{\hat{x}}(\check{\epsilon}_\delta)$  est utilisée à la fois pour IHINT et pour HOP.

### 4.2.2 Codage progressif en résolution

Avant la compression, la décomposition est faite complètement. La valeur de la plus basse résolution de l'image, qui est alors un unique pixel, est tout d'abord transmise. Pour permettre une transmission scalable en résolution, la décomposition est compressée en partant de la plus basse résolution. Pour IHINT, les bandes  $\bar{\check{\epsilon}}_\delta(LH)$ ,  $\bar{\check{\epsilon}}_\delta(HL)$  puis  $\bar{\check{\epsilon}}_\delta(HH)$ , pour HOP les bandes  $\bar{\check{\epsilon}}_\delta(LH)$  puis  $\bar{\check{\epsilon}}_\delta(H)$ , contenant toutes des valeurs résiduelles quantifiées et réorganisées, sont séquentiellement codées, ligne par ligne, colonne

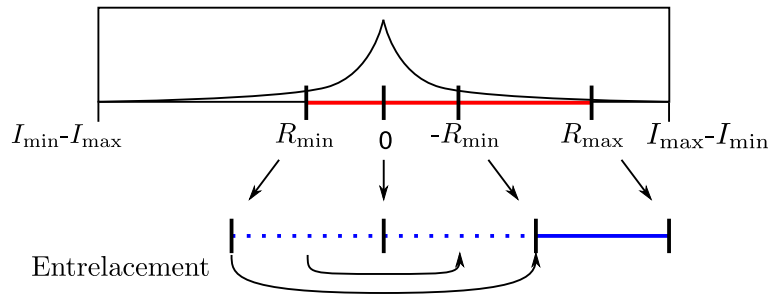


FIG. 4.5 – Réorganisation des valeurs résiduelles

par colonne. Un contexte de codage  $CC(\sigma)$  est sélectionné par quantification logarithmique de la moyenne de l'amplitude résiduelle locale, calculée sur le voisinage causal intra et inter-bande du symbole  $\sigma$  à compresser.  $CC$  permet d'effectuer le codage entropique de  $\sigma$  à l'aide d'une PDF itérativement calculée chaque fois qu'un symbole a précédemment été codé dans le même contexte.

A partir d'une résolution minimale de  $32 \times 32$  (largeur  $\geq 32$  et hauteur  $\geq 32$ ), ces statistiques et les codeurs associés sont réinitialisés à leur état initial avant la compression de chaque nouveau niveau de résolution. Ceci permet de décompresser indépendamment une petite image, et chacune de ses extensions successives.

Cette méthode de codage multi-résolution peut être utilisée pour accélérer la navigation dans une grosse banque d'images, par exemple, en ne transmettant et n'affichant que des imageries (résumant l'information contenue dans les images en pleine résolution) afin de permettre une sélection rapide et facile de celles souhaitées. Il n'est alors nécessaire de transmettre et décoder que l'information requise pour reconstruire les plus hautes résolutions.

Afin d'être efficace avec la contrainte de scalabilité en résolution, le codeur entropique se doit d'estimer rapidement et efficacement la PDF des données à compresser. Dans la littérature il existe des codeurs entropiques avec une adaptation rapide, reposant sur des modèles de PDF usuels (Gaussienne généralisée, Laplacienne, Géométrique, ...) et ne nécessitant ainsi que de l'ajustement de quelques paramètres pour estimer la PDF des données. Mais évidemment, ces codeurs sont moins efficaces que si une modélisation plus fine de la PDF était utilisée. Le problème d'utiliser un modèle plus fin est qu'il nécessite plus de paramètres et requiert donc plus de données d'« apprentissage » pour pouvoir être correctement ajusté.

L'approche proposée pour HOP est une combinaison des deux techniques qui viennent d'être mentionnées. Elle utilise un ensemble restreint de paramètres pour rapidement estimer la PDF et avoir une bonne efficacité de codage, et un ensemble de paramètres plus large pour une adaptation plus lente mais plus adaptée à la véritable distribution des données. En fait, ceci est fait avec l'aide de deux codeurs distincts qui sont employés dans une compression en deux phases, illustrées dans la FIG. 4.6. Un codeur entropique

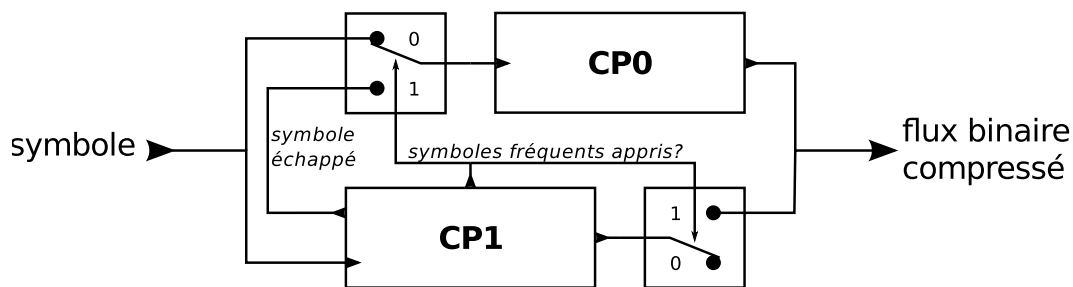


FIG. 4.6 – Codeur en deux phases

à paramètres restreints (codeur de la phase 0 : CP0) permet l'approximation rapide de la distribution des données et compresse les premiers coefficients, tandis qu'un autre codeur entropique (codeur de la phase 1 : CP1) apprend des statistiques plus fines de la distribution. Lorsque CP1 estime que ses statistiques sont suffisamment correctes, il est utilisé à la place de CP0 pour la compression des coefficients suivants, tout en continuant de mettre à jour son modèle de distribution. Lorsqu'une valeur de coefficient est inconnue de CP1, un symbole d'échappement est envoyé et CP0 est utilisé pour coder cette valeur. Cet algorithme est davantage détaillé dans la sous-section 4.2.4.

### 4.2.3 Validation de l'approche pour la scalabilité en résolution

La TAB. 4.3 illustre quelques résultats de compression obtenus pour l'ensemble des résolutions disponibles au décodage de trois images de l'annexe A. L'efficacité de HOP, HOP-LSE et HOP-LSE<sup>+</sup> est confrontée à IHINT et JPEG-2000 (J2K). La comparaison de l'ensemble des résultats, obtenus en compression sans perte, avec ceux de J2K qui utilise une technique de référence pour la compression multi-échelle, montre que l'approche présentée est plutôt bien adaptée pour la scalabilité.

Pour le mode sans perte, même si il est difficile de comparer J2K avec IHINT à cause de la compaction d'énergie des ondelettes, le codage entropique proposé semble permettre d'obtenir des performances similaires voir meilleures en terme de scalabilité. Pour les petites résolutions, aussi bien en sans perte qu'en presque sans perte, les approches prédictives orientées (HOP, HOP-LSE et HOP-LSE<sup>+</sup>) sont plus efficaces que celle non-orientée de IHINT. HOP obtient également de meilleurs résultats que HOP-LSE et HOP-LSE<sup>+</sup> qui peuvent s'expliquer par le manque de données pour une optimisation efficace des prédicteurs au sens des moindres carrés sur les données aliasées (par le sous échantillonnage) des images basses résolutions. Cette remarque pourrait être prise en compte afin d'améliorer légèrement la compression en désactivant l'optimisation sur les plus basses résolutions.

### 4.2.4 Détails du codeur entropique

Pour le codeur de la phase 0 (CP0), qui effectue la compression avec un nombre limité de paramètres pour un alphabet de taille potentiellement infinie, chaque symbole

Sans perte Image douce (CT) : A.1-(e)						Presque sans perte PAE=4 Image douce (CT) : A.1-(e)					
dim	32	64	128	256	512	32	64	128	256	512	
	(bpp)	(bppa)	(bppa)	(bppa)	(bppa)	(bpp)	(bppa)	(bppa)	(bppa)	(bppa)	
J2K	<u>11.68</u>	<u>8.35</u>	<u>7.34</u>	<u>6.38</u>	4.26	4.88	-	-	-	-	
IHINT	9.94	8.16	7.13	6.08	4.34	4.86	<u>6.73</u>	<u>5.20</u>	<u>4.36</u>	<u>3.43</u>	
HOP	<b>9.52</b>	<b>7.83</b>	<b>6.81</b>	6.00	<u>4.66</u>	<u>5.07</u>	<b>6.39</b>	<b>4.93</b>	<b>4.10</b>	3.38	
HOP-LSE	<b>9.52</b>	7.85	6.84	<b>5.96</b>	4.20	4.72	6.40	4.95	4.12	3.35	
HOP-LSE <sup>+</sup>	<b>9.52</b>	7.85	6.84	<b>5.96</b>	<b>3.96</b>	<b>4.54</b>	6.40	4.95	4.13	<b>3.34</b>	
										1.92	
										2.37	
										<u>2.16</u>	
										<u>2.53</u>	
										1.79	
										2.24	
										<b>2.13</b>	

CT native bruitée : A.1-(k)						CT native bruitée : A.1-(k)					
dim	32	64	128	256	512	32	64	128	256	512	
	(bpp)	(bppa)	(bppa)	(bppa)	(bppa)	(bpp)	(bppa)	(bppa)	(bppa)	(bppa)	
J2K	<u>10.71</u>	7.05	6.04	<u>5.50</u>	<u>4.74</u>	4.99	-	-	-	-	
IHINT	9.34	<u>7.16</u>	<u>6.18</u>	5.45	4.70	4.96	<u>6.32</u>	<u>4.51</u>	<u>3.72</u>	<u>3.12</u>	
HOP	<b>8.89</b>	<b>6.74</b>	<b>5.87</b>	<b>5.21</b>	4.59	4.81	<b>5.89</b>	<b>4.15</b>	<b>3.46</b>	<b>2.92</b>	
HOP-LSE	8.90	6.78	5.90	5.22	4.54	4.78	5.90	4.18	3.50	2.93	
HOP-LSE <sup>+</sup>	8.90	6.79	5.91	5.22	<b>4.52</b>	<b>4.76</b>	5.90	4.19	3.50	2.93	
										2.37	
										2.56	
										2.32	
										2.53	
										<b>2.30</b>	
										<b>2.51</b>	

IRM native bruitée : A.2-(l)					IRM native bruitée : A.2-(l)				
dim	32	64	128	256	32	64	128	256	
	(bpp)	(bppa)	(bppa)	(bppa)	(bpp)	(bppa)	(bppa)	(bppa)	
J2K	<u>9.84</u>	6.72	<u>6.28</u>	<u>5.41</u>	5.70	-	-	-	
IHINT	8.69	<u>6.83</u>	6.19	5.20	5.52	<u>5.67</u>	<u>3.78</u>	<u>3.18</u>	
HOP	<b>8.27</b>	<b>6.57</b>	<b>5.87</b>	5.12	5.37	<b>5.22</b>	<b>3.48</b>	<b>2.87</b>	
HOP-LSE	8.28	6.59	5.89	5.04	5.32	5.23	3.52	2.89	
HOP-LSE <sup>+</sup>	<b>8.27</b>	6.60	5.88	<b>5.03</b>	<b>5.31</b>	5.23	3.54	<b>2.87</b>	
								2.18	
								<u>2.31</u>	
								<u>2.59</u>	
								2.42	
								<b>2.36</b>	
								<b>2.36</b>	

Les images décompressées ont une taille  $\text{dim} \times \text{dim}$  avec 12 bits par pixel. La notation bppa signifie bits par pixel ajouté, et correspond au débit utilisé pour coder les  $\text{dim} \times \text{dim} - \text{dim}/2 \times \text{dim}/2$  pixels nécessaires à obtenir cette résolution à partir de la précédente. Les meilleurs résultats sont en gras et les moins bons soulignés.

TAB. 4.3 – Résultats pour la scalabilité en résolution



$s$  est décomposé en

$$m = s \bmod 2^k, \quad (4.16)$$

$$d = \lfloor s/2^k \rfloor, \quad (4.17)$$

avec  $k$  le nombre de bits nécessaires à représenter  $Q_\delta(T_{\text{Noise}})$ . Cette décomposition, qui est similaire à celle pouvant être utilisée pour un codage Golomb-Rice, est compressée à l'aide de  $k + 1$  codeurs arithmétiques adaptatifs : un codeur pour chaque bit de  $m$  (qui contiennent principalement du bruit), et un codeur pour la représentation unaire de  $d$ . Avec IHINT,  $T_{\text{Noise}}$  est fixé à 0 : sa prédiction n'étant pas adaptative, ce seuil n'est pas utilisé lors de la décorrélation, et de meilleurs taux de compression ont été obtenus ainsi.

Les  $k$  codeurs arithmétiques binaires, utilisés pour  $m$ , maintiennent indépendamment le nombre d'occurrences des symboles 0 ( $F0$ ) et 1 ( $F1$ ). Lorsque  $F0 + F1$  dépasse un seuil  $TB$ ,  $F0$  et  $F1$  sont divisés par 2, ce qui permet une meilleure adaptation aux statistiques locales. Le codeur binaire utilisé avec la représentation unaire de  $d$  compresse une séquence de  $d$  symboles 1, suivie par un symbole 0, tout en maintenant constamment leurs fréquences d'apparitions. Suite à des expérimentations, le codage de  $d$  est moins dépendant des statistiques locales, et un seuil  $TU$  a seulement été mis en place de sorte à assurer que la capacité de représentation entière du nombre d'occurrences ne soit pas dépassée. Quelques détails sur ce codage arithmétique « unaire » peuvent être trouvés dans un rapport technique de Krivoulets [Kri02], dans lequel il a été conçu pour des sources suivant une distribution Laplacienne.

Les statistiques d'un alphabet de taille réduite adaptative, incluant un symbole d'échappement, sont apprises durant la phase 0 de codage, précédemment décrite. En commençant avec l'alphabet  $\mathcal{A}$  qui contient uniquement le symbole d'échappement (ESC) avec une fréquence  $F_{\mathcal{A}}(\text{ESC}) = 1$ , il est conçu comme suit :

- Lorsqu'un symbole  $s \notin \mathcal{A}$  est rencontré,  $F_{\mathcal{A}}(\text{ESC})$  est incrémenté de un,  $\mathcal{A}$  devient  $\mathcal{A} \cup \{s\}$  et  $F_{\mathcal{A}}(s) = 1$ .
- Lorsqu'un symbole  $s \in \mathcal{A}$  est rencontré,  $F_{\mathcal{A}}(s)$  est incrémenté de un.
- Lorsque  $SF_{\mathcal{A}} = \sum_{s \in \mathcal{A} \setminus \{\text{ESC}\}} F_{\mathcal{A}}(s)$  est supérieur à un seuil  $T SF_{\mathcal{A}}$ , les fréquences sont réinitialisées : la fréquence de chaque symbole  $s \in \mathcal{A} \setminus \{\text{ESC}\}$  est mise à  $F_{\mathcal{A}}(s)/2$ , et celle de  $F_{\mathcal{A}}(\text{ESC})$  est mise à  $\max(1, F_{\mathcal{A}}(\text{ESC})/2)$ . Tous les symboles avec une fréquence nulle sont supprimés de  $\mathcal{A}$ .

Le passage à la seconde phase de codage se produit lorsque les symboles les plus fréquents ont été pris en compte (en s'appuyant sur la probabilité du symbole d'échappement). Ceci est estimé simplement en comparant  $F_{\mathcal{A}}(\text{ESC})$  à un seuil  $SF_{\mathcal{A}}/TS$ . Si  $F_{\mathcal{A}}(\text{ESC})$  est plus petit, il est déduit que la plupart du temps les symboles rencontrés sont présents dans  $\mathcal{A}$ . Alors, puisque les symboles fréquents sont dans  $\mathcal{A}$ , ESC devrait être moins utilisé et  $F_{\mathcal{A}}(\text{ESC})$  très probablement décroître après chaque nouvelle réinitialisation des fréquences. Plutôt que d'attendre que cela se produise, et parce que

	codeur CP0	codeur CP1	combinaison
MeDEISA CT	5.033	5.051	5.019
VHP-Male CT	4.843	4.864	4.826
VHP-Male MRI	4.832	4.892	4.804
CIPR (8bits) CT	2.097	2.076	2.071
CIPR (8bits) MRI	2.736	2.748	2.715

TAB. 4.4 – Résultats pour la compression des résidus de HOP (bpp) en utilisant uniquement le codeur CP0, uniquement le codeur CP1, et la combinaison proposée

$F_{\mathcal{A}}(\text{ESC})$  introduirait pendant ce temps un cout de codage plus important des autres symboles de  $\mathcal{A}$ ,  $F_{\mathcal{A}}(\text{ESC})$  est empiriquement réduit à  $\max(1, F_{\mathcal{A}}(\text{ESC})/TS)$ .

Une fois que le codage de la seconde phase (CP1) est démarré, les symboles suivants sont compressés à l'aide d'un codage arithmétique guidé par l'alphabet  $\mathcal{A}$ , et les mêmes règles que celles utilisées durant sa conception sont utilisées afin de continuer à adapter les statistiques. Lorsqu'un symbole d'échappement est rencontré, CP0 est conservé pour coder le symbole échappé  $e$  par la valeur  $e\text{-Card}(\{s \in \mathcal{A}/s < e\})$ . Parce que cette petite astuce de réorganisation des valeurs échappées (qui réduit leur cout de codage) implique principalement des modifications dans les statistiques du codeur binaire utilisé pour la représentation unaire de  $m$ , lorsque le changement de phase de codage intervient, les fréquences d'apparition des symboles 0 et 1 sont réinitialisées à 1.

La TAB. 4.4 donne quelques résultats obtenus avec la méthode de codage proposée, et permet de les comparer avec ceux obtenus lorsque seul le codeur de la phase 0 ou seul le codeur de la phase 1 sont utilisés pour la compression. Ils montrent que CP0 seul peut atteindre des résultats intéressants. Puisque CP1 seul est moins efficace mais que la combinaison introduite permet d'obtenir les meilleurs résultats, il peut en être déduit que l'alphabet dynamique permet une bonne approximation de la PDF une fois que la table des fréquences est suffisamment alimentée, et que l'approche en deux phases est pertinente, notamment pour la scalabilité en résolution précédemment présentée : CP0 est plus efficace pour les images de petites dimensions/résolutions, tandis que CP1 renforce la compression des bandes de résidus plus grandes.

Pour toutes les expérimentations de ce chapitre, les seuils/constantes ont été fixés empiriquement à  $TSF_{\mathcal{A}} = 32(Q_{\delta}(I_{\max}) - Q_{\delta}(I_{\min}))$ ,  $TS = 16$ , et  $TB = 1024$ .

		Compression sans perte : débits moyens (bpp)							
		non scalable		scalable					
		Bases	CALIC	JLS	SPIHT	J2K	IHINT	HOP	HOP-LSE
CT	CIPR (8bits)	<b>1.92</b>	1.96	<u>2.19</u>	2.17	2.14	2.07	1.96	1.94
	MeDEISA	4.72	4.87	4.74	4.84	4.83	<u>5.02</u>	4.68	<b>4.51</b>
	PMR	3.69	<u>3.99</u>	3.71	3.88	3.91	3.76	3.39	<b>3.27</b>
	VHP-Female	<b>4.65</b>	4.73	4.87	<u>4.91</u>	4.89	4.73	4.71	4.71
	VHP-Male	<b>4.75</b>	4.82	4.97	<u>5.01</u>	5.00	4.83	4.82	4.84
	VHP-Harvard	<b>5.03</b>	5.05	5.28	5.35	<u>5.37</u>	5.18	5.22	5.18
	totalité	<b>4.60</b>	4.70	4.78	<u>4.84</u>	4.83	4.72	4.63	<b>4.60</b>
MRI	CIPR (8bits)	2.69	2.79	2.82	<u>2.96</u>	2.79	2.71	2.63	<b>2.59</b>
	MeDEISA	3.15	3.29	3.24	<u>3.33</u>	3.18	3.11	3.06	<b>3.05</b>
	PMR	3.00	<u>3.23</u>	3.15	3.15	3.03	3.09	2.88	<b>2.78</b>
	VHP-Female	4.41	<u>4.62</u>	4.50	4.61	4.49	4.43	4.36	<b>4.33</b>
	VHP-Male	4.90	<u>5.08</u>	4.91	5.01	4.87	4.81	4.77	<b>4.73</b>
	VHP-Harvard	<b>4.68</b>	4.85	4.96	<u>5.05</u>	4.86	<b>4.68</b>	<b>4.68</b>	<b>4.68</b>
	VHP-Harvard-3D	3.70	<u>4.13</u>	3.58	3.82	3.64	3.81	3.38	<b>3.14</b>
totalité	3.84	<u>4.07</u>	3.90	4.01	3.86	3.86	3.71	<b>3.63</b>	
totalité		4.47	4.59	4.63	<u>4.70</u>	4.67	4.58	4.48	<b>4.44</b>

Les meilleurs résultats sont en gras et les moins bons soulignés. Les cellules des meilleurs résultats par catégorie (scalable/non-scalable) sont grisées.

TAB. 4.5 – Débits moyens pour la compression sans perte

		Compression presque sans perte, PAE=4						Lossy PSNR
		Débits / PSNR (bpp / dB)						
Bases		JLS	IHINT	HOP	HOP-LSE	HOP-LSE <sup>+</sup>	J2K-9/7	
CT	CIPR (8bits)	0.72 / 42.70	0.65 / 43.23	0.59 / 42.70	<b>0.57 / 42.82</b>	0.57 / 42.80	<u>44.94</u>	
	MeDEISA	2.46 / 64.73	2.35 / 64.92	2.49 / 64.85	2.22 / 64.87	<b>2.12 / 64.87</b>	<u>64.91</u>	
	PMR	1.61 / 64.42	1.32 / 64.88	1.21 / 64.92	1.01 / 65.04	<b>1.00 / 65.08</b>	<u>65.92</u>	
	VHP-Female	2.33 / 64.76	2.42 / 64.97	2.26 / 64.87	2.24 / 64.89	<b>2.22 / 64.90</b>	63.92	
	VHP-Male	2.46 / 64.90	2.58 / 65.11	2.41 / 65.02	2.39 / 65.03	<b>2.37 / 65.03</b>	63.86	
	VHP-Harvard	2.63 / 64.59	2.76 / 64.90	<b>2.60 / 64.83</b>	2.69 / 64.85	2.66 / 64.85	63.27	
	totalité	2.34 / 64.28	2.38 / 64.52	2.27 / 64.43	2.20 / 64.45	<b>2.17 / 64.46</b>	63.73	
MRI	CIPR (8bits)	0.91 / 41.35	0.76 / 42.07	0.70 / 41.81	<b>0.67 / 41.89</b>	0.68 / 41.90	<u>43.72</u>	
	MeDEISA	1.02 / 65.69	1.03 / 65.96	0.92 / 65.66	<b>0.88 / 65.71</b>	0.88 / 65.60	<u>66.83</u>	
	PMR	1.30 / 65.16	1.10 / 65.73	1.12 / 65.19	1.00 / 65.27	<b>0.97 / 65.43</b>	<u>67.00</u>	
	VHP-Female	1.75 / 64.58	1.70 / 64.61	1.56 / 64.79	1.51 / 64.79	<b>1.50 / 64.77</b>	<u>65.03</u>	
	VHP-Male	2.14 / 64.37	2.00 / 64.39	1.89 / 64.48	1.85 / 64.48	<b>1.83 / 64.48</b>	64.09	
	VHP-Harvard	2.44 / 64.68	2.43 / 64.92	2.27 / 64.61	2.27 / 64.61	<b>2.26 / 64.65</b>	63.65	
	VHP-Harvard-3D	1.53 / 64.40	1.21 / 65.01	1.24 / 64.81	0.99 / 64.90	<b>0.98 / 64.94</b>	<u>65.58</u>	
totalité	1.56 / 63.67	1.43 / 63.97	1.36 / 63.82	1.27 / 63.86	<b>1.26 / 63.87</b>	<u>64.48</u>		
totalité		2.21 / 64.17	2.22 / 64.42	2.12 / 64.32	2.05 / 64.35	<b>2.02 / 64.36</b>	63.85	

Pour chacune des base d'images, le débit moyen du meilleur des codeurs (MdC) presque sans perte et le PSNR moyen associé sont en gras. Le PSNR moyen obtenu par le MdC est comparé avec celui de JPEG-2000 en mode de compression lossy (J2K-9/7) contraint pour chacune des images par le débit obtenu par le MdC. Le meilleur PSNR moyen entre le MdC et J2K est souligné.

TAB. 4.6 – Débits et PSNR moyens pour la compression presque sans perte avec un PAE de 4

## 4.3 Résultats

### 4.3.1 Compression sans perte

HOP, HOP-LSE et HOP-LSE<sup>+</sup> sont comparés avec l'implémentation de IHINT décrite dans ce chapitre, avec les standards JPEG-LS<sup>1</sup> (JLS) et JPEG2000<sup>2</sup> (J2K) et avec les logiciels de référence de SPIHT<sup>3</sup> (scalable en qualité) et CALIC<sup>4</sup>. La TAB. 4.5 rapporte les débits moyens obtenus lors de la compression sans perte de bases d'images introduites dans 1.1.3.1.

Sur les CT, CALIC donne toujours les meilleures performances de compression, à l'exception des images douces (MeDEISA et PMR) pour lesquelles l'optimisation dynamique des prédicteurs est plus efficace. Sur les IRM, à l'exception du sous-ensemble d'images douces (VHP-Harvard-3D et PMR), ses résultats de compression sont équivalents à HOP.

Pour les codeurs scalables uniquement, mises à part les CT de MeDEISA pour lesquelles HOP n'est pas efficace, J2K offre majoritairement les moins bons résultats de compression. Cependant, en dehors des images douces, HOP est toujours meilleur que SPIHT, J2K et IHINT. Les résultats de IHINT sont similaires à J2K sur les CT mais compétitifs avec CALIC et HOP sur les IRM.

L'optimisation proposée pour les prédicteurs permet de contourner l'inefficacité de HOP sur les images douces, et HOP-LSE et HOP-LSE<sup>+</sup> fournissent les meilleurs résultats de compression sur celles-ci. Ils permettent la plupart du temps d'être plus performants que HOP avec ses prédicteurs statiques.

Bien que sur les CT de MeDEISA HOP soit moins efficace que J2K ou IHINT de 3%, avec HOP-LSE et HOP-LSE<sup>+</sup> il permet d'économiser 3.3% et 6.8% de l'espace de stockage requis par ceux-ci. De plus, HOP, HOP-LSE et HOP-LSE<sup>+</sup> améliorent la compression de plus de 3% sur les autres CT. Et, sur les CT 8 bits, HOP-LSE<sup>+</sup> réduit le débit par plus de 10% en comparaison à J2K.

Sur les IRM sans particularité, l'optimisation dynamique des prédicteurs peut améliorer la compression de HOP jusqu'à environ 2%, ce qui produit un gain de codage allant de 5% à plus de 7% comparativement à J2K, et environ 3% par rapport à IHINT. Sur les IRM 8 bits, HOP, HOP-LSE et HOP-LSE<sup>+</sup> améliorent encore la compression de 8.4%, 11.1% et 12.5% par rapport à J2K, et 2.9%, 5.7% et 7.2% par rapport à IHINT. Enfin, sur les IRM les plus douces (VHP-Harvard-3D) HOP-LSE<sup>+</sup> économise 17.8% de l'espace requis par J2K et 13.7% de celui nécessité par IHINT.

---

<sup>1</sup><http://www.hpl.hp.com/loco/>

<sup>2</sup><http://www.kakadusoftware.com/>

<sup>3</sup><http://www.cipr.rpi.edu/research/SPIHT/spiht3.html> (version linux)

<sup>4</sup>[ftp://ftp.csd.uwo.ca/pub/from\\_wu/v.arith](ftp://ftp.csd.uwo.ca/pub/from_wu/v.arith) (version sun)

**Synthèse des résultats** En moyennant sur l'ensemble des images de toutes des bases, HOP, HOP-LSE et HOP-LSE<sup>+</sup> sont 2.48%, 4.34% et 4.96% plus efficaces que J2K sur les CT ; et 3.74%, 7.48% et 9.48% sur les IRM.

### 4.3.2 Compression presque sans perte

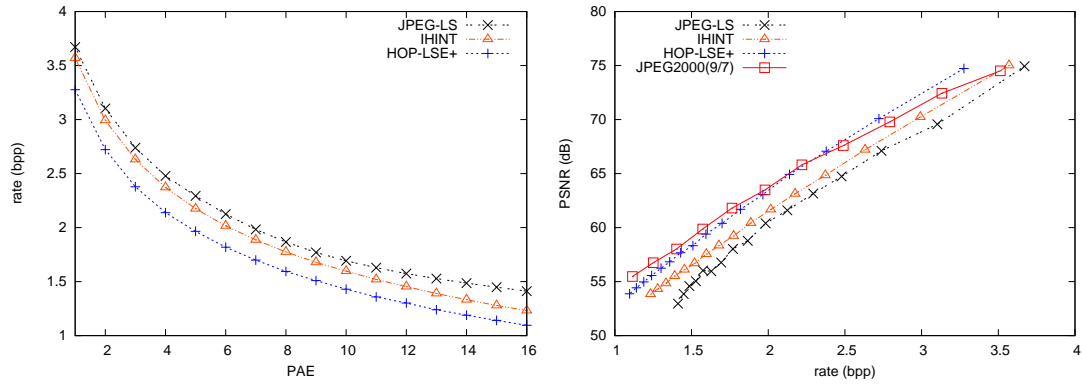
Pour les comparaisons presque sans perte, seuls IHINT et JPEG-LS sont conservés car les autres logiciels précédemment utilisés ne proposent pas une compression contrainte par le PAE. Cependant, les performances en terme de PSNR des algorithmes contraints par le PAE sont comparées avec le mode avec pertes de JPEG-2000 (contraint en débit), utilisant des ondelettes bi-orthogonales à valeurs réelles (noyau 9/7). Les débits et PSNR moyens obtenus par JPEG-LS, IHINT, HOP, HOP-LSE et HOP-LSE<sup>+</sup> pour un faible PAE de 4 (afin de ne pas perturber le diagnostic) sont donnés dans la TAB. 4.6. Les résultats en PSNR de JPEG-2000 sont obtenus à l'aide du logiciel Kakadu v6.0, utilisant un pas de quantification (Qstep) de  $1/2^d$ , avec  $d$  étant la profondeur en bits de l'image originale (8 ou 12).

Les mêmes images que celles utilisées pour 4.2.3 sont utilisées dans la FIG. 4.7 afin de montrer le type de courbes débit/distorsion pouvant être obtenues avec JPEG-LS, IHINT et la meilleure des prédictions hiérarchiques orientées : HOP-LSE<sup>+</sup> (pour une meilleure lisibilité, les résultats de HOP et HOP-LSE ne sont pas tracés). Sur les CT de MeDEISA, qui illustre les résultats sur les images douces, et qui est l'une des images les moins bien compressée par HOP (en comparaison aux autres algorithmes), HOP serait aux alentours de 0.21 bpp moins efficace que IHINT quand le PAE vaut 1 (le PSNR est inférieur d'environ 1.92 dB) et 0.06 bpp moins efficace lorsque le PAE vaut 16 (le PSNR est inférieur d'environ 0.96 dB). Les résultats de HOP-LSE seraient entre IHINT et HOP-LSE<sup>+</sup>. Sur les deux autres images, qui fournissent des exemples pour des CT et IRM natives bruitées, les résultats de HOP et HOP-LSE seraient similaires à HOP-LSE<sup>+</sup>.

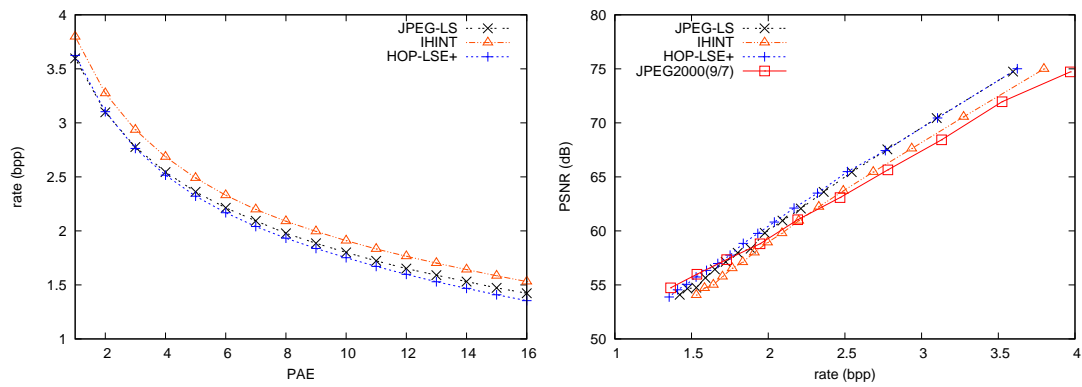
**Synthèse des résultats** Un PAE de 4 permet une amélioration de la compression par un facteur supérieur à 2 en comparaison à une compression en mode sans perte et n'a pas d'impact visuel. Les résultats de HOP, HOP-LSE et HOP-LSE<sup>+</sup> montrent qu'en terme de PSNR ils peuvent atteindre des performances équivalentes voire meilleures que JPEG-2000 sur les images natives bruitées. Leurs PSNR sont légèrement moins bons sur les images 8 bits et sur les images douces. Cependant l'approche presque sans perte bénéficie du contrôle de qualité en PAE, contrairement à JPEG-2000.

Comme en mode sans perte, les trois approches HOP obtiennent des résultats relativement équivalents sur les images les plus bruitées. L'approche orientée de HOP est, encore une fois, meilleure que celle non-orientée de IHINT, excepté sur les sous-ensembles d'images les plus douces pour lesquelles HOP-LSE et HOP-LSE<sup>+</sup> font la différence avec l'optimisation de leurs prédicteurs qui permet d'améliorer HOP par plus de 10% et jusqu'à 20% sur le sous-ensemble VHP-Harvard-3D. En comparaison au standard JPEG-LS, HOP améliore à la fois le débit et le PSNR et propose la scalabilité en résolution.

Résultats sur une image douce (CT) : A.1-(e)



Résultats sur une CT native bruitée : A.1-(k)



Résultats sur une IRM native bruitée : A.2-(l)

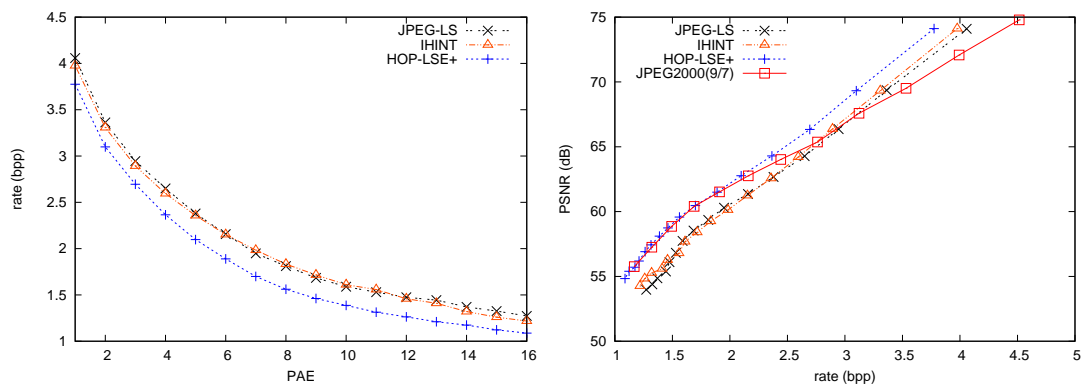


FIG. 4.7 – Résultats débit/distorsion pour la compression presque sans perte.

## Conclusion et perspectives

Une nouvelle approche hiérarchique prédictive (HOP) et deux variantes s'appuyant sur l'optimisation au sens des moindres carrés de prédicteurs (HOP-LSE et HOP-LSE<sup>+</sup>) ont été présentés dans ce chapitre, dans le cadre de la compression sans perte et presque sans perte, scalable en résolution, pour les images médicales. Une nouvelle méthode contextuelle d'annulation des biais, basée sur une correction séquentielle, a été proposée afin d'améliorer l'efficacité de la prédiction. Et une approche originale pour la compression, s'appuyant sur un codeur entropique en deux phases, a été conçue afin d'améliorer la compression dans le contexte de la scalabilité en résolution.

Bien qu'une représentation scalable en résolution ait été utilisée, les résultats de ce chapitre montrent qu'il est possible d'obtenir quelques améliorations en comparaison aux algorithmes de référence pour la compression d'images médicales bruitées, que ce soit en sans perte ou en presque sans perte. Pour un support de prédiction plus étendu, l'optimisation au sens des moindres carrés a également permis de pousser plus loin la compression d'images douces, pour lesquelles HOP n'était pas des plus efficaces.

Pour de faibles distorsions, l'algorithme obtient un PSNR équivalent voire meilleur que JPEG-2000 sur les images bruitées et bénéficie en plus du contrôle de la qualité induit par l'utilisation du PAE, ce qui n'est pas le cas de JPEG-2000. La compression presque sans perte utilisant un PAE de 4, qui n'a visuellement pas d'impact, permet de diviser par 2 la taille requise pour le stockage et la transmission des images en comparaison à une compression sans perte.

Les différentes étapes de l'algorithme peuvent certainement être améliorées davantage, dans le but d'accélérer la vitesse d'exécution, par exemple en effectuant l'optimisation des prédicteurs moins régulièrement qu'à chaque pixel pour HOP-LSE et HOP-LSE<sup>+</sup>, et/ou d'augmenter l'efficacité de la compression. L'utilisation de prédicteurs statiques un peu plus robustes ou d'une grille quinconce pour le sous-échantillonnage pourrait être intéressante en particulier pour modifier la prédiction verticale ( $o = V$ ) qui est la principale source de pénalisation de HOP (en comparaison à IHINT par exemple) sur les images douces, et pourrait permettre une légère amélioration des résultats sans nécessiter l'optimisation.

## Chapitre 5

# Décomposition adaptative en paquets d'ondelettes

### Introduction

Ce chapitre s'intéresse davantage à la compression sans perte permettant une transmission scalable en qualité. Dans les travaux récents, les ondelettes sont principalement employées pour répondre à ce besoin. Afin de produire un flux de données intégrant la scalabilité, les coefficients d'ondelettes sont successivement raffinés à l'aide de quantificateurs imbriqués. Pour être simple et pratique, les raffinements successifs sont effectués à l'aide d'un codage par plan de bits. Ce schéma de compression permet de proposer un bon compromis débit/distorsion lorsque le flux est convenablement organisé. Les algorithmes populaires EZW [Sha93], LZC [TZ94], SPIHT [SP96b] et EBCOT [Tau00], qui constitue la ligne directrice du standard JPEG-2000 [TM01], s'appuient sur ce modèle de compression.

Même si de tels codeurs sont généralement moins efficaces pour la compression sans perte que les approches prédictives, la scalabilité en qualité (difficilement offerte par les modèles prédictifs) est mieux adaptée pour la télé-transmission interactive. Les codeurs intra-bandes (comme LZC et EBCOT) offrent plus de flexibilité que les approches inter-bandes (comme EZW et SPIHT) et sont donc souvent préférés. Ils peuvent, par exemple, facilement faire bénéficier de la scalabilité en résolution (offerte par la transformée en ondelettes) en changeant simplement l'organisation de l'information dans le flux de données. EBCOT, qui découpe le codage de chaque sous-bande en macro-blocs (*code-blocks*) permet également, lorsqu'utilisé avec un protocole de communication tel que JPIP (JPEG-2000 part 9 : ISO/IEC International Standard 15444-9 – ITU-T Recommendation T.808), de ne transmettre que l'information nécessaire à la construction d'une certaine localisation spatiale, à une certaine résolution. Cette fonctionnalité est très utile pour de la consultation interactive à distance de très grandes images. Dans le principe d'utilisation on peut, par exemple, penser à google-maps.

Pour la compression sans perte, une décomposition dyadique en ondelettes entières



5/3 est traditionnellement utilisée. Cependant, selon les données à compresser, cette transformée n'est pas toujours très efficace. Dans ce chapitre, un modèle de représentation plus générique, en partie inspirée par [OP09], sous forme d'arbre de décomposition en paquets d'ondelettes [CMW92, Mal08] est exploité pour améliorer les taux de compression sur les images les plus corrélées. Il permet une optimisation pratique de la décomposition pour des données  $N$ -dimensionnelles (ici des images 2D et volumiques) afin de mieux l'adapter à la régularité (qui peut changer en fonction des modalités d'images, du matériel d'acquisition et d'éventuels post-traitements) et au contenu de l'image.

La modélisation de la base de décomposition est présentée dans la section 5.1, ainsi que les filtres utilisés pour générer les ondelettes à support paramétrable, et une approche pour l'optimisation de ce modèle est donnée dans la section suivante (sec. 5.2). Suit une description de la méthode utilisée pour le codage du modèle optimisé et pour la compression scalable des coefficients d'ondelettes (sec. 5.3). Puis les résultats obtenus sont commentés dans la section 5.4. Enfin, la dernière section (sec. 5.5) détaille une extension pour la compression presque sans perte, explique comment la décomposition est étendue pour la compression d'une région d'intérêt, et commente un modèle qui avait été initialement étudié, s'appuyant sur des ondelettes orientées, mais qui s'étant révélé moins efficace en compression sans perte n'a pas été approfondi.

Les principales contributions de ce chapitre concernent le modèle utilisé pour la décomposition, son optimisation pour la compression sans perte, et son extension presque sans perte.

## 5.1 Base de décomposition

### 5.1.1 Modèle de représentation de la structure de décomposition

#### 5.1.1.1 Représentation par arbre binaire

En se plaçant dans un espace à  $N$  dimensions (3 pour les images volumiques), une base de décomposition adaptative en paquets d'ondelettes est modélisée par un arbre binaire  $\Theta$ .

Chaque nœud  $\Theta_n$  est indexé par une chaîne  $n$  de symboles  $\{\mathbf{L}, \mathbf{H}\}$ , qui décrit le chemin d'accès à  $\Theta_n$  dans l'arbre (à partir de la racine). Chaque nœud est associé à une sous-bande  $\beta_n$  qui résulte de la décomposition de la bande associée à son nœud parent.  $\mathbf{L}$  symbolise alors les basses fréquences et est associé au fils gauche dans l'arbre tandis que  $\mathbf{H}$  symbolise les hautes fréquences et est associé au fils droit. Avec  $\emptyset$  la chaîne vide, le nœud racine  $\Theta_\emptyset$  est couplé à l'ensemble de données initial (l'image multidimensionnelle)  $\beta_\emptyset$ .

Chaque nœud se voit assigner une valeur  $\Delta_n \in \{0, 1, \dots, N\}$  qui correspond à l'indice de la dimension selon laquelle la sous-bande associée est décomposée. Si  $\Delta_n = 0$ , elle n'est pas décomposée et le nœud est une feuille de l'arbre. Les deux enfants d'un nœud

interne (décomposé)  $\Theta_n$  sont  $\Theta_{n:L}$  pour la sous-bande basses fréquences, et  $\Theta_{n:H}$  pour celle hautes fréquences, avec “ $\cdot : \cdot$ ” l’opérateur de concaténation de chaînes.

### 5.1.1.2 Paramètres d’adaptation supplémentaires

Pour permettre d’aller au delà d’une décomposition en paquets d’ondelettes avec des atomes fixés, et afin de rendre la décomposition plus adaptative, un modèle plus souple peut être utilisé. Des données supplémentaires  $\Omega_n$ , spécifiant les informations/paramètres supplémentaires requis par la décomposition adaptative, sont alors assignés à chaque nœud interne.

Pour cette étude,  $\Omega_n$  contient simplement un paramètre  $a$  qui sert à générer un filtre interpolatif. Ce dernier est utilisé dans un schéma de lifting (5.1.2) afin de décomposer toute la sous-bande. Pour une approche plus complexe, comme par exemple l’ADL (*Adaptive Directional Lifting*) [DWW<sup>+</sup>07, WSJN07, LN08],  $\Omega_n$  pourrait être la carte d’orientation des filtres ainsi que leurs paramètres.

## 5.1.2 Familles d’ondelettes utilisées

Afin d’étendre l’approche proposée dans [TL10a] concernant l’utilisation des filtres de Lanczos pour la construction des étapes de lifting, les filtres interpolatifs de Deslauriers et Dubuc [DD89], qui eux sont optimaux en terme de nombre de moments nuls, vont également être considérés dans ce chapitre.

### 5.1.2.1 Filtres interpolatifs de Lanczos normalisés

Les filtres de Lanczos sont bien connus en traitement d’images pour la qualité de leur interpolation et leur facilité de paramétrage. Ils n’ont pas vraiment de propriétés mathématiques remarquables mais possèdent une bonne réponse fréquentielle. Avec la définition classique des filtres de Lanczos [Duc79] :

$$L_a(t) = \begin{cases} \frac{a \sin(\pi t) \sin(\pi t/a)}{\pi^2 t^2} & \text{si } -a < t < a \text{ et } t \neq 0, \\ 1 & \text{si } t = 0, \\ 0 & \text{sinon,} \end{cases} \quad (5.1)$$

où  $a \in \mathbb{N}^*$  est un paramètre permettant de fixer la taille du support, des filtres interpolatifs de Lanczos, discrets et normalisés, sont définis sous une forme intéressante pour le lifting par :

$$\tilde{L}_{a,F,P}(k) = \frac{L_a\left(k + \frac{P}{F}\right)}{\sum_{z \in \mathbb{Z}} L_a\left(z + \frac{P}{F}\right)}, \quad k \in \mathbb{Z}, \quad (5.2)$$

où  $F \in \mathbb{N}^*$  est un facteur d'expansion entier, et  $P \in \mathbb{Z}/F$  est la position à laquelle l'interpolation est effectuée. Un signal discret  $s$  peut alors être raffiné d'un facteur  $F$  :

$$s_F(t = k \cdot F + P) = \left( s * \tilde{L}_{a,F,P} \right) (k), \quad k \in \mathbb{Z}, \quad (5.3)$$

et vérifie la propriété :

$$s_F(k \cdot F) = s(k), \quad \forall k \in \mathbb{Z}, \quad (5.4)$$

$\tilde{L}_{a,F,0}$  étant la fonction discrète  $\delta$  de Dirac.

### 5.1.2.2 Filtres interpolatifs de Deslauriers et Dubuc

Les fonctions d'échelles Deslauriers-Dubuc [DD89] permettent un raffinement par interpolation d'un facteur 2. Leur principale propriété est d'être des filtres interpolatifs optimaux (au sens polynomial) pour une taille de support donné, en ayant  $2a$  moments nuls pour un support de taille  $4a - 1$ . Elles peuvent être définies par [Ber99]-Annexe B :

$$D_a^2(k) = \begin{cases} 1 & \text{si } k = 0, \\ 0 & \text{sinon si } k \in 2\mathbb{Z}, \text{ ou si } |k| \geq 2a, \\ \frac{2(-1)^{i+1} (2a)!^2}{2^{4a} a!^2 (a-i)!(a+i-1)!(2i-1)} & \text{sinon avec } k = \pm(2i-1). \end{cases} \quad (5.5)$$

Ainsi, pour un facteur d'expansion  $F = 2$ , les filtres interpolatifs Deslauriers-Dubuc peuvent être reformulés de la même façon que pour les filtres Lanczos par :

$$\tilde{D}_{a,2,P}^2(k) = D_a^2(2k + P). \quad (5.6)$$

### 5.1.2.3 Ondelettes construites par lifting

Pour la suite seul  $F = 2$  et  $P = 1$  sont nécessaires.  $\tilde{L}_{a,2,1}$  et  $\tilde{D}_{a,2,1}^2$  seront alors notés  $\tilde{L}_a$  et  $\tilde{D}_a^2$ . Enfin, seule une famille de filtres sera utilisée pour la décomposition d'une image. On généralise donc l'approche avec  $\tilde{F}_a$  qui fournit des filtres prédictifs dont la longueur du support est  $2a$ . Ceux-ci vont être utilisés dans un schéma de lifting [Swe96b] composé de seulement deux étapes : prédiction  $P_a$  et mise à jour  $U_a$ . Après le découpage d'un signal  $s$  (en dimension 1) en deux versions sous-échantillonnées  $e[s](k) = s(2k)$  (échantillons pairs) et  $o[s](k) = s(2k+1)$  (échantillons impairs), les deux étapes peuvent être définies par :

$$\begin{aligned} P_a[s](k) &= o[s](k) - \left[ \left( e[s] * \tilde{F}_a \right) (k) \right], \\ U_a[s](k) &= e[s](k) + \left[ \frac{1}{2} \left( P_a[s] * \tilde{F}_a \right) (k-1) \right]. \end{aligned} \quad (5.7)$$

En N-dimension, pour une décomposition de la dimension  $d$ , les échantillons pairs sont notés  $e_d$ , les impairs  $o_d$ , la prédiction  $P_{a,d}$  et la mise à jour  $U_{a,d}$ .

Il peut être noté que pour  $a = 1$ , (5.7) est équivalent, aussi bien pour  $\tilde{L}_a$  que pour  $\tilde{D}_a^2$ , au schéma de lifting 5/3 traditionnellement utilisé en compression sans perte, et pour  $a = 2$ ,  $\tilde{L}_a$  est également équivalent à  $\tilde{D}_a^2$  qui correspond au schéma de lifting introduit dans [Swe96b] ( $N = 4$  et  $\tilde{N} = 4$ ) pour les filtres Deslauriers-Dubuc.

## 5.2 Optimisation de la base pour la compression sans perte

Pour adapter la base de décomposition à la nature de l'image, aussi bien l'arbre de décomposition que le paramètre  $a$  des filtres utilisés pour chaque nœud de décomposition sont optimisés. Afin d'éviter de devoir résoudre un problème NP complet, l'optimisation s'effectue par une construction incrémentale de l'arbre à l'aide d'heuristiques locales sous-optimales (sans remise en cause des paramètres précédemment optimisés).

Pour optimiser les paramètres d'un nœud  $\Theta_n$ , la fonction suivante est minimisée :

$$L_{\Theta}(n, \omega_n, \delta_n) = \rho_{\Theta}(n, \omega_n, \delta_n) + \mathcal{W}_{\Theta}(n, \omega_n, \delta_n), \quad (5.8)$$

avec  $\delta_n$  la dimension de la décomposition,  $\omega_n$  les paramètres de décomposition,  $\rho_{\Theta}$  une fonction de pénalisation et  $\mathcal{W}_{\Theta}$  une fonction de coût.

Pour la compression sans perte, en considérant un modèle de distribution géométrique symétrique centré en 0 [MSW00] (TSGD) pour la distribution des coefficients d'ondelettes, la norme  $L_1$  de la sous-bande prédite est utilisée comme critère de minimisation (normalisée par le nombre de coefficients de la sous-bande : i.e moyenne des valeurs absolues). Même si ce critère ne permet de minimiser que l'entropie d'ordre zéro d'une telle distribution, il permet tout de même de bien réduire l'entropie pour un codage contextuel. Ainsi la fonction de coût suivante est utilisée :

$$\mathcal{W}_{\Theta}(n, \omega_n, \delta_n) = \frac{1}{\text{Card}(H)} \sum |H|, \quad (5.9)$$

$H = P_{\omega_n, \delta_n}[\beta_n]$  étant la bande hautes fréquences résultante de la décomposition (résidus de la prédiction par interpolation).

### 5.2.1 Sélection du filtre pour une étape de décomposition

Le paramètre  $a$  du filtre à utiliser pour la décomposition d'un nœud  $\Theta_n$  pour la dimension  $d$  est optimisé en minimisant  $L_{\Theta}(n, a, d)$ , pour  $n$  et  $d$  fixés.

Pour cette étude,  $\rho_{\Theta} = 0$ , et l'optimisation est effectuée itérativement, en commençant par  $a = 1$  et en l'incrémentant tant que  $L_{\Theta}(n, a+1, d) < L_{\Theta}(n, a, d)$ . Cette méthode est assez rapide et souvent optimale avec les familles de filtres choisies. Le paramètre optimal ainsi trouvé est noté  $\hat{a}_{n,d}$ . Dans 5.2.2, il est calculé pour chaque dimension potentiellement décomposable, afin d'optimiser l'arbre. Un exemple de fonctions  $L_{\Theta}$  à minimiser est donné dans la FIG. 5.1.

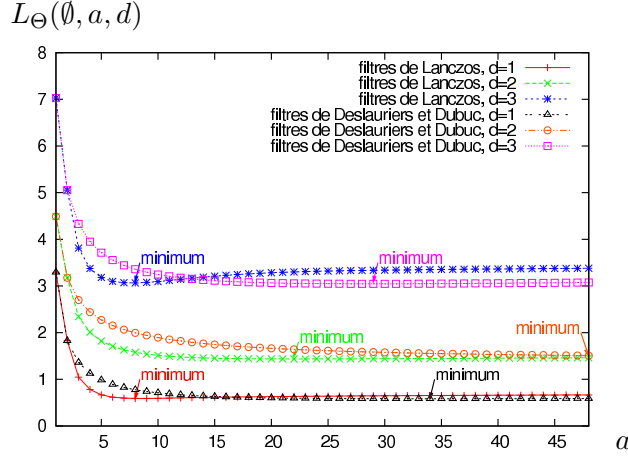


FIG. 5.1 – Exemple de minimisation de  $L_{\Theta}$  pour l’optimisation de la longueur du filtre ( $a \in [1..48]$ ) et la sélection de la dimension de décomposition ( $d \in \{1, 2, 3\}$ ). Il correspond au premier niveau de décomposition de l’IRM VHP-Harvard-3D, avec les filtres  $L_a$  ou  $D_a^2$ .

## 5.2.2 Construction de l’arbre de décomposition

L’arbre de décomposition  $\Theta$  est construit itérativement. Ce processus peut être exprimé récursivement, en commençant par la racine  $\Theta_{\emptyset}$ .

### 5.2.2.1 Décomposition d’une composante image

Si la chaîne d’index  $n$  du nœud courant  $\Theta_n$  ne contient aucun  $H$ ,  $\beta_n$  est une composante image. Dans ce cas, si une certaine condition  $C_I(\Theta, n)$  est vérifiée (ex. une résolution d’image minimum), le nœud n’est pas décomposé,  $\Delta_n$  est mis à 0 et l’arbre de décomposition est terminé.

Si la condition d’arrêt  $C_I(\Theta, n)$  n’est pas vérifiée,  $d$  est optimisé pour minimiser  $L_{\Theta}(n, \hat{a}_{n,d}, d)$  (le  $d$  optimal est noté  $\hat{d}_n$ ).  $\Delta_n$  est mis à  $\hat{d}_n$ ,  $\Omega_n$  à  $\hat{a}_{n,d}$ , le nœud  $\Theta_n$  est décomposé et ses deux enfants sont ajoutés en utilisant :

$$\begin{aligned} \beta_{n:H} &= P_{\Omega_n, \Delta_n} [\beta_n], \\ \beta_{n:L} &= U_{\Omega_n, \Delta_n} [\beta_n], \end{aligned} \quad (5.10)$$

et la récursivité continue sur  $\Theta_{n:H}$ , puis sur  $\Theta_{n:L}$ .

### 5.2.2.2 Décomposition d’une bande fréquentielle

Si  $\beta_n$  n’est pas une composante image (i.e. c’est une bande fréquentielle) et si une certaine condition  $C_F(\Theta, n)$  (ex. une taille minimale),  $\Theta_n$  n’est pas décomposé,  $\Delta_n$  est mis à 0 et la récursion continue sur les autres nœuds.

Si  $C_F(\Theta, n)$  n’est pas vérifiée,  $d$  est optimisé pour minimiser  $L_{\Theta}(n, \hat{a}_{n,d}, d)$ , comme

pour une composante image. Ensuite, si l'inéquation suivante est vérifiée :

$$L_{\Theta}(n, \hat{a}_n, \hat{d}_n) < \frac{1}{\text{Card}(\beta_n)} \sum |\beta_n|, \quad (5.11)$$

$\Delta_n$  est mis à  $\hat{d}_n$ ,  $\Omega_n$  à  $\hat{a}_n$ , le nœud  $\Theta_n$  est décomposé, ses deux enfants ajoutés en utilisant (5.10) et la récursion continue sur  $\Theta_{n:H}$ , puis sur  $\Theta_{n:L}$ .

Si (5.11) n'est pas vérifiée, la décomposition n'apporte rien. Donc  $\Theta_n$  n'est pas décomposé,  $\Delta_n$  est mis à 0 et la récursivité continue sur les autres nœuds.

### 5.3 Méthode de codage

Le codeur pour la décomposition optimisée (ODC : *optimized decomposition coder*), développé pour cette étude, utilise un principe très similaire à LZC (*layered zero coding*) [TZ94], qui peut se résumer par l'utilisation d'un codeur arithmétique binaire adaptatif pour la compression de chaque plan de bits des coefficients de chaque sous-bande, avec l'aide d'une sélection de contextes de codage 2D (ODC) ou 3D (ODC-3D).

La structure de la décomposition est d'abord transmise en début du flux de données.  $\Theta$  est codé à l'aide d'un parcours en profondeur préfixe de l'arbre, en transmettant un bit à 1 lorsque le nœud traversé possède des fils ou un bit à 0 sinon. Pour chaque nœud  $n$  possédant des fils (i.e. décomposé),  $\Theta_n$  et  $\Omega_n$  sont également transmis. Aucune compression n'est faite (équiprobabilité des symboles potentiels), le coût de transmission de la structure de décomposition avec les paramètres des filtres (indices) représentant généralement moins de  $10^{-3}$  bit par pixel pour une coupe  $512 \times 512$ .

L'en-tête des informations relatives à l'image (largeur, hauteurs, ...) contient également le nombre de bits maximal  $b_{\max}$  requis pour la valeur absolue d'un coefficient.

Le codage se fait par plan de bits sur les valeurs absolues des coefficients. Lorsqu'un coefficient devient significatif, son signe est également codé. Pour que la représentation progressive soit assez efficace, l'ordonnement du codage par plan de bits s'effectue en fonction de la décomposition. Afin de favoriser les bandes basses fréquences qui contiennent généralement le plus d'informations, les sous-bandes (feuilles) sont d'abord triées à l'aide d'un parcours main gauche de l'arbre de décomposition (en considérant que  $\Theta_{n:L}$  et  $\Theta_{n:H}$  sont respectivement les fils gauche et droit de  $\Theta_n$ ). Ensuite, chaque sous-bande  $\Theta_k$  se voit attribuer un indice de décalage  $s_{\Theta_k} = \lfloor \log_2 \sqrt{E(f_k^*)} \rfloor$  pour le codage de ses plans de bits, avec  $E(f_k^*)$  l'énergie du filtre de synthèse  $f_k^*$  pour  $\Theta_k$ .  $s_{\Theta_k}$  permet de s'approcher d'une normalisation des coefficients afin de mieux prendre en considération la non orthonormalité de la décomposition.

Le codage s'effectue alors du plan de bits  $b_{\max} + \max_k s_{\Theta_k}$  au plan de bits  $\min_k s_{\Theta_k}$ . Pour un plan de bits  $b$ , si la sous-bande  $\Theta_k$  (en suivant l'ordre de parcours) vérifie  $b_{\max} + s_{\Theta_k} \leq b \leq s_{\Theta_k}$ , alors le plan de bits  $b - s_{\Theta_k}$  de  $\Theta_k$  est codé.

Pour un codage scalable en résolution, il suffit de conserver le même ordonnancement des feuilles, et de transmettre la totalité des plans de bits de la sous-bande associée à chacune d'entre elle.

## 5.4 Résultats pour les images médicales volumiques

L'approche présentée a été appliquée à la compression de tomographies et d'IRM<sup>1</sup>, pour un modèle bidimensionnel (compression coupe par coupe) et volumique (compression de la totalité du volume). La TAB. 5.1 reporte le nombre de bits par pixel (bpp) moyen après compression de chacun des ensembles de test (les résultats ne sont pas exactement les mêmes que dans [TL10a], le codeur entropique ayant été légèrement amélioré entre temps). La taille minimale pour la composante image a été fixée à  $16 \times 16$  en 2D et fixée à  $16 \times 16 \times \max(1, \lfloor Z/32 \rfloor)$  en 3D pour chacun des volumes ayant une taille initiale  $X \times Y \times Z$ . Le paramètre  $a$  pour les filtres de Lanczos (L-) et de Deslauriers-Dubuc (D-) a été optimisé dans l'intervalle  $[1..16]$  pour une décomposition optimisée sans contrainte (-OD : *optimized decomposition*) et pour une décomposition forcée de suivre le schéma dyadique traditionnelle de Mallat (-dy) et pour laquelle le filtre est optimisé par orientation (horizontale, verticale, transversale) pour obtenir l'ensemble des sous-bandes d'un même niveau de décomposition dyadique.

Le fait d'utiliser la contrainte suivante lors de l'optimisation :

$$\left\lfloor \frac{\Lambda_\delta(\beta_{\mathbf{L}^k N}) + 1}{2} \right\rfloor = \Lambda_\delta(\beta_{\mathbf{L}^{(k+1)N}}), \quad \forall \delta \in \{1 \dots N\}, \quad (5.12)$$

avec  $\mathbf{L}^0$  la chaîne vide,  $\mathbf{L}^k$  la chaîne contenant  $k$  symboles  $\mathbf{L}$  successifs, et  $\Lambda_\delta(\beta_n)$  la longueur de la bande  $\beta_n$  le long de la dimension  $\delta$ , peut permettre de générer une décomposition scalable en résolution (à condition de changer l'ordonnancement des plans de bits compressés). Parce que cette contrainte n'altère que très peu les résultats de la compression sans perte (seule une légère augmentation du débit, inférieure à 0.01 bpp, a pu être constatée), les résultats ne sont pas reportés.

Les résultats de l'optimisation de la taille des supports des filtres (L-OD, L-dyadic) sont comparés à ceux obtenus avec les ondelettes 5/3 (5/3-OD et 5/3-dyadic) traditionnellement employées en compression sans perte, et sont confrontés à ceux obtenus par quatre logiciels de référence 2D : les codeurs prédictifs (DPCM) CALIC et JPEG-LS (JLS), et les codeurs ondelettes SPIHT et JPEG-2000 (J2K); et pour le standard de compression d'images volumique JPEG-2000-P10 (extension volumique de JPEG-2000 : J2K-3D) utilisant une décomposition dyadique sur 5 niveaux à l'aide du logiciel openjpeg<sup>2</sup>. Les meilleurs résultats sur l'ensemble des algorithmes sont soulignés, tandis que les meilleurs résultats par catégories (DPCM 2D non scalable, 2D scalable et 3D scalable) sont en gras.

<sup>1</sup>Quelques résultats obtenus sur les lames virtuelles sont présentés et commentés dans le prochain chapitre : 6.5.3.2

<sup>2</sup><http://www.openjpeg.org/>

Comme mentionné dans la description des bases d'images 1.1.3.1, les valeurs négatives ont été forcées à 0 par simplicité. Ceci n'affecte que les CT de MeDEISA qui possèdent des valeurs égales à -3000 en dehors de la zone (disque) de reconstruction. Ces pixels n'ont aucune signification diagnostique, et la discontinuité induite entre l'intérieur (valeurs positives) et l'extérieur du disque a un impact non négligeable sur la compression, en particulier pour les ondelettes à large support. Pour exemple, le débit moyen de la compression 3D des CTs de MeDEISA serait augmenté d'environ 6% avec les "L-dy", et de 3% avec les "L-OD" et seulement de 1% pour les "5/3-dy" si cette simplification n'avait pas été faite.

#### 5.4.1 Analyse en compression 2D

Comme au chapitre précédent, l'algorithme DPCM CALIC se montre efficace sur les images les plus bruitées, mais ODC est plus performant sur les images moins bruitées et sur la majorité des IRM, même si le codage entropique est effectué par plan de bits (et donc théoriquement moins optimal).

En comparant les résultats (TAB. 5.1) de ODC obtenus pour la décomposition dyadique 5/3 avec les autres codeurs ondelettes, SPIHT (qui utilise une transformée légèrement différente) semble lui être à peu près équivalent. Cependant SPIHT n'offre pas la flexibilité de JPEG-2000 quant à l'organisation du flux de données compressées. Bien que ODC fonctionne sur un schéma de codage par plan de bits simplifié en comparaison à JPEG-2000, il peut permettre de proposer une représentation progressive en qualité ou en résolution. Il montre de meilleures performances de compression sans perte que JPEG-2000 : sur l'ensemble de la base d'images, le débit moyen est réduit de 2.13% en 2D et de 3.79% en 3D. Cette amélioration peut en partie être expliquée par la syntaxe plus complexe dans le flux de données de JPEG-2000 (marqueurs, etc...) et par l'usage de *code blocks* qui lui offrent plus de flexibilité (surtout en régime progressif ou avec pertes). Le codeur de JPEG-2000 est également optimisé pour être assez rapide en exécution (certains algorithmes de codage sont simplifiés pour accélérer l'exécution au détriment de légères pertes de performances), ce qui n'est pas le cas de ODC.

Considérant l'approche présentée dans sa totalité (L-OD, D-OD) utilisant le codeur ODC, il peut déjà être constaté que les résultats obtenus à l'aide de la famille des filtres de Lanczos (L) sont très similaires à ceux obtenus avec la famille des filtres de Deslauriers et Dubuc (D), bien que ces derniers soient théoriquement plus optimaux (nombre de moments nuls). Sur VHP-Harvard-3D, les "L" permettent même une amélioration de plus de 3% par rapport aux "D" qui s'explique par la limitation de l'optimisation de  $a$  dans l'intervalle [1..16]. D'après la FIG. 5.1, la famille "D" pourrait donner des résultats équivalents avec des filtres plus longs (et donc un temps d'optimisation et un temps de transformation plus élevés).

Cette constatation permet de faire remarquer que l'optimisation des filtres "L" a généralement une convergence plus rapide que celle des filtres "D" sur les images les plus douces. Et, même si ils ne sont pas optimaux, ils permettent d'obtenir des résultats, certes un peu moindres, mais similaires aux filtres "D" pour des supports de tailles plus



		Débits moyens (bpp)													
		2D DPCM		2D scalable							3D scalable				
		Base	CALIC	JLS	SPIHT	J2K	ODC				J2K-3D	ODC-3D			
							5/3-dy	5/3-OD	L-dy	L-OD		D-OD	5/3-dy	L-dy	L-OD
CT	CIPR (8bits)	<b>1.92</b>	1.96	2.19	2.17	2.14	2.17	<b>2.04</b>	2.08	2.07	1.56	1.43	1.38	1.35	<b>1.34</b>
	MeDEISA	<b>4.72</b>	4.87	4.74	4.84	4.74	4.73	4.18	<b>4.15</b>	4.16	4.59	4.42	3.87	<b>3.85</b>	<b>3.85</b>
	PMR	<b>3.69</b>	3.99	3.71	3.88	3.78	3.80	3.23	3.23	<b>3.21</b>	3.29	3.11	2.77	2.75	<b>2.73</b>
	VHP-Female	<b>4.65</b>	4.73	4.87	4.91	4.81	4.81	4.77	<b>4.76</b>	<b>4.76</b>	4.73	4.56	<b>4.53</b>	4.56	4.56
	VHP-Male	<b>4.75</b>	4.82	4.97	5.01	4.91	4.91	4.89	<b>4.88</b>	<b>4.88</b>	4.85	4.68	<b>4.67</b>	4.68	4.68
	VHP-Harvard	<b>5.03</b>	5.05	5.28	5.35	5.23	<b>5.19</b>	5.24	5.20	5.20	5.03	4.82	4.82	4.78	4.78
	totalité	<b>4.60</b>	4.70	4.78	4.84	4.74	4.73	4.59	<b>4.57</b>	<b>4.57</b>	4.61	4.43	<b>4.30</b>	4.31	<b>4.30</b>
MRI	CIPR (8bits)	<b>2.69</b>	2.79	2.82	2.96	2.87	2.85	2.67	<b>2.62</b>	<b>2.62</b>	2.12	1.88	1.82	<b>1.67</b>	1.68
	MeDEISA	<b>3.15</b>	3.29	3.24	3.33	3.25	3.23	3.20	<b>3.17</b>	<b>3.17</b>	3.19	3.07	3.01	<b>2.96</b>	<b>2.96</b>
	PMR	<b>3.00</b>	3.23	3.15	3.15	3.10	3.09	2.74	<b>2.72</b>	<b>2.72</b>	3.07	2.96	2.64	<b>2.53</b>	2.54
	VHP-Female	<b>4.41</b>	4.62	4.50	4.61	4.49	4.46	4.39	4.37	<b>4.36</b>	4.62	4.45	4.36	<b>4.29</b>	<b>4.29</b>
	VHP-Male	<b>4.90</b>	5.08	4.91	5.01	4.86	4.84	4.82	<b>4.79</b>	<b>4.79</b>	5.01	4.82	4.78	<b>4.71</b>	<b>4.71</b>
	VHP-Harvard	<b>4.68</b>	4.85	4.96	5.05	4.93	4.89	4.92	<b>4.88</b>	<b>4.88</b>	5.01	4.80	4.80	<b>4.75</b>	<b>4.75</b>
	VHP-Harvard-3D	<b>3.70</b>	4.13	3.58	3.82	3.73	3.71	2.73	<b>2.63</b>	2.72	3.45	3.31	2.50	<b>2.32</b>	2.37
totalité	<b>3.84</b>	4.07	3.90	4.01	3.91	3.89	3.63	<b>3.59</b>	3.60	3.86	3.71	3.47	<b>3.38</b>	3.39	
totalité		<b>4.47</b>	4.59	4.63	4.70	4.60	4.59	4.42	<b>4.41</b>	<b>4.41</b>	4.48	4.31	4.16	<b>4.15</b>	<b>4.15</b>

TAB. 5.1 – Débits moyens pour la compression sans perte

petites.

Par rapport à la décomposition de référence (5/3-dy), les plus fortes améliorations en compression de L-OD sont obtenues sur les données les plus corrélées (images douces contenant le moins d'information/bruit aléatoire) : pour les CT sont rapportées des réductions moyennes du débit de 14.55% pour PMR et de 12.45% pour MeDEISA ; pour les coupes IRM, l'amélioration la plus significative est obtenue sur VHP-Harvard-3D, avec une réduction de près de 29.5%. Pour PMR, qui contient également un bruit d'acquisition (ou plutôt de construction) corrélé, la taille est réduite de 12.25%. Même si elle est moins significative, la réduction du débit entre 1.4% et 2.9% sur les IRM avec une certaine corrélation dans le bruit de construction (MeDEISA, VHP-Male/Female) rend déjà ODC compétitif avec CALIC. Sur les ensembles d'images les plus bruitées (CT de VHP-Male/Female/Harvard, IRM de VHP-Harvard), l'amélioration apportée par l'optimisation est inférieure à 1%.

La comparaison du passage de L-dy à L-OD avec le passage de 5/3-dy à 5/3-OD, montre que la structure de l'arbre décomposition a plus d'impact lorsqu'elle est optimisée conjointement avec la taille du support que lorsqu'elle est optimisée uniquement pour les filtres 5/3. Les meilleurs gains liés à l'optimisation de la structure sont obtenus sur les IRM de VHP-Harvard-3D (3.59%) et de CIPR (1.86%) pour les "L", contre 0.8% pour les "5/3" sur les IRM de VHP-Harvard. Pour les autres bases d'images la structure n'apporte qu'une amélioration inférieure à 1%. Lorsque comparé à l'apport de l'optimisation du support des filtres (amélioration de 5/3-dy à L-dy et de 5/3-OD à L-OD) l'impact de l'optimisation de la structure de l'arbre de décomposition est assez faible pour les images tomographiques, et celle-ci semble être mieux adaptée pour les IRM.

Même si la simplicité du critère d'optimisation  $\mathcal{W}_\Theta$  rend très certainement les résultats sous-optimaux, et provoque parfois une légère baisse des performances sur certaines bases d'images (principalement sur les images tomographiques), en comparaison à une compression basée sur une décomposition dyadique en ondelettes 5/3, le schéma d'optimisation présenté permet de bien s'adapter aux variétés des contenus : les performances de compression sont améliorées sur les images les plus douces et restent similaires sur les images les plus bruitées. Des exemples de décompositions obtenues en suivant cette approche sont donnés dans FIG. 5.2 et FIG. 5.3.

Pour ce qui est du rapport débit/PSNR obtenu avec le codage progressif présenté précédemment, les mêmes constatations que pour la compression sans perte peuvent être faites : le PSNR est amélioré, en comparaison à l'utilisation des ondelettes 5/3, sur les images les plus corrélées, et il reste à peu près équivalent sur les images les plus bruitées. La FIG. 5.4 illustre ce comportement. Pour une meilleure lisibilité, les résultats pour la décomposition dyadique ne sont pas présentés, étant très proches de ceux obtenus avec la décomposition adaptative.

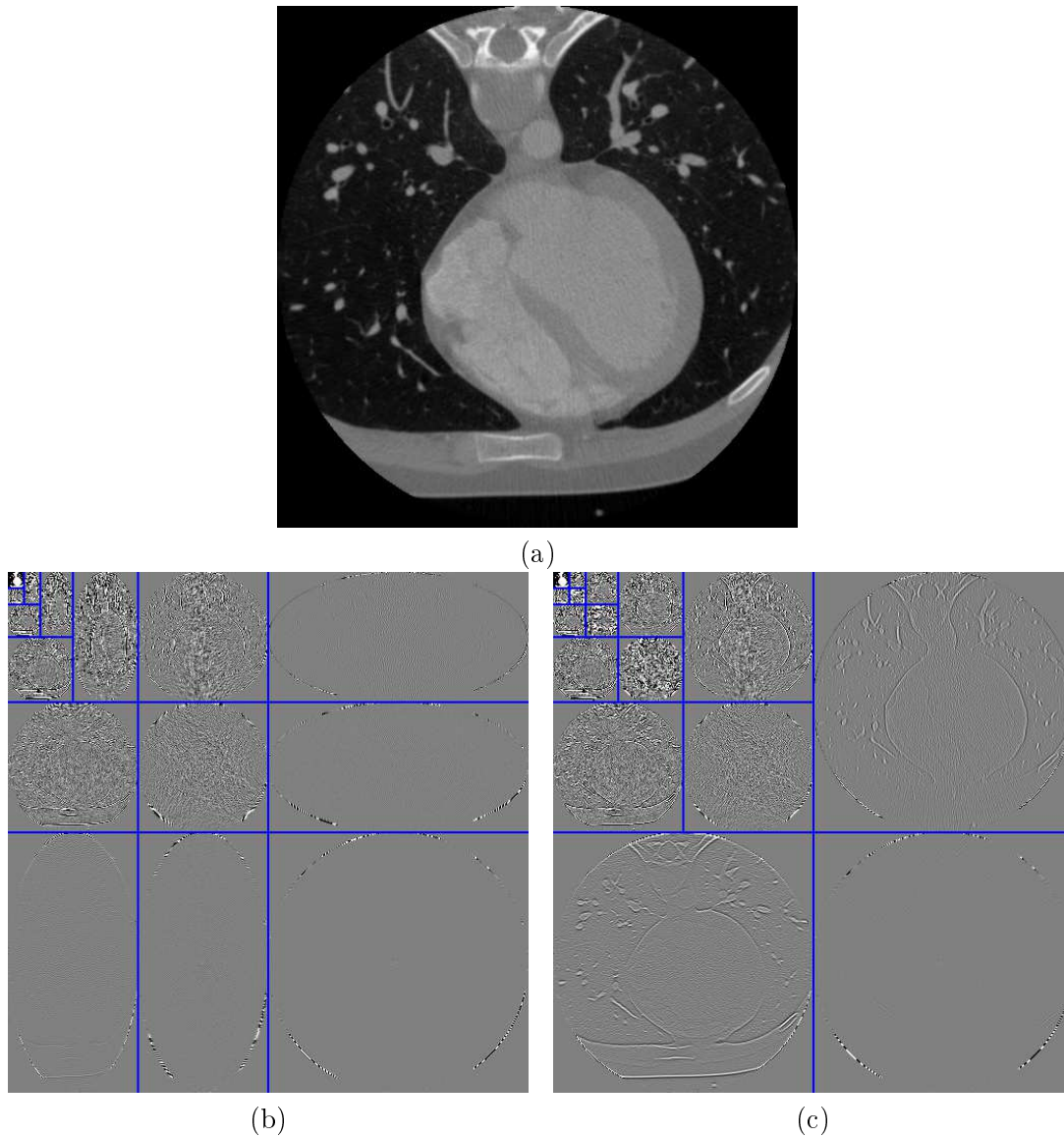


FIG. 5.2 – Une coupe tomographique (a) de MeDEISA (CT\_data\_1(105)), les valeurs de l'image entre 0 et 2272 sont mises à l'échelle pour l'affichage en 255 niveaux de gris; et le résultat de ses décompositions (b) avec la base optimisée L-OD et (c) avec la transformée dyadique en ondelettes  $5/3$ . Les valeurs des coefficients sont tronquées entre  $-128$  et  $127$  et mises à l'échelle pour l'affichage en 255 niveaux de gris.

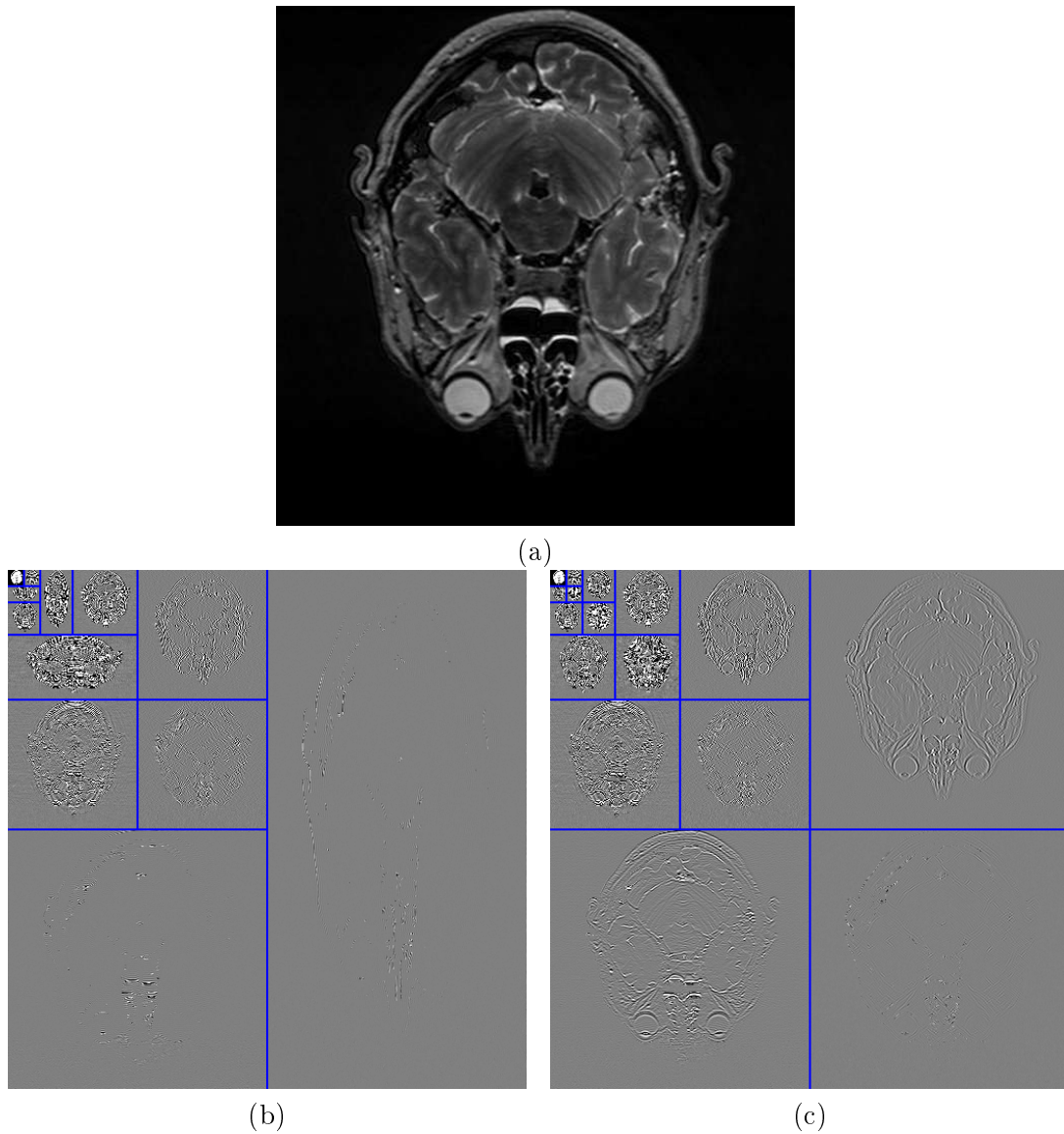


FIG. 5.3 – Une coupe IRM-3D (a) de VHP-Harvard-3D (T2\_512(115)), les valeurs de l'image entre 0 et 1032 sont mises à l'échelle pour l'affichage en 255 niveaux de gris ; et le résultat de ses décompositions (b) avec la base optimisée L-OD et (c) avec la transformée dyadique en ondelettes 5/3. Les valeurs des coefficients sont tronquées entre  $-64$  et  $63$  et mises à l'échelle pour l'affichage en 255 niveaux de gris.

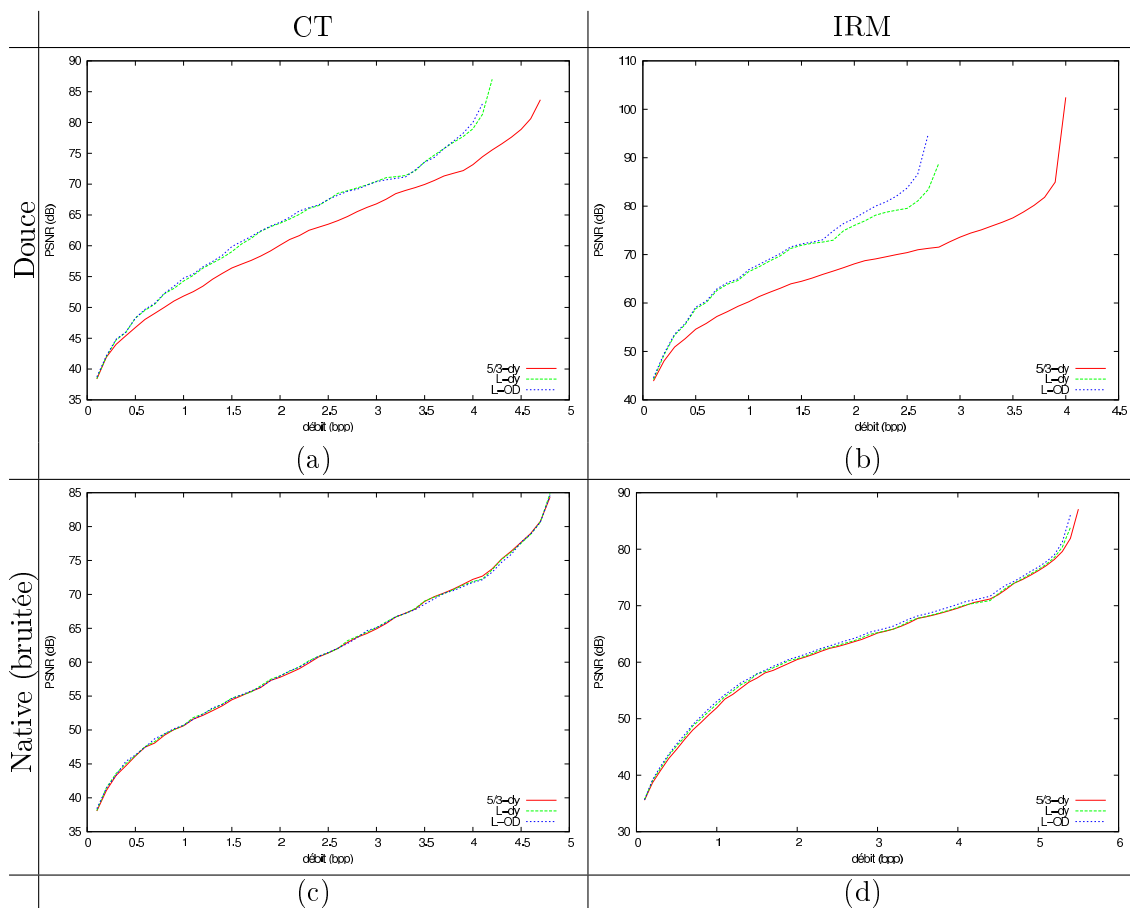


Figure 5.4: Courbes débit/distorsion obtenues pour la représentation progressive en qualité « *lossy to lossless* » de coupes IRM et tomographiques, douces et natives, après l'optimisation de la décomposition pour la compression sans perte. (a) MeDEISA (CT\_data\_1(105)), (b) VHP-Harvard-3D (T2\_512(115)), (c) NLM-VHP Male (normalCT(248)), (d) NLM-VHP Male (PD-1(21)).

### 5.4.2 Analyse en compression 3D

Pour la compression volumique, les remarques sont encore similaires au cas 2D. Les volumes les plus corrélés sont mieux compressés avec l'optimisation des filtres et de la décomposition L-OD qu'avec une approche 5/3 dyadique. Le gain de compression sur l'IRM VHP-Harvard-3D est de 29.9%, et pour les CT peu bruitées de MeDEISA et de PMR il est d'environ 12%. Sur les ensembles d'images bruitées, l'amélioration est plus faible. On ne note aucune amélioration pour les images tomographiques de VHP-Male/Female : pour ces images volumiques, qui contiennent les acquisitions de corps entiers, le type de bruit et de contenu n'est pas réparti de façon homogène tout au long de l'axe z, l'arbre de décomposition ainsi que l'optimisation des filtres sont alors beaucoup moins adaptés. Une compression indépendante d'une succession de volumes plus petits (comme les GOP en vidéo) permettrait très certainement d'obtenir de meilleurs résultats sur ce type d'images en adaptant mieux la décomposition localement.

Comme en 2D, l'arbre de décomposition prend plus d'importance sur les volumes IRM avec plus de 7% de réduction du débit pour CIPR et VHP-Harvard-3D (en comparant L-dy à L-OD), alors que pour les CT la plus grosse amélioration porte sur les données de CIPR avec seulement une réduction de 0.03 bpp (soit environ 2%).

Sur la totalité des bases de test, en comparaison à l'approche 2D, l'approche volumique permet d'économiser un peu moins de 6% de l'espace de stockage. Les gains les plus importants portent sur les images les moins bruitées : pour les images 8 bits, qui sont très corrélées en inter-coupe et ont probablement subi un filtrage sur cette dimension, on note plus de 35% d'économie ; pour les approches de construction volumique, les CT de PMR et l'IRM de VHP-Harvard-3D, les gains sont respectivement de 15% et 12.9%. En dehors de ces images : pour les IRM, c'est la résolution assez faible de l'axe transversal qui est la plus pénalisante, avec moins de 2% de gain pour les volumes de VHP-Male et VHP Female ; pour les tomographies dont la résolution transversale est généralement plus fine, le bruit d'amplitude élevée et sans corrélation inter-coupe aurait tendance à expliquer l'amélioration d'à peine plus de 4% pour VHP-Male et VHP-Female. Un autre facteur est, comme il a été précédemment mentionné, le fait que ces volumes contiennent des coupes pour un corps entiers, et que l'adaptation de la décomposition soit globale. Ainsi cette dernière ne peut pas prendre en considération des particularités plus locales, alors que la reconstruction tomographique diffère en fonction de localisation. Par exemple, au niveau de la tête, les images sont un peu plus douces car un facteur de zoom est appliqué lors de la construction des images, tandis que pour l'abdomen la pleine résolution est utilisée.

## 5.5 Extensions

### 5.5.1 Modèle prédictif pour la compression presque sans perte

Une simplification de l'approche présentée consiste à n'utiliser que l'étape de prédiction dans le schéma de lifting. L'algorithme devient alors un algorithme prédictif par

interpolation et peut donc être étendu en presque sans perte (dans le chapitre précédent, cet aspect est plus détaillé et en particulier pour IHINT qui utilise un algorithme prédictif par interpolation).

L'extension en presque sans perte reste un peu plus complexe que celle de IHINT (c.f. 4.1.4), chaque sous-bande pouvant subir des re-décompositions par prédiction. Pour simplifier l'implémentation, l'optimisation de la décomposition est complètement effectuée pour la prédiction sans perte. Une fois l'arbre de décomposition et les longueurs des filtres obtenus, on applique l'algorithme suivant en partant de l'image originale pour effectuer son approximation presque sans perte :

L'arbre de décomposition est parcouru à l'aide d'un parcours main gauche. Lorsqu'un nœud est rencontré de manière :

- préfixe, si il est un nœud interne (décomposition), les valeurs des pixels de la sous-bande correspondante sont réorganisées en deux sous-bandes (pairs/impairs) comme si elle avait été décomposée, mais sans appliquer la prédiction. Dans le cas où le nœud est une feuille (sous-bande finale), son contenu est  $\tilde{Q}_\delta$  approximé.
- infixe, si il est un nœud interne, la prédiction de la sous-bande droite (pixels précédemment impairs) à l'aide de la sous-bande gauche (pixels pairs) est appliquée.
- postfixe, si il est un nœud interne, la prédiction est annulée et les pixels des deux sous-bandes sont réorganisés afin de reformer la sous-bande correspondante à l'approximation (respectant le PAE  $\delta$ ) de la sous-bande initiale.

Un fois l'arbre totalement parcouru, l'image finale (respectant le PAE  $\delta$ ) est obtenue. Il suffit alors d'appliquer la décomposition complète, et de  $Q_\delta$  quantifier chacune des sous-bandes.

Les résultats pour la compression sans perte avec cette approche (PAE=0) sont présentés dans la TAB. 5.2. Globalement les mêmes commentaires que précédemment peuvent être faits sur l'optimisation. On peut également noter que le fait de supprimer le filtrage passe-bas (seconde étape de lifting) tend à réduire les performances de codage : sur l'ensemble de toute la base, le débit est augmenté d'environ 1.4% pour la décomposition prédictive adaptative, en comparaison à la décomposition en ondelettes adaptative originale.

Les résultats de la compression presque sans perte pour un faible PAE de 4 sont présentés dans la TAB. 5.3. En ce qui concerne le débit, en comparaison à la compression sans perte, quelques changements de comportement commencent à apparaître. Dans le cas 2D, les écarts sont moins importants : sur la totalité de la base d'images, l'optimisation complète "L-OD" n'apporte plus que 2.26% de gain de compression (relativement à "3-dy") soit environ 1% de moins qu'en sans perte. Ceci est notamment flagrant sur les tomographies, et en particulier celles de PMR pour lesquelles le gain était supérieur à 10% en sans perte alors qu'il ne l'est plus que d'à peine à 1%. L'optimisation reste tout de même avantageuse sur les IRM, avec une amélioration moyenne de la compression de 5.4%. Avec le passage en volumique (ODC-3D) le gain est réduit encore davantage en tomographie. Sur PMR, il était de 7.19% en sans perte, alors qu'en presque sans perte il devient négatif avec un taux de compression de 10.71:1 pour "L-OD" contre 13.19:1 pour "3-dy", soit -23%! Le gain obtenu sur les IRM reste proche de celui en sans perte

		Débits moyens (bpp)									
		Base	2D scalable					3D scalable			
			predictive ODC					predictive ODC-3D			
		3-dy	3-OD	L-dy	L-OD	D-OD	3-dy	L-dy	L-OD	D-OD	
CT	CIPR (8bits)	2.08	2.14	<b>2.02</b>	2.08	2.07	1.40	1.39	<b>1.34</b>	<b>1.34</b>	
	MeDEISA	4.63	4.62	4.14	<b>4.11</b>	<b>4.11</b>	4.35	<b>3.88</b>	<b>3.88</b>	<b>3.88</b>	
	PMR	3.72	3.78	<b>3.31</b>	3.34	<b>3.31</b>	3.06	2.85	2.84	<b>2.82</b>	
	VHP-Female	4.87	4.88	<b>4.86</b>	<b>4.86</b>	<b>4.86</b>	<b>4.71</b>	4.73	4.73	4.73	
	VHP-Male	4.98	4.98	4.98	4.98	<b>4.97</b>	<b>4.84</b>	4.86	4.85	4.85	
	VHP-Harvard	5.26	<b>5.21</b>	5.26	5.22	5.22	<b>4.97</b>	4.98	4.98	4.98	
	totalité	4.76	4.76	4.64	4.64	<b>4.63</b>	4.53	4.45	4.45	<b>4.44</b>	
MRI	CIPR (8bits)	2.87	2.85	2.72	<b>2.65</b>	<b>2.65</b>	1.89	1.89	<b>1.67</b>	1.69	
	MeDEISA	3.31	3.28	3.28	3.25	<b>3.24</b>	3.18	3.15	<b>3.06</b>	<b>3.06</b>	
	PMR	3.08	3.08	2.78	<b>2.75</b>	2.76	2.98	2.71	<b>2.57</b>	2.58	
	VHP-Female	4.57	4.53	4.51	4.47	<b>4.46</b>	4.89	4.55	4.42	<b>4.41</b>	
	VHP-Male	4.94	4.91	4.92	<b>4.88</b>	<b>4.88</b>	4.96	4.96	<b>4.82</b>	<b>4.82</b>	
	VHP-Harvard	5.04	4.98	5.05	<b>4.99</b>	<b>4.99</b>	4.98	4.99	<b>4.87</b>	<b>4.87</b>	
	VHP-Harvard-3D	3.67	3.67	2.80	<b>2.66</b>	2.75	3.24	2.61	<b>2.36</b>	2.40	
	totalité	3.94	3.92	3.71	<b>3.65</b>	3.67	3.78	3.61	<b>3.46</b>	3.47	
	totalité	4.62	4.62	4.49	<b>4.47</b>	<b>4.47</b>	4.41	4.31	<b>4.28</b>	<b>4.28</b>	

TAB. 5.2 – Débits moyens pour la compression sans perte avec la version uniquement prédictive

		Débits moyens/PSNR moyens (bpp/dB)						
		Base	2D scalable			3D scalable		
			predictive ODC			predictive ODC-3D		
		3-dy	L-dy	L-OD	3-dy	L-dy	L-OD	
CT	CIPR (8bits)	<b>0.64</b> /42.34	0.67/42.13	0.71/42.08	<b>0.31</b> /42.52	0.38/42.17	0.38/42.17	
	MeDEISA	2.25/64.65	<b>1.99</b> /64.61	<b>1.99</b> /64.60	2.06/64.17	<b>1.86</b> /64.12	1.89/64.11	
	PMR	1.28/64.84	<b>1.21</b> /64.50	1.27/64.48	<b>0.91</b> /64.95	1.06/64.47	1.12/64.38	
	VHP-Female	<b>2.43</b> /64.67	<b>2.43</b> /64.65	<b>2.43</b> /64.63	<b>2.31</b> /64.81	2.33/64.78	2.32/64.76	
	VHP-Male	<b>2.58</b> /64.81	2.59/64.79	<b>2.58</b> /64.78	<b>2.49</b> /63.34	2.50/63.32	<b>2.49</b> /63.27	
	VHP-Harvard	2.75/64.59	2.75/64.58	<b>2.71</b> /64.58	<b>2.50</b> /63.37	2.51/63.36	2.51/63.32	
	totalité	2.36/64.23	<b>2.32</b> /64.18	<b>2.32</b> /64.16	2.21/63.53	<b>2.20</b> /63.47	<b>2.20</b> /63.43	
MRI	CIPR (8bits)	0.79/41.85	0.81/41.69	<b>0.78</b> /41.72	<b>0.37</b> /42.34	0.47/41.86	0.41/41.91	
	MeDEISA	1.07/65.59	1.09/65.54	<b>1.05</b> /65.54	1.01/64.68	1.03/64.61	<b>0.95</b> /64.69	
	PMR	1.13/65.04	1.08/65.06	<b>1.05</b> /65.07	1.13/64.73	1.10/64.72	<b>0.97</b> /64.78	
	VHP-Female	1.75/64.58	1.75/64.53	<b>1.70</b> /64.54	1.81/64.37	1.82/64.30	<b>1.69</b> /64.35	
	VHP-Male	2.05/64.37	2.06/64.35	<b>2.02</b> /64.35	2.11/64.20	2.12/64.17	<b>1.99</b> /64.18	
	VHP-Harvard	2.52/64.66	2.53/64.66	<b>2.48</b> /64.65	2.51/64.57	2.51/64.57	<b>2.40</b> /64.57	
	VHP-Harvard-3D	1.25/64.94	1.07/64.87	<b>0.98</b> /64.94	1.03/65.03	1.01/64.67	<b>0.85</b> /64.82	
	totalité	1.48/63.75	1.44/63.71	<b>1.40</b> /63.73	1.43/63.49	1.43/63.36	<b>1.31</b> /63.43	
	totalité	2.21/64.15	2.17/64.10	<b>2.16</b> /64.09	2.08/63.53	2.07/63.45	<b>2.05</b> /63.43	

TAB. 5.3 – Débits et PSNR moyens pour la compression presque sans perte avec la version uniquement prédictive (PAE=4)



(8.39% contre 8.47%). Enfin, les gains de 8.25% sur les CT filtrées de MeDEISA et de 17.5% sur VHP-Harvard-3D sont toujours non négligeables.

Les mauvais résultats obtenus sur les CT de PMR (et les résultats peu concluants en général sur les CT) peuvent s'expliquer par le procédé utilisé pour simplifier l'implémentation qui consiste à utiliser l'optimisation pour la prédiction sans perte. Ainsi, les erreurs d'approximations qui auront lieu après quantification ne sont pas prises en compte. Ceci peut avoir deux effets néfastes : i) une baisse du PSNR : les erreurs dues à la quantification, qui ne sont pas uniformément réparties après reconstruction, vont avoir tendance à perturber l'interpolation pour l'approximation des plus hautes résolutions et donc à générer du bruit sur la prédiction. ii) une augmentation du débit : le bruit de prédiction se répercute sur les sous-bandes suivantes et, si il est trop important en comparaison au PAE, reste présent après quantification.

Une décomposition assez riche (en terme de nombre de nœuds dans l'arbre) et la longueur du support des filtres peuvent amplifier ce phénomène (ce qui est en particulier le cas de PMR), et la compression est alors moins efficace. Ce problème est illustré dans la FIG. 5.5. Une solution consisterait à prendre en compte la quantification lors de l'optimisation de la décomposition. Cependant elle serait beaucoup plus complexe en temps de calcul, notamment à cause de la quantification qui doit avoir lieu sur les feuilles de l'arbre de décomposition : la décomposition jusqu'à cette feuille doit donc être préalablement fixée et, à moins d'utiliser de nouvelles heuristiques, une approche plus gloutonne serait alors nécessaire.

Contrairement à HOP et IHINT tels qu'implémentés au chapitre précédent, la décomposition n'est pas complète (jusqu'à ce que la basse résolution ne soit plus qu'un unique pixel) afin que le codeur par plan de bits ne soit pas trop pénalisé (ce qui serait le cas si les sous-bandes étaient trop petites). Ainsi, pour le modèle prédictif de cette section, lorsque le pas de quantification est important, la quantification de l'image basse résolution peut fortement influencer sur la répartition des résidus de prédiction, et explique des résultats étranges lorsque le PAE est élevé.

Pour illustrer ce phénomène, on prend pour exemple une interpolation linéaire (filtre "3", i.e.  $a = 1$ ) : l'image basse résolution originale contient deux pixels valant  $p = 238$  pour prédire par interpolation ( $\hat{x} = 238$ ) une valeur de  $x = 235$  dans la sous-bande suivante. En sans perte, le résidu est alors  $r = -3$ . Pour un PAE  $\delta = 10$ ,  $\tilde{Q}_\delta(p) = 11 * (2 * 10 + 1) = 231$  et le résidu quantifié vaut  $r_\delta = Q_{10}(235 - 231) = 0$ . Pour un PAE  $\delta = 12$   $\tilde{Q}_\delta(p) = 10 * (2 * 12 + 1) = 250$ , et  $r_\delta = Q_{12}(235 - 250) = -1$ .

Ainsi, même si le PAE est plus important, un faible bruit peut persister si l'approximation de la basse résolution n'est vraiment pas bonne. Lorsque ce phénomène est trop fréquent pour un PAE donné, le PSNR risque de chuter et le débit d'augmenter. C'est ce qui se produit pour l'image IRM (VHP-Male pd-1 21) dans la FIG. 5.6.

Afin de corriger ce problème, une solution consisterait à transmettre une valeur supplémentaire indiquant un recentrage optimal à appliquer sur l'ensemble des pixels avant l'approximation presque sans perte (et à annuler après reconstruction). Ce paramètre serait relativement peu couteux puisque  $2\delta + 1$  valeurs seulement sont envisageables.

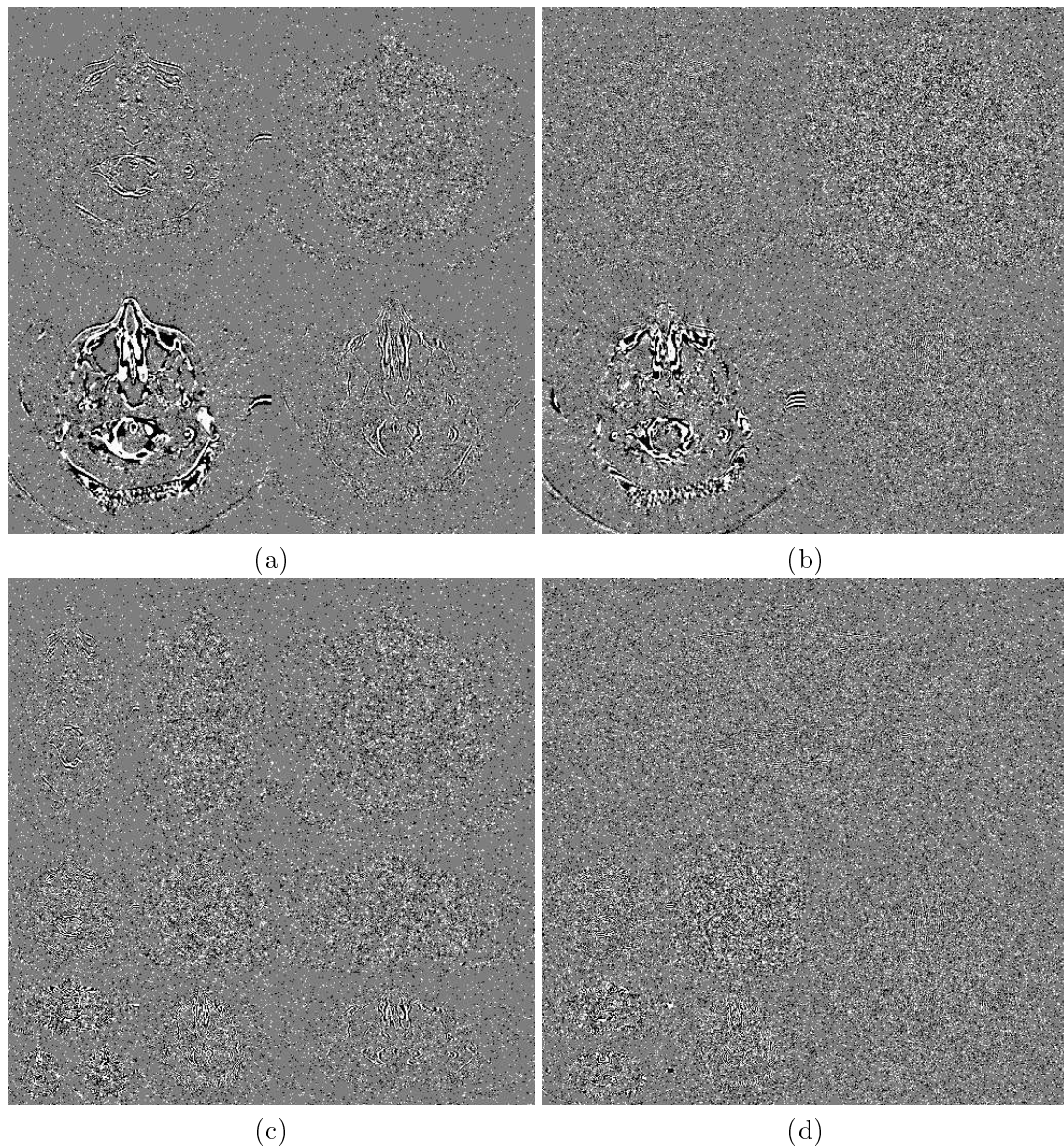


FIG. 5.5 – Problème de bruit induit par l'utilisation de la décomposition optimisée pour la compression sans perte, et appliquée en compression presque sans perte. Illustration sur une coupe de la décomposition volumique d'une CT de PMR pour un PAE de 4. Pour l'impression et l'affichage, les valeurs des résidus quantifiés sont saturées entre -1 et 1 (valeurs très majoritaires de l'amplitude du bruit résiduel) et ajustées entre 0 et 255. (a) 3-dy : 0.90 bpp, (b) L-dy : 1.04 bpp, (c) 3-OD : 1.03 bpp, (d) L-OD : 1.11 bpp (débits de la compression presque sans perte du volume complet).

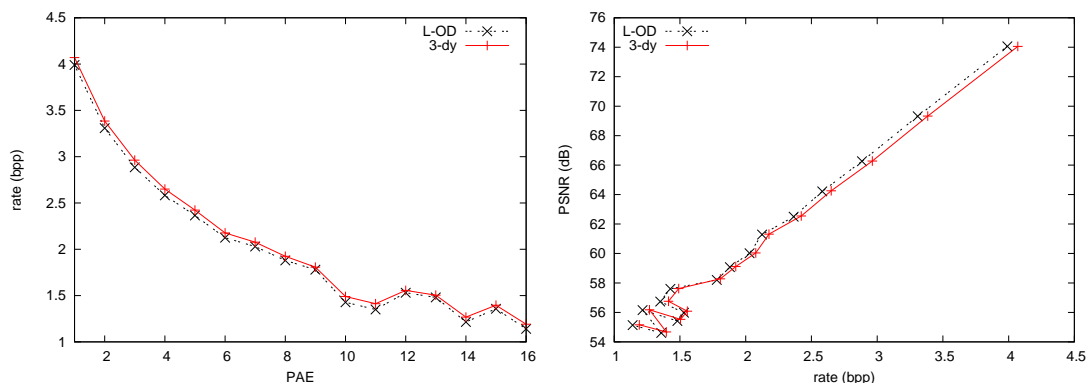


FIG. 5.6 – Illustration du problème pouvant survenir après une quantification inappropriée de la bande basse résolution pour la compression presque sans perte. Exemple de résultat obtenus sur l'image A.2-(1) pour des PAE allant de 1 à 16.

Dans l'état actuel du codec, l'image est d'abord centrée autour de sa valeur moyenne  $m$  (principalement pour réduire l'énergie et donc le débit lors du codage de l'image basse résolution dans le schéma ondelettes) et cette moyenne doit être transmise. Pour prendre en considération le problème susmentionné, et afin de trouver une bonne solution (même si elle n'est pas toujours optimale car elle ne considère pas la totalité de la décomposition) l'implémentation mise en place cherche la valeur de centrage  $m + s$  qui minimise l'erreur quadratique moyenne de l'approximation  $\tilde{Q}_\delta$  de l'image, et transmet  $m + s$  à la place de la moyenne  $m$ . Les résultats obtenus en utilisant cette méthode sont présentés dans la FIG. 5.7.

### 5.5.2 Compression d'une région d'intérêt uniquement

L'algorithme de compression présenté dans ce chapitre a également été étendu pour ne compresser que les informations contenues dans une région d'intérêt (ROI). La région est spécifiée par un masque binaire (sans aucune contrainte) de même dimension que l'image, qui est compressé à l'aide d'un codeur arithmétique binaire, contextuel et adaptatif et placé en début du flux de données.

La méthode SA-DWT [LL00] qui permet d'effectuer la transformée en ondelettes sur des données contraintes par un tel masque arbitraire est légèrement modifiée, de sorte à ce que les segments commençant à une position impaire ne soient pas translatés (seuls les pixels solitaires sont translatés afin d'être déplacés dans la composante basse résolution), car cette translation implique un déphasage sur les coefficients qui tend à perturber les décompositions transverses. Le lifting contraint par la ROI s'effectue segment par segment, et les valeurs extérieures à chacun d'entre eux sont estimées par une symétrisation conservant la parité de la position. Comme pour la SA-DWT, le résultat de la transformée permet toujours d'obtenir un nombre de coefficient non redondant (i.e. identique au nombre de pixels dans la ROI). Un masque, correspondant

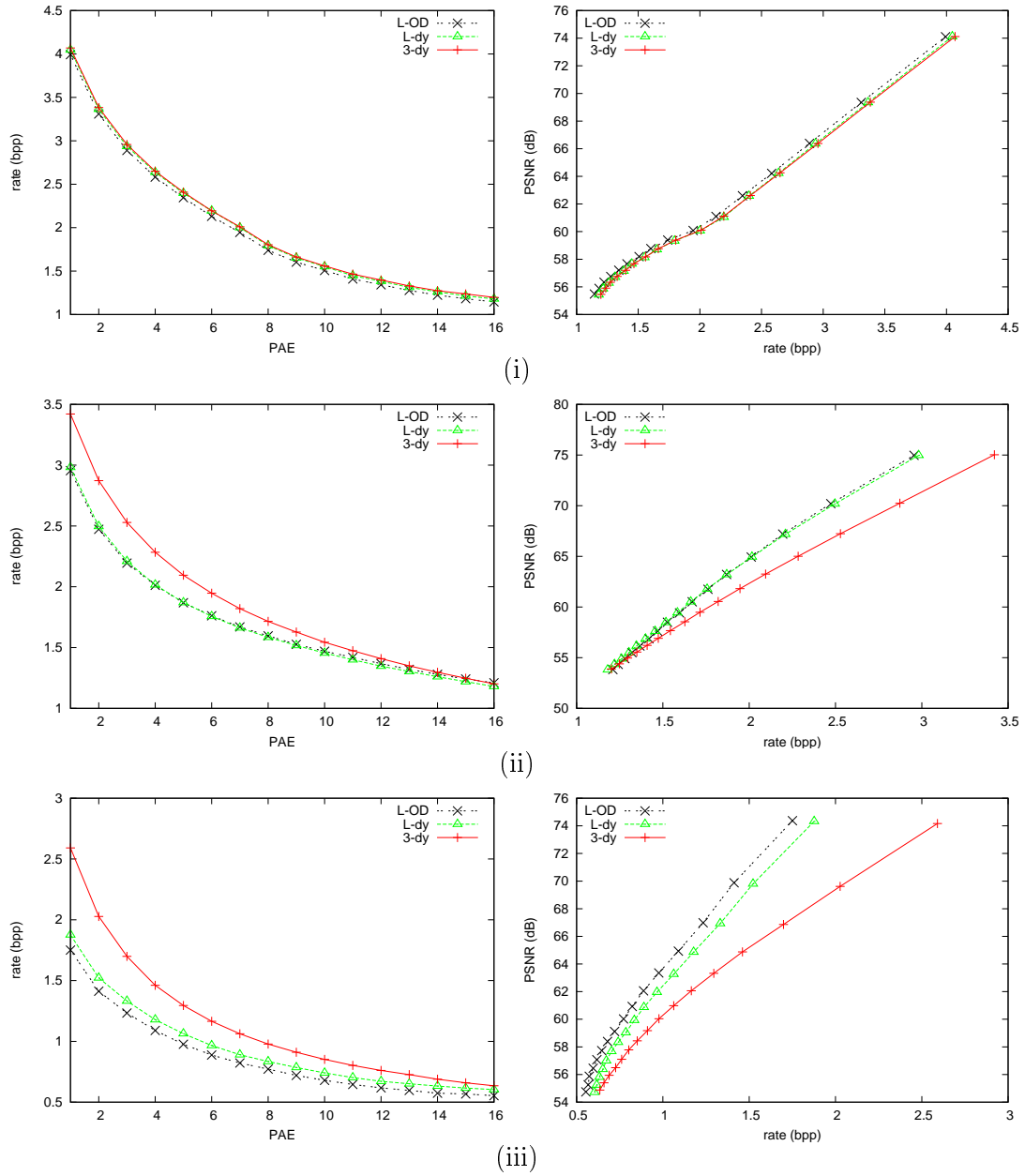


FIG. 5.7 – Résultats de la compression presque sans perte pour des PAE allant de 1 à 16 i) d'une IRM bruitée A.2-(l), ii) d'une image TDM douce A.1-(e), et iii) d'une IRM douce A.2-(p). La méthode d'optimisation de recentrage pour la quantification de l'image basse résolution a été appliquée.

à la ROI après transformation, est généré et permet de diriger par la suite le codage arithmétique par plan de bits, pour que celui-ci ne code que l'information utile.

L'optimisation de la décomposition s'effectue sur la ROI uniquement, en utilisant le lifting contraint par la région d'intérêt.

Quelques résultats de compression en utilisant cette approche sur les lames virtuelles sont présentés dans [TL11].

### 5.5.3 Utilisation d'ondelettes orientées

A l'origine, la technique d'optimisation présentée a été développée pour être utilisée avec un modèle de lifting 2D en ondelettes orientées, dérivé de l'ADL (*adaptive directional lifting*) [DWW<sup>+</sup>07, LN08], pour lequel les filtres de Lanczos permettaient de construire des filtres d'interpolation orientés. En effet, avec l'ADL, des ondelettes sont construites pour différentes orientations par le biais de deux filtres d'interpolation utilisés dans l'étape de lifting de prédiction. L'orientation était optimisée par blocs, et l'indice du filtre sélectionné était codé entropiquement en fonction des orientations voisines.

Les paramètres à optimiser ( $\Omega_n$ ) étaient alors deux longueurs de filtre Lanczos : une pour l'interpolation transversale permettant d'obtenir une précision sous-pixelique, et une pour le filtre de l'interpolation orientée (qui s'appuie sur les sous-pixels) ; la précision sous-pixelique à utiliser (i.e. le nombre d'orientations possibles) ; et enfin la carte des orientations à utiliser pour la sous-bande concernée (optimisée et compressée efficacement par blocs comme dans [WSJN07]). L'algorithme d'optimisation était similaire à celui présenté dans ce chapitre et cherchait à minimiser conjointement une estimation de l'entropie d'ordre zéro des coefficients et de la carte des orientations. Un tel schéma de compression permettait d'obtenir une meilleure entropie d'ordre zéro qu'avec la version non orientée de ce chapitre, mais les coefficients résultants des diverses orientations étaient alors moins bien compressés par le codeur entropique par plan de bits à cause d'une moins bonne entropie conditionnelle pour la modélisation contextuelle utilisée.

Ce travail mériterait d'être approfondi. Par exemple, l'optimisation pourrait prendre en considération un critère plus spécifique au codeur, et celui-ci pourrait certainement être amélioré afin d'être plus adapté. Enfin, dans un modèle de compression avec pertes, il pourrait permettre d'améliorer la qualité visuelle.

## Conclusion et perspectives

Un nouvel ensemble de bases de décomposition optimisées pour la compression sans perte a été présenté dans ce chapitre. Il s'appuie sur les paquets d'ondelettes non stationnaires et l'utilisation, dans un schéma de lifting, de familles de filtres particuliers tels que les filtres de Lanczos ou ceux de Deslauriers et Dubuc. Un processus d'optimisation, relativement rapide et simple, a été proposé afin de trouver une bonne base de décomposition pour la compression sans perte, et les résultats ont été analysés sur un ensemble d'images volumiques conséquent permettant de prendre en considération

les variabilités de contenus existants en tomographie et en IRM. L'adaptation de la longueur du support des filtres considérée est particulièrement bien adaptée pour l'optimisation, et permet d'améliorer la compression jusqu'à 29% en 2D et en 3D sur les images les plus douces/corrélées. Même si moins significative, l'optimisation de l'arbre de décomposition peut également permettre d'améliorer la compression, surtout sur les IRM et encore plus sur les IRM-3D. L'approche conserve des résultats équivalents à la décomposition traditionnelle en ondelettes 5/3 sur les images les plus bruitées.

Même si mathématiquement non optimaux en terme de moment nuls, les filtres de Lanczos se montrent aussi performants dans ce schéma de compression que les filtres optimaux de Deslauriers et Dubuc. Il a même pu être noté que l'optimisation sur les filtres de Lanczos tend à converger vers des longueurs ( $a$ ) plus faibles (et donc plus rapidement).

Des extensions de l'algorithme ont également été proposées pour la compression presque sans perte et la compression des données uniquement contenues dans une région d'intérêt. L'extension presque sans perte peut poser quelques problèmes de bruit résiduel perturbant la compression : elle nécessiterait un schéma d'optimisation plus complexe, dont les principes ont été présentés, prenant en considération le bruit sur la prédiction engendré par la quantification. Un problème similaire est engendré par la quantification de la représentation basse résolution. Une solution simple a été proposée, mais un modèle plus optimal pourrait être utilisé.

Comme autres perspectives, l'implémentation pourrait être étendue aux images couleurs ou multispectrales. L'algorithme d'optimisation devrait être revu avec un critère de compacité d'énergie ou de variation totale pour la compression avec pertes ou le débruitage. Pour une compression avec pertes il serait également opportun d'utiliser un codeur mieux adapté tel que EBCOT (JPEG-2000) par exemple. Pour de grosses images, la décomposition adaptative pourrait être optimisée par macro-blocs (comme les *groups of pictures* en vidéo, ou les *tiles* de JPEG-2000 par exemple), éventuellement de tailles variables dans l'espace, afin de mieux s'adapter aux particularités plus locales : dans les longues séquences tomographiques (corps complets de VHP-Male et VHP-Female par exemple), ou encore sur les lames virtuelles ou les images satellitaires. Il pourrait être intéressant de voir l'apport permis par l'utilisation d'autres familles de filtres plus spécifiques, ou une optimisation préalable de familles d'ondelettes en fonction de la modalité ciblée.



## Chapitre 6

# Compression scalable exploitant un apprentissage multi-bases, application aux lames virtuelles

### Introduction

Dans ce chapitre, on cherche à pousser plus loin la progressivité en qualité afin de fournir une représentation efficace pour la consultation d'images à distance et/ou pour leur compression avec pertes. Pour cela on s'intéresse notamment à l'optimisation de la représentation pour une classe d'images particulière. Les systèmes d'archivages ne stockent généralement que quelques modalités d'images pour lesquelles on peut chercher à être le plus optimal possible en exploitant l'apprentissage. Ce modèle de compression est ici principalement développé pour les lames virtuelles qui ont un type de contenu texturé bien particulier (principalement des tissus cellulaires). Cependant, il pourrait très bien s'appliquer à d'autres classes d'images. En partant du principe que le client consulte un nombre restreint de modalités d'images, les informations relatives à l'apprentissage sont supposées être préalablement stockées dans son ordinateur.

Dans ce cadre applicatif, une modélisation est fournie pour une progressivité en qualité efficace en régime irréversible, dans un premier temps, puis ce modèle est adapté pour permettre d'effectuer une compression réversible et donc une progressivité jusqu'au sans perte dans un second temps. Le chapitre s'organise en quatre sections. La première (6.1) rappelle comment construire une base optimale pour la décorrélation, à savoir la KLT, et les principales propriétés de cette base. La seconde section (6.2) motive tout d'abord l'utilisation d'un apprentissage multi KLT ( $K$ -KLT). L'algorithme d'apprentissage mis en place est ensuite décrit ainsi qu'une approche pour son implémentation supportant un très grand nombre de vecteurs d'apprentissage. La troisième section (6.3) s'intéresse à l'application de cet apprentissage dans un schéma de compression d'images. L'approche de compression, dérivée de SPIHT, est mise en place pour divers cadres d'utilisation des  $K$ -KLT : dans le domaine image directement, ou lors-



qu'employées comme post-transformées. Cette section se clôt sur une comparaison de divers résultats. L'avant dernière section (6.4) présente une solution pour le passage à une compression sans perte, les problèmes engendrés par cette solution, et les résultats obtenus. Enfin, la dernière section (6.5) propose des améliorations des codecs afin qu'ils puissent être comparés aux références usuelles, et s'achève sur leurs résultats.

Les principales contributions de ce chapitre sont i) la méthode d'optimisation des  $K$ -KLT et en particulier le critère utilisé pour prendre en considération l'ensemble des coefficient transformés, ii) les schémas d'utilisation des  $K$ -KLT pour permettre de décorrélérer les coefficients d'une transformée en ondelettes, iii) les extensions de SPIHT mises en place pour la compression progressive d'images et leur optimisation exploitant le codage arithmétique.

## 6.1 Base optimale pour la décorrélation

### 6.1.1 KLT, PCA, POD, SVD

Ce chapitre s'appuie sur la KLT (*Karhunen-Loeve transform*) également connue sous le sigle de PCA (*principal component analysis* ou encore POD (*proper orthogonal decomposition*)) selon le domaine d'application. Il existe de nombreuses publications et tutoriels autour de cette transformée tels que [Cha00] ou [Don00] par exemple.

Soit  $A$  une matrice d'apprentissage de taille  $D \times N$  dont les  $N$  vecteurs colonnes sont des réalisations d'une variable aléatoire  $X$  de dimension  $D$ . Dans ce chapitre on suppose que  $X$  est centrée autour du vecteur nul. Cette hypothèse évite de devoir conserver un vecteur moyen  $\mu_X$  et est intéressante lorsque les données de  $A$ , servant à la modélisation, ne sont pas spécialement représentatives des variations d'intensité moyennes pouvant survenir sur d'autres ensembles de données. En compression d'images elle a relativement peu d'incidence sur les résultats, mais dans le cas où ce vecteur moyen est vraiment représentatif, celui-ci peut permettre de réduire en partie l'énergie des coefficients à coder, et donc diminuer légèrement l'entropie.

L'estimation de la KLT peut se faire à l'aide de la SVD (*Singular Value Decomposition*) de  $A$ . La base  $\kappa$  formée par les vecteurs  $\kappa_i$  associés aux  $D$  valeurs singulières triées dans l'ordre décroissant est la même que celle obtenue pour la SVD de la matrice de covariance  $AA^T$ , et qui correspond à sa décomposition en valeurs propres (variance des coefficients transformés) et vecteurs propres.

### 6.1.2 Propriétés de la KLT

#### Propriété 6.1.1 : OPTIMALITÉ POUR LA DÉCORRÉLATION

Cette base orthonormale est statistiquement optimale pour décorrélérer l'information contenue dans les vecteurs de  $A$  : la matrice de covariance des données transformées

$A^\top \kappa$  est une matrice diagonale (valeurs propres triées de la matrice de covariance de  $A$ ) et donc aucune covariance ne subsiste dans les données transformées. Si la matrice  $A$  est suffisamment représentative pour permettre une modélisation correcte des statistiques de  $X$ , et donc celles des données sur lesquelles on souhaite appliquer la KLT,  $\kappa$  sera optimale. Dans la suite de ce chapitre, on considère qu'elle l'est.

Soit  $\Phi$  l'ensemble des bases orthonormales de dimension  $D$ .

**Propriété 6.1.2 : MINIMISATION DE LA MSE POUR  $d \leq D$  COEFFICIENTS**

La KLT est la transformée orthonormale qui minimise l'espérance de l'erreur quadratique moyenne sur  $X$  lorsqu'on ne conserve que les  $d \leq D$  premiers coefficients transformés (associés aux  $d$  premiers vecteurs principaux) [TM01]-sec. 4.1.2 :

$$E \left[ \left\| X - \sum_{i=1}^{d \leq D} \langle X, \kappa_i \rangle \kappa_i \right\|^2 \right] \leq E \left[ \left\| X - \sum_{i=1}^{d \leq D} \langle X, \phi_i \rangle \phi_i \right\|^2 \right] \quad \forall \phi \in \Phi. \quad (6.1)$$

**Propriété 6.1.3 : MAXIMISATION DE L'ÉNERGIE POUR  $d \leq D$  COEFFICIENTS**

Sachant que les bases orthonormées préservent l'énergie dans l'espace transformé :

$$\|X\|^2 = \sum_{i=1}^D \langle X, \kappa_i \rangle^2 = \sum_{i=1}^D \langle X, \phi_i \rangle^2 \quad \forall \phi \in \Phi, \quad (6.2)$$

la KLT maximise également l'énergie contenue dans les  $d$  premiers coefficients :

$$E \left[ \sum_{i=1}^d \langle X, \kappa_i \rangle^2 \right] \geq E \left[ \sum_{i=1}^d \langle X, \phi_i \rangle^2 \right] \quad \forall \phi \in \Phi, \forall d \leq D, \quad (6.3)$$

et donc, minimise l'énergie contenue dans les derniers coefficients :

$$E \left[ \sum_{i=d+1}^D \langle X, \kappa_i \rangle^2 \right] \leq E \left[ \sum_{i=d+1}^D \langle X, \phi_i \rangle^2 \right] \quad \forall \phi \in \Phi, \forall d \leq D, \quad (6.4)$$

**Propriété 6.1.4 : MINIMISATION D'UN CRITÈRE DE COMPACITÉ D'ÉNERGIE**

On peut montrer, en sommant (6.4) pour  $d = 1..(D-1)$  que la KLT minimise aussi :

$$\frac{E \left[ \sum_{i=1}^D (i-1) \langle X, \kappa_i \rangle^2 \right]}{(D-1)E [\|X\|^2]} \leq \frac{E \left[ \sum_{i=1}^D (i-1) \langle X, \phi_i \rangle^2 \right]}{(D-1)E [\|X\|^2]} \quad \forall \phi \in \Phi. \quad (6.5)$$

Ce critère de compacité de l'énergie (compris entre 0 et 1) prend en compte l'ensemble

des coefficients et sera exploité par la suite dans l'algorithme d'apprentissage. On peut rapprocher cette écriture de celle du centre de gravité des coefficients. Cependant l'espérance de ce dernier ne serait pas spécialement minimisée à cause d'un terme de covariance avec  $1/\|X\|^2$  qui apparaîtrait.

### Propriété 6.1.5 : OPTIMALITÉ DU GAIN DE CODAGE

Enfin, la KLT est également optimale en terme de gain [TM01]-sec. 4.3.2 pour un codage indépendant des coefficients lorsqu'ils sont quantifiés uniformément. Bien que l'hypothèse d'indépendance pour le codage des coefficients soit rarement vérifiée, cette propriété rend la KLT très intéressante pour une transmission progressive par plan de bits.

## 6.2 Apprentissage multi-bases

### 6.2.1 Motivations

Les propriétés de la KLT la rendent forcément très attrayante pour la compression. Cependant pour que la décorrélation soit vraiment optimale pour une image donnée, la base  $\kappa$  doit être construite sur les données de cette image particulière, et donc transmise avec elle. Le coût de cette transmission dépend alors du  $D$  utilisé. Le plus souvent,  $X$  est modélisée par des blocs de pixels de l'image de tailles  $\sqrt{D} \times \sqrt{D}$ , et  $D$  est choisi pour qu'il subsiste peu de corrélation entre les blocs contigus (une taille  $8 \times 8$  est assez courante). L'apport de l'utilisation d'une KLT optimale ne compense alors généralement pas l'information supplémentaire requise par  $\kappa$ , surtout lorsque le codage se fait avec pertes.

La KLT sera donc plutôt « apprise » pour une classe d'images particulières, et dans ce cas  $\kappa$  sera fixée au codeur et au décodeur. La construction de  $\kappa$  se fera alors sur une matrice  $A$  représentative, construite à l'aide de vecteurs extraits d'images d'apprentissage.

Si les images ont des propriétés bien spécifiques, la KLT améliorera alors très certainement le rapport débit/distorsion en comparaison à une DCT classique. Cependant, pour des images possédant diverses particularités locales (contours orientés, textures, ...), le fait d'utiliser une base unique pour la totalité de l'image n'est probablement pas la meilleure solution.

Les décompositions parcimonieuses permettent de répondre à ce besoin d'adaptation aux caractéristiques locales. Elles utilisent un ensemble redondant de vecteurs de projection dont seuls les plus pertinents sont retenus afin de représenter efficacement un vecteur (bloc) de pixels particulier. Des algorithmes d'apprentissage performants tels que  $K$ -SVD [AEB06] ou ITD (*Iteration-Tuned Dictionaries*) [JZK10a, JZK10b, JZK10c] et leur application à la compression d'image ont récemment été proposés. Le problème de la décomposition parcimonieuse est qu'elle rend nécessaire la transmission pour chaque bloc des index des vecteurs retenus pour la décomposition. De plus ces représentations

n'utilisent pas une base orthonormée (le nombre de vecteurs de projection retenus est préférablement bien inférieur à  $D$  et ils ne sont pas spécialement orthogonaux entre eux), ce qui rend les possibilités pour effectuer une compression (éventuellement progressive jusqu'au) sans perte, moins pratiques et efficaces.

Les images anatomopathologiques qui ont des structures très répétitives semblent être des bonnes candidates pour une approche par apprentissage. Cependant, ces images étant employées en milieu hospitalier, il est souhaitable de conserver l'opportunité d'effectuer une compression pouvant aller jusqu'au sans perte. Il a donc été choisi d'effectuer un apprentissage multi-bases orthonormées, en sachant les bases orthonormées peuvent être approximées par lifting pour permettre une compression réversible (c.f. 6.4).

### 6.2.2 Algorithme d'apprentissage : construction itérative des KLT

D'après la Propriété 6.1.4, minimiser  $E \left[ \sum_{i=1}^D (i-1) \langle X, \phi_i \rangle^2 \right]$ , qui peut servir d'indicateur de compacité de l'énergie, permet d'obtenir la KLT. Lorsqu'on ne considère qu'une seule classe de réalisations de  $X$ , la KLT est la décomposition optimale, en terme de MSE, pour un nombre restreint de coefficients (Propriété 6.1.2) ou pour une quantification uniforme (Propriété 6.1.5), et peut être utilisée dans un modèle de compression *lossy-to-lossless*, lorsqu'elle est approximée à l'aide d'une décomposition en étapes de lifting telle que la factorisation matricielle PLUS [HS01, HS03, Hao04].

Afin de combiner l'apprentissage, faisant l'intérêt des représentations parcimonieuses, à un modèle de bases orthonormées, il est envisagé de découper l'espace des réalisations de  $X$  en  $K$  classes de réalisations  $X^{[k]}$  qui auront chacune leur spécificités. L'apprentissage est optimisé à l'aide d'un algorithme dont le système de convergence est de type K-means/EM, et le critère à minimiser est :

$$Q \left( C, (\kappa)_{k \in [1..K]} \right) = \frac{E \left[ \sum_{i=1}^D (i-1) \langle X, (\kappa_i)_{C(X)} \rangle^2 \right]}{(D-1)E [\|X\|^2]} . \quad (6.6)$$

Le paramètre  $C$  est une fonction qui permet la classification des réalisations de  $X$ , et  $(\kappa)_{k \in [1..K]}$  est la famille des  $K$  bases associées à chacune des classes.

Très classiquement, à chaque itération  $j$ ,

(Étape 1)  $Q$  est minimisé pour  $C$  fixé à  $C^{[[j-1]]}$  (c.f. étape 2) afin d'obtenir une famille de bases  $(\kappa)_{k \in [1..K]}^{[[j]]}$  optimale. Cette étape consiste simplement à calculer la KLT pour chacune des classes.

(Étape 2) Pour  $(\kappa)_{k \in [1..K]}^{[[j]]}$  fixé, la classification optimale  $C^{[[j]]}$  est cherchée à son tour. Afin de minimiser  $Q$ , celle-ci correspond à

$$C^{[[j]]}(X) = \arg \min_k \sum_{i=1}^D (i-1) \langle X, (\kappa_i)_k^{[[j]]} \rangle^2 .$$

On peut facilement voir que :

$$Q\left(C^{[[j-1]]}, (\kappa)_{k \in [1..K]}^{[[j]]}\right) \leq Q\left(C^{[[j-1]]}, (\kappa)_{k \in [1..K]}^{[[j-1]]}\right), \quad (6.7)$$

et que :

$$Q\left(C^{[[j]]}, (\kappa)_{k \in [1..K]}^{[[j]]}\right) \leq Q\left(C^{[[j-1]]}, (\kappa)_{k \in [1..K]}^{[[j]]}\right). \quad (6.8)$$

Ainsi, cette approche est assurée de converger, et le critère  $Q$  obtenu pour chaque étape de chaque itération sera décroissant.

Initialisation : Même si ce n'est pas nécessairement optimal, l'algorithme s'initialise en utilisant  $C^{[[0]]}$  comme associant aléatoirement une classe à chacun des vecteurs d'apprentissage. Ainsi, les bases calculées à l'étape 1 de la première itération seront assez similaires, mais cela n'empêche pas la convergence.

L'algorithme obtenu n'atteindra généralement pas un optimum global, mais cette approche assez simple est suffisante pour pouvoir évaluer l'impact sur la compression de l'apprentissage multi-bases. L'apprentissage n'étant pas le point principal de ce chapitre nous n'entrerons pas plus dans les détails. Nous avons pu trouver que des approches similaires avaient déjà été proposées dans la littérature : dans [YSM10] un algorithme MAP-EM est utilisé pour l'apprentissage dans le cadre de traitements d'images (débruitage, super-résolution, ...); et dans [LZ05b, LZ05a, LLZ07] l'algorithme e-LBG (*extended LBG*) pour construire des k-PCA, bien que peu détaillé, semble suivre la même technique que celle présentée mais utilise un critère de fidélité (erreur de reconstruction) basé sur un nombre restreint de vecteurs pour une compression avec pertes. D'après les propriétés 6.1.2 et 6.1.4, l'optimisation devrait alors donner des résultats équivalents (du moins sur les vecteurs principaux retenus dans la k-PCA).

### 6.2.3 Note concernant l'implémentation

Même s'il existe d'autres méthodes pour calculer la KLT (qui peuvent permettre de réduire la complexité calculatoire), l'approche retenue s'appuie sur le calcul des vecteurs propres de la matrice de covariance. Celle-ci a l'avantage de pouvoir être calculée, itérativement sur les vecteurs d'apprentissage, dans un espace mémoire en  $O(D^2)$ . Le fait que le calcul soit itératif permet d'éviter de stocker en mémoire de grosses matrices (tous les vecteurs d'apprentissage), et facilite le traitement multi-classes. En effet, au lieu de devoir attribuer une classe à l'ensemble des vecteurs afin de construire une matrice d'apprentissage pour chaque classe (et ce, à chaque itération de l'algorithme de minimisation), les vecteurs peuvent être traités séquentiellement (à partir d'un disque dur et d'une petite mémoire tampon pour réduire les accès disque) : le vecteur à traiter est classé, et permet une itération supplémentaire pour le calcul de la matrice de covariance de la classe concernée.

Le calcul itératif d'une matrice de covariance est effectué à l'aide de l'accumulation  $\chi_i = \chi_{i-1} + x_i x_i^\top$  (des calculs et de la mémoire sont économisés en sachant que  $\chi_i$  est symétrique) et si les  $x_i$  ne sont pas considérés comme centrés, on peut également

effectuer l'accumulation  $\varsigma_i = \varsigma_{i-1} + x_i$ . La matrice de covariance est alors estimée par  $\text{cov}(X) = \frac{1}{N}\chi_N$  (la division par  $N$  n'est pas nécessaire pour le calcul des vecteurs propres), et si  $X$  n'est pas considérée comme centrée, elle peut être estimée par  $\text{cov}(X) = \frac{1}{N}(\chi_N - \varsigma_N \varsigma_N^\top / N)$  (simple application de la formule  $\text{cov}(A, B) = E(AB) - E(A)E(B)$ )

Le problème principal de cette approche concerne la précision des calculs. En effet, pour une représentation en virgule flottante (mantisse/exposant), l'accumulation sur un grand nombre d'itérations peut peu à peu faire croître l'exposant (lorsque les valeurs sont strictement positives, par exemple, comme c'est souvent le cas pour les valeurs des pixels en image). Dans ce cas, il se produit une perte de précision lorsqu'une nouvelle valeur doit être ajoutée dans l'accumulateur, son exposant étant plus faible. Dans un cas extrême, si la différence d'exposant est trop importante, il se peut même que l'accumulateur ne subisse plus aucune modification, même si  $N$  continue de croître.

Afin de permettre de limiter ce problème, l'implémentation réalisée utilise une chaîne d'accumulateurs (créés quand nécessaires) dont le nombre d'accumulations possibles est limité à  $\Delta$ . Si un accumulateur vient à atteindre un nombre d'accumulation égale à  $\Delta$ , son contenu est accumulé dans le suivant et il est réinitialisé. Ainsi après l'utilisation de  $N$  vecteurs  $\lfloor \log_\Delta(N) \rfloor + 1$  accumulateurs seront utilisés. L'accumulation finale est calculée en sommant le contenu de tous les accumulateurs intermédiaires, en commençant par le premier. Même si cette approche peut être sous optimale en terme de précision, elle est certainement plus rapide qu'un système optimal qui simulerait une représentation en précision infinie, et ne nécessite pas trop de mémoire.

### 6.3 Application en compression d'image dans un schéma scalable en qualité

Plusieurs approches vont être proposées pour l'utilisation des  $K$ -KLT qui viennent d'être présentées. Afin d'évaluer les différentes représentations en compression progressive d'image, le codage des vecteurs transformés s'appuie sur un algorithme de codage par plan de bits, dérivé de SPIHT [SP96b], sans aucun codage arithmétique (qui pourrait biaiser la comparaison des résultats). Ainsi, les courbes débit/distorsion obtenues pour chacune des techniques seront relativement comparables entre elles, et également avec la compression d'une décomposition ondelettes uniquement. Cependant, rien n'empêchera par la suite d'optimiser le codage afin d'améliorer les résultats pour une représentation particulière.

Dans cette section, le fonctionnement général de l'algorithme SPIHT va être rappelé et il sera proposé un modèle pour l'intégration du codage d'un vecteur transformé par KLT. Après quoi, plusieurs schémas d'application des  $K$ -KLT seront décrits : une méthode pour leur utilisation dans le domaine image directement, et des approches pour leur utilisation dans le domaine transformé, après une décomposition en ondelettes.

### 6.3.1 Algorithme SPIHT

L'algorithme SPIHT [SP96b] s'appuie sur la compression de trois listes : i) une liste de pixels/coefficients non significatifs (LIP), ii) une liste d'ensembles (typés) englobants des pixels non significatifs (LIS), et iii) une liste des pixels déjà significatifs. Par significatif est entendu que la valeur absolue du coefficient (ou d'un des coefficients dans le cas d'ensembles) est supérieure à zéro, lorsqu'elle est quantifiée par un pas correspondant au plan de bit actuel.

Pour le codage itératif des plan de bits, les listes sont parcourues à tour de rôle :

1. Pour chacun des pixels de la LIP, un bit est transmis à 1 ou 0 pour indiquer si le coefficient devient significatif ou non (les coefficients placés dans la LIP ont généralement de fortes chances de le devenir). S'il le devient, un autre bit est utilisé pour transmettre le signe du coefficient, et le coefficient est déplacé à la fin de la LSP.
2. Ensuite, pour chacun des ensembles contenus dans la LIS (y compris ceux qui y seront ajoutés durant cette étape), un bit est transmis à 1 ou 0 pour indiquer s'il devient significatif ou non. S'il le devient, des traitements spécifiques au type de l'ensemble sont effectués. Ces ensembles seront modélisés différemment selon la décomposition pour laquelle doit être adapté SPIHT et seront détaillés dans la description de chacune des représentations proposées.

Dans l'algorithme SPIHT original, deux types d'ensembles peuvent être distingués : i) les macro-ensembles MaSET qui peuvent être schématisés comme des ensembles d'ensembles, et ii) les ensembles mixtes MiSET qui peuvent contenir des coefficients et des ensembles. Dans [SP96b] les ensembles de type *A* sont des MiSET, et les ensembles de type *B* sont des MaSET.

Si un MaSET devient significatif, il est retiré de la LIS et les sous-ensembles qu'il contient sont ajoutés à la fin de celle-ci. Si un MiSET devient significatif, la signifiante des coefficients qu'il contient est transmise. Pour toutes les valeurs significatives, le signe est également transmis et le coefficient est placé dans la LSP. Les autres sont placés dans la LIP. Le MiSET est alors retiré de la LIS et les sous-ensembles qu'il contient sont ajoutés à la fin de celle-ci et immédiatement traités.

Pour le codage d'un vecteur  $x$  transformé par KLT, on ajoute un nouveau type d'ensemble : le  $K_i$ SET. Celui-ci regroupe les coefficients  $y_j = \langle x, \kappa_j \rangle$  pour  $j = i..D$ . Lorsqu'un  $K_i$ SET devient significatif, la signifiante des coefficients  $i+s < D$  est transmise, pour  $s = 0$  jusqu'à  $s = S$  l'indice du premier coefficient significatif, ainsi que le signe de  $y_{i+S}$ . Les coefficients non significatifs  $y_{\{i+s < i+S\}}$ , qui ont une forte probabilité de devenir significatifs aux plans de bits suivants, sont ajoutés à la LIP, et  $y_{i+S}$  est ajouté à la LSP. Le  $K_i$ SET est alors retiré de la LIS, et, si  $i + S + 1 \leq D$ , un nouveau  $K_{i+S+1}$ SET, contenant les derniers coefficients, est ajouté à la fin de celle-ci.

3. Enfin, pour tous les coefficients de la LSP qui n'ont pas été ajoutés durant cette itération, le bit de raffinement de leur valeur est transmis.

### 6.3.2 Application dans le domaine direct (ADD)

L'application des  $K$ -KLT dans le domaine direct s'effectue selon le schéma classique de décorrélation par blocs de pixels. Comme pour la DCT par blocs de JPEG ou pour les représentations parcimonieuses, qui, généralement, s'appliquent également dans le domaine direct, des artefacts de type bloc apparaissent et viennent perturber l'appréciation du contenu à bas débits (un exemple est disponible en FIG. 6.15). Ce problème est souvent reproché à ce type d'approche.

L'algorithme pour la compression avec les  $K$ -KLT dans le domaine direct n'est pas décrit ici : il correspond à celui utilisé pour l'approche intra-bande dans le domaine ondelette (c.f. 6.3.3.2), dans le cas où la transformée en ondelettes est effectuée sur 0 niveau de décomposition (i.e. elle n'est pas appliquée). L'algorithme est donc équivalent à celui pour la compression d'une unique bande basse résolution.

Des exemples de résultats d'apprentissage sont donnés dans la FIG. 6.1 et la FIG. 6.2.

### 6.3.3 Application après une transformée en ondelettes

Afin de contourner le problème d'artefacts de blocs, il a été envisagé d'utiliser les  $K$ -KLT dans le domaine ondelettes, qui devrait également permettre de mieux prendre en compte le voisinage lors de la décorrélation (grâce au support des ondelettes). Les corrélations des coefficients et donc l'apprentissage sont spécifiques à la décomposition en ondelettes utilisée. Celle-ci doit donc être préalablement figée pour la classe d'images à laquelle on souhaite l'appliquer, contrairement au chapitre précédent où la décomposition est optimisée pour chaque image. Pour simplifier les expérimentations, et afin que le codage par plan de bits soit optimal, les bases d'ondelettes choisies sont orthonormées [Dau88, Dau92, Dau93, CD93]. Les *symlet* de Daubechies [Dau93] cherchent à maximiser la symétrie des ondelettes orthonormales à support compact en prenant celles qui sont les plus proches d'un filtre à phase linéaire, pour une taille de support donné. Elles ont été préférées afin d'éviter de générer des artefacts désagréables en régime avec pertes, liés au déphasage.

#### 6.3.3.1 Approche globale (AG)

L'idée derrière l'approche globale (AG) est de combiner les  $K$ -KLT aux ondelettes afin de produire des transformées orthonormées optimisées se chevauchant et qui pourraient être vues comme des LOT : *lapped orthogonal transforms* (c.f. [TM01]-sec. 4.2.4 ou [dQ00] par exemple). Pour ce faire, on choisit d'appliquer une transformée en ondelettes, et de décorréler un ensemble de coefficients mis en relation par leur position spatiale (arbre hiérarchique FIG. 6.3) et extrait sous forme de bloc hiérarchique.

Pour le codage de cette représentation, comme la majorité de l'information se situe dans le premier coefficient des KLT, la LIP est initialisée avec le premier coefficient ( $< x, (\kappa_1)_{C(x)} >$ ) de l'ensemble des blocs, et la LIS avec leur  $K_2$ SET respectif. L'algorithme SPIHT est ensuite lancé. Afin d'améliorer légèrement la scalabilité en qualité, on



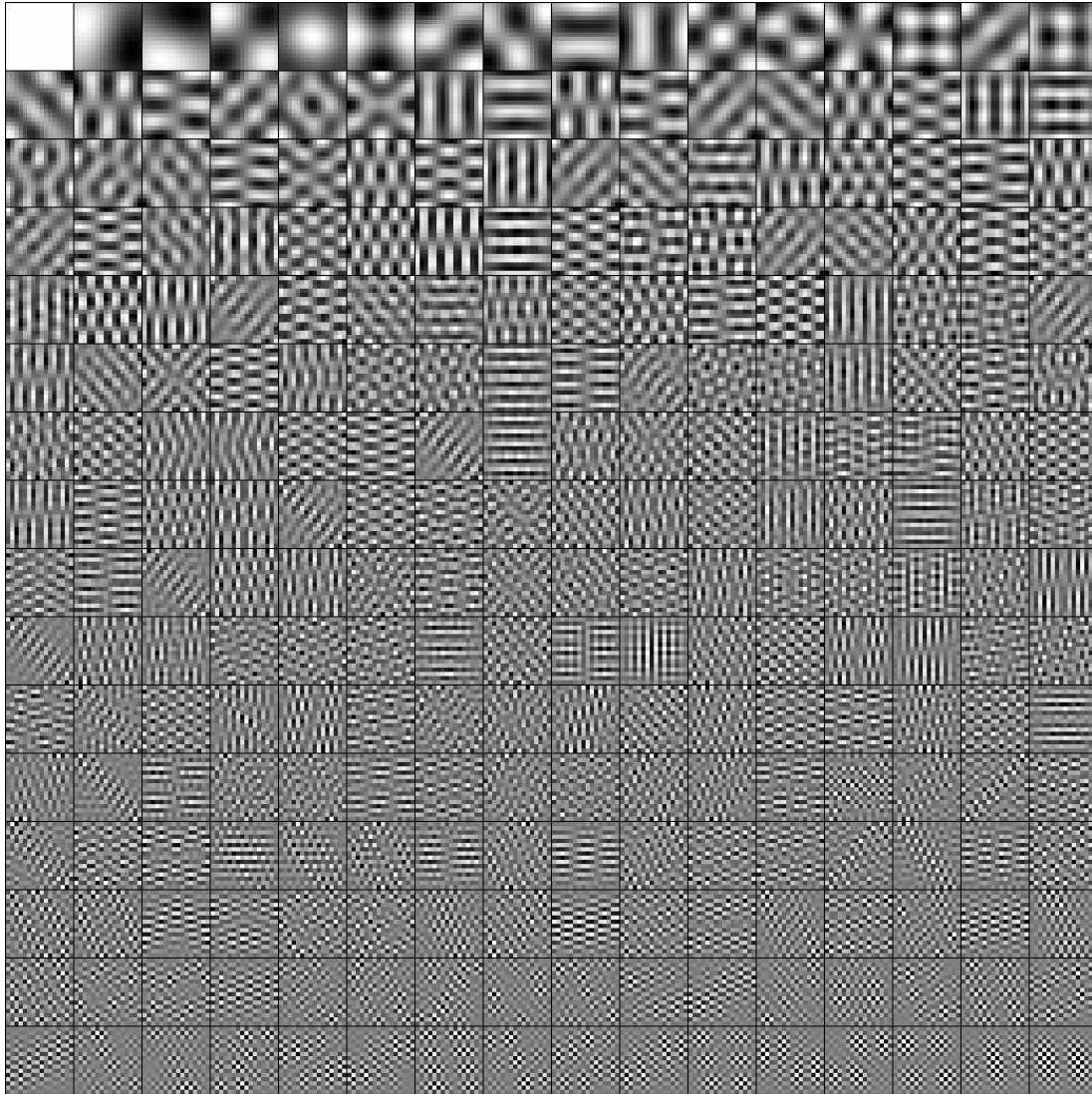


FIG. 6.1 – Domaine direct : vecteurs (ordonnés ligne par ligne, colonne par colonne) de la base obtenue pour un apprentissage mono KLT (sur la lame virtuelle 01 A.3) sur des blocs  $16 \times 16$ .

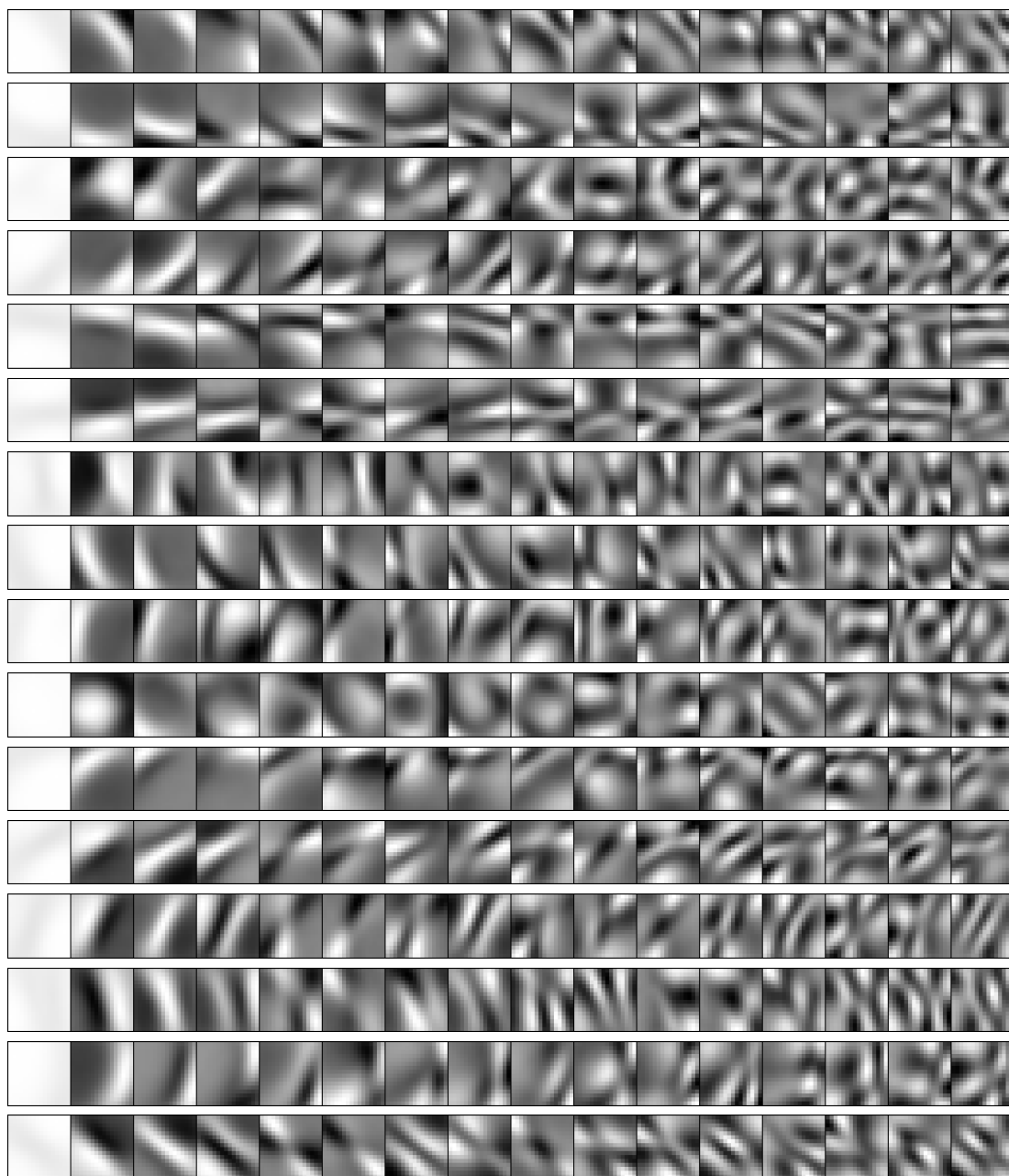


FIG. 6.2 – Domaine direct : 16 premiers vecteurs (ordonnés de gauche à droite) de chaque base obtenue pour un apprentissage 16-KLT (sur la lame virtuelle 01 A.3) sur des blocs  $16 \times 16$  obtenus sur une décomposition dyadique sur 3 niveaux.

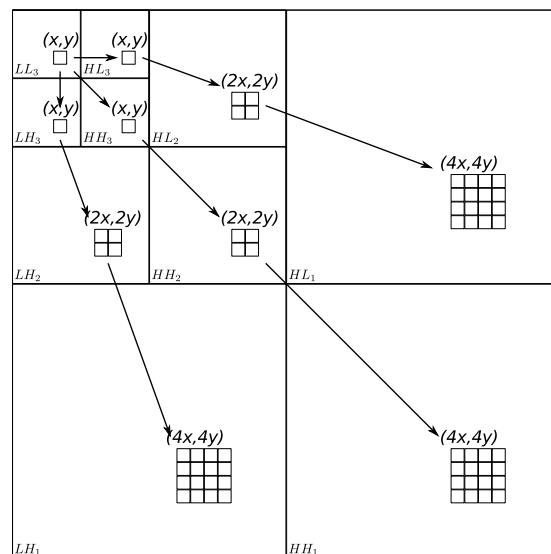


FIG. 6.3 – Arbre hiérarchique des coefficients d'ondelettes mis en relation par leur position spatiale pour une décomposition dyadique, pouvant permettre de construire un bloc de coefficients hiérarchiques (ici  $8 \times 8$ ).

considère (à bas débit) que le premier vecteur propre des KLT correspond à la composante continue de la bande basse résolution (ce qui est très proche de la réalité sur les résultats obtenus par apprentissage), et l'indice de la classe de chaque bloc n'est alors transmis que lorsque l'un des coefficients suivants (i.e. le  $K_2SET$ ) devient significatif. Ceci évite de devoir transmettre trop d'information avant le codage des premiers coefficients. A titre indicatif, avec 64 classes pour des blocs  $8 \times 8$  cette information nécessite un tout petit peu moins de 0.1 bit par pixel, et l'approche utilisée permet alors de gagner plusieurs dB sur le PSNR obtenu pour des débits proches.

Pour être un peu plus efficace à bas débit, les  $K_2SET$  sont regroupés par blocs de  $2 \times 2$  en  $MaSET$ , ce qui est assez similaire aux regroupements que fait SPIHT à l'aide de ses ensembles de type A.

Des exemples de résultats d'apprentissage sont donnés dans la FIG. 6.4 et la FIG. 6.5.

### 6.3.3.2 Approche intra sous-bande (AISB)

Cette seconde approche s'appuie sur un apprentissage  $K$ -KLT spécifique à chacune des sous-bandes de la décomposition en ondelettes. Pour une décomposition dyadique sur  $L$  niveaux cela représente donc  $3L + 1$  apprentissages. Un découpage en blocs de chaque bande est utilisé afin de constituer les vecteurs à transformer par KLT. Cette approche est plus souple que la précédente, car la taille des blocs utilisée n'est pas dépendante du nombre de niveaux de la décomposition en ondelettes. Dans le cas où le nombre de classe et la taille des blocs sont les mêmes que pour l'approche globale, le coût de codage de l'indice de la classe ainsi que la complexité calculatoire lors de la

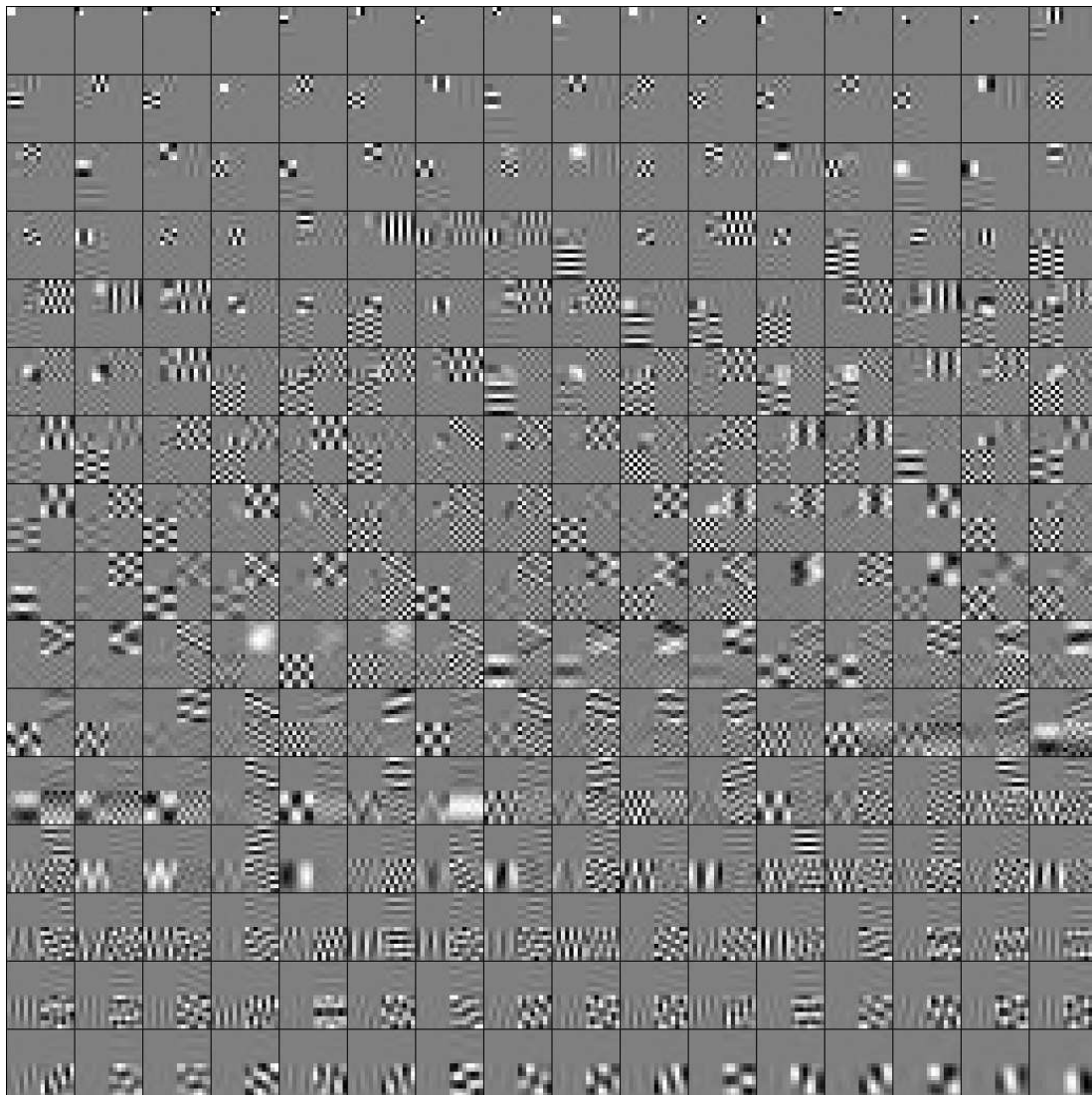


FIG. 6.4 – Approche globale : vecteurs (ordonnés ligne par ligne, colonne par colonne) de la base obtenue pour un apprentissage mono KLT (sur la lame virtuelle 01 A.3) sur des blocs  $16 \times 16$  obtenus sur une décomposition dyadique sur 3 niveaux.

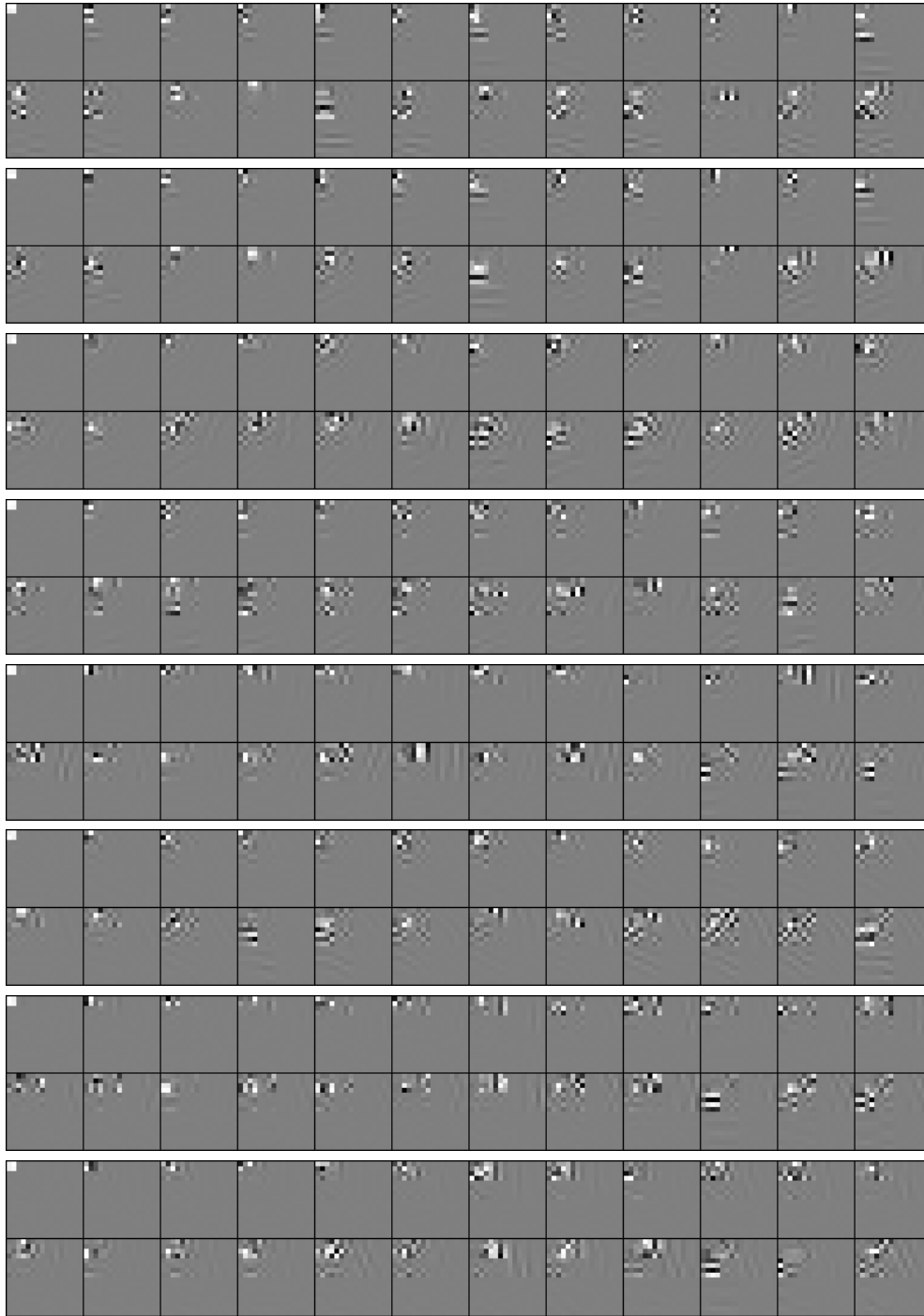


FIG. 6.5 – Approche globale : 24 premiers vecteurs (ordonnés ligne par ligne, colonne par colonne) de chaque base obtenue pour un apprentissage 8-KLT (sur la lame virtuelle 01 A.3) sur des blocs  $16 \times 16$  obtenus sur une décomposition dyadique sur 3 niveaux.

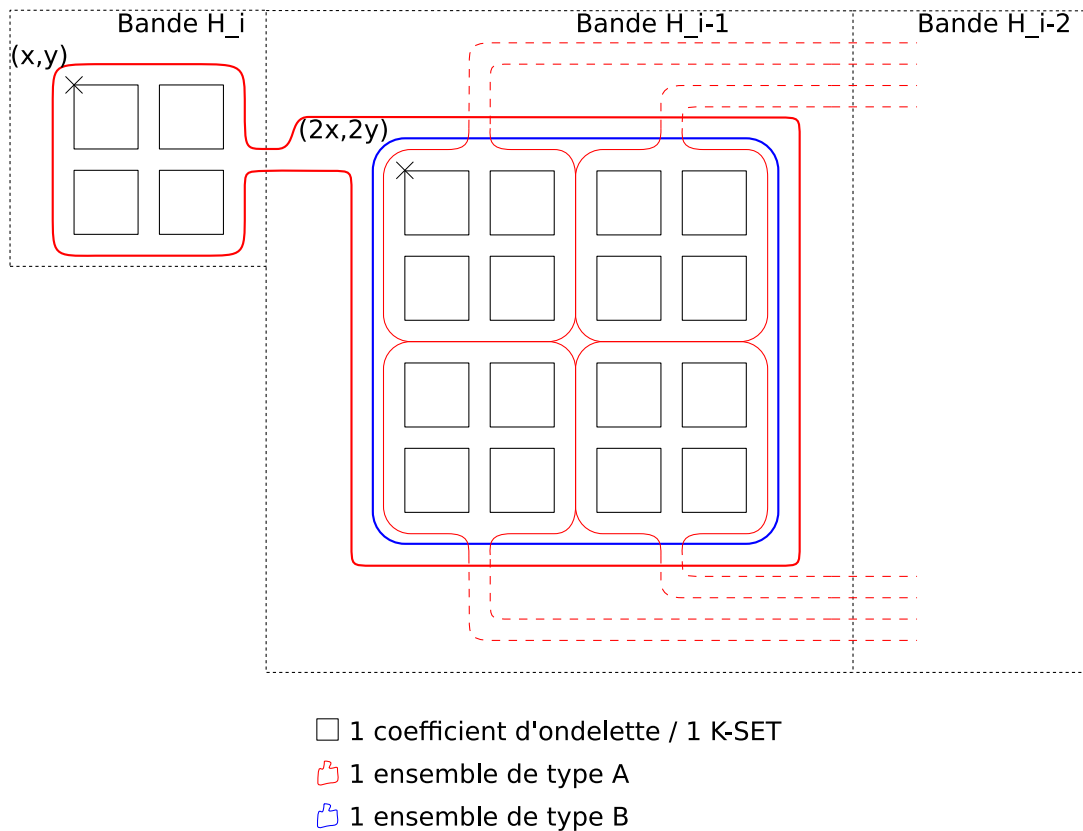


FIG. 6.6 – Ensembles de type *A* et *B* utilisés par SPIHT et pour le codage de l'approche intra-bande : le type *A* contient  $2 \times 2$  coefficients et 1 ensemble de type *B* ; le type *B* contient  $2 \times 2$  ensembles de type *A*.

compression/décompression sont identiques, alors que le fait de dissocier les sous-bandes devrait permettre de mieux adapter les KLT aux spécificités de chacune d'entre elle, et pourrait également permettre un codage multi-résolution.

Afin de tirer parti des spécificités spatio-fréquentielles exploitées par SPIHT concernant la distribution énergétique des coefficients de la transformée en ondelettes, la méthode de codage mise en place implique une taille de blocs identique dans chacune des sous-bandes pour pouvoir conserver les relations hiérarchiques de la FIG. 6.3. L'ensemble de type *A* de SPIHT (qui correspond à un MiSET contenant  $2 \times 2$  coefficients d'ondelettes et un ensemble de type *B* pour leurs descendants hiérarchiques) est alors étendu en un MaSET de type *A* qui contient  $2 \times 2$  K<sub>1</sub>SET et un MaSET de type *B*. Ce dernier est l'extension de l'ensemble de type *B* de SPIHT : il contient  $2 \times 2$  MaSET de type *A* faisant référence à la descendance hiérarchique (voir FIG. 6.6).

Puisque la majorité de l'énergie se situe dans le premier coefficient des KLT de la bande basse résolution/fréquence, la LIP est d'abord initialisée à l'aide du premier

coefficient des blocs de cette même bande et la LIS avec leurs  $K_2$ SET respectifs regroupés par blocs de  $2 \times 2$  en MaSET (comme le fait l'approche globale). Les MaSET de type  $A$  du plus haut niveau de décomposition sont également ajoutés à la LIS, pour chacune des orientations horizontale, verticale et diagonale. L'algorithme SPIHT est ensuite lancé, et lorsqu'un  $K_1$ SET devient significatif, la classe du bloc associé est transmise.

Comme avec l'approche globale, afin d'améliorer la scalabilité en qualité à bas débit, l'algorithme considère que le premier vecteur propre des KLT de la bande basse résolution n'est autre que la composante continue. L'indice de la classe d'un bloc de cette sous-bande n'est alors transmis que lorsque le  $K_2$ SET correspondant devient significatif.

Durant la rédaction de cette thèse nous avons pu constater qu'une approche similaire pour la décorrélation avait récemment été proposée [DCMT07, Del08] et qualifiée de « post-transformée dans le domaine ondelettes ». La différence majeure avec l'approche proposée réside dans l'apprentissage des bases. Dans [Del08], elle est effectuée pour une classification des blocs d'apprentissage suite à une analyse en bandelettes, qui va donc privilégier les orientations. Ici la classification est « apprise » conjointement avec les bases de manière à maximiser la compacité de l'énergie, et peut donc mieux s'adapter à des structures ne suivant pas des orientations strictes (courbes, textures, ...).

Dans le but de rendre le calcul des KLT plus efficace en ayant un nombre de vecteurs d'apprentissage plus important, les blocs utilisés pour construire la matrice  $A$  ont été extraits à l'aide d'une fenêtre glissante. Afin d'améliorer encore la décorrélation, il pourrait être envisagé d'optimiser la taille des blocs et le nombre de classes pour chacune des sous-bandes. Cependant si la taille des blocs devait varier d'une sous-bande à l'autre, la méthode de compression devrait être changée.

Des exemples de résultats d'apprentissage sont donnés dans la FIG. 6.7 et la FIG. 6.8.

### 6.3.3.3 Autres approches envisageables

Pour décorréler les coefficients d'ondelettes, deux méthodes intermédiaires pourraient être envisagées : une approche intra-orientation et une approche intra-résolution. Ces approches sont simplement mentionnées à titre d'exemples et ne seront pas exploitées.

La première suivrait le modèle de l'approche globale, mais avec quatre apprentissages différents : trois pour chacune des orientations, et un pour la bande basse résolution. Elle aurait pour objectif de mieux discerner les différentes orientations locales présentes dans les coefficients d'ondelettes et de traiter séparément la composante basse résolution.

La seconde suivrait plus le modèle de l'approche intra-bande, mais l'apprentissage prendrait en compte les coefficients des trois sous-bandes d'un même niveau de résolution. Même si intuitivement cette approche peut sembler moins adéquate que la précédente (puisque habituellement on entend plus parler des relations hiérarchiques entre les coefficients d'ondelettes que de leur relations spatiales), les bases obtenues pour l'approche globale (cf. FIG. 6.4 et FIG. 6.5) montrent qu'il existe une certaine corrélation entre ces sous-bandes dont les KLT peuvent tirer parti.

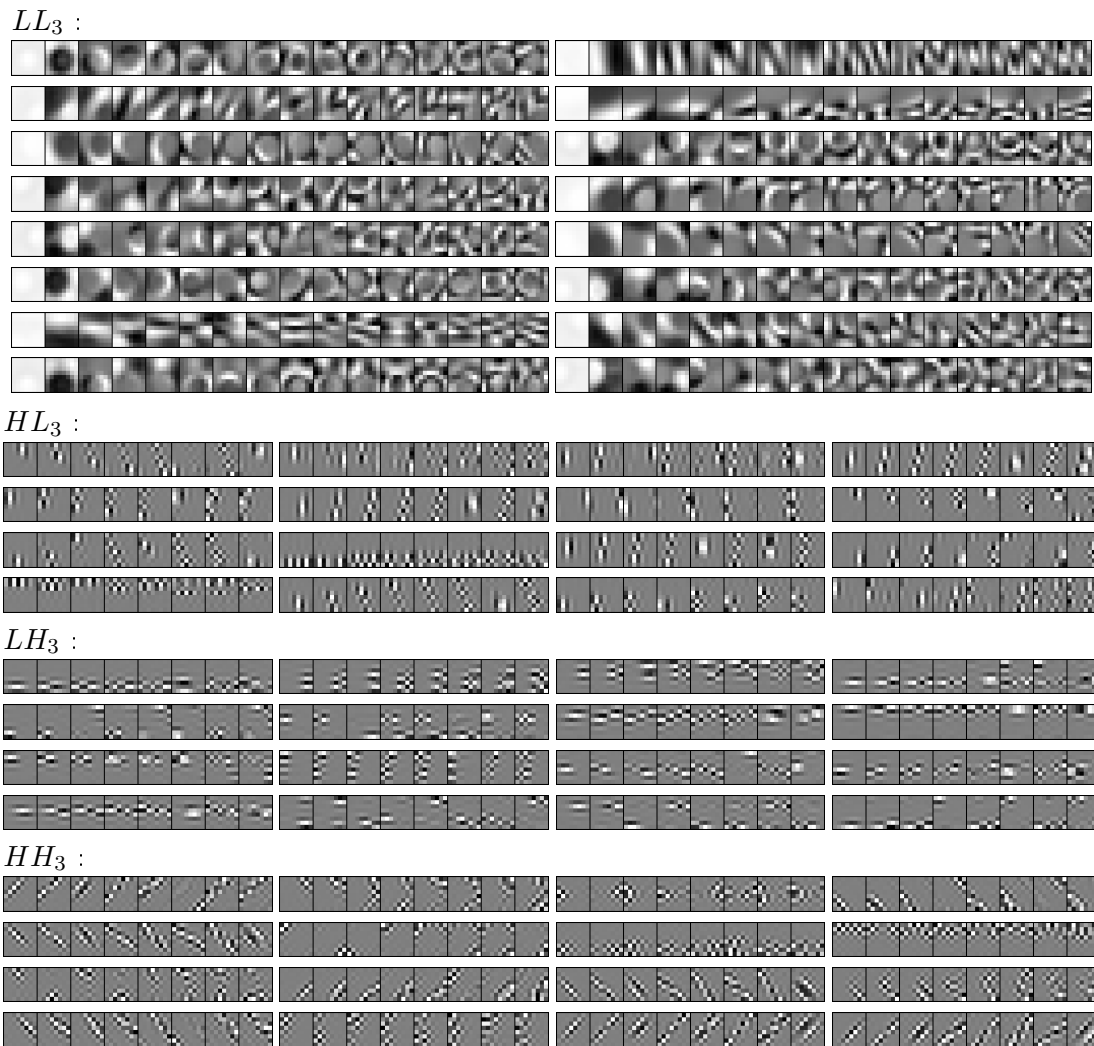


FIG. 6.7 – Approche intra sous-bande : premiers vecteurs de chaque base obtenue pour un apprentissage 16-KLT (sur la lame virtuelle 01 A.3) sur des blocs  $8 \times 8$  pour une décomposition dyadique sur 3 niveaux (1).



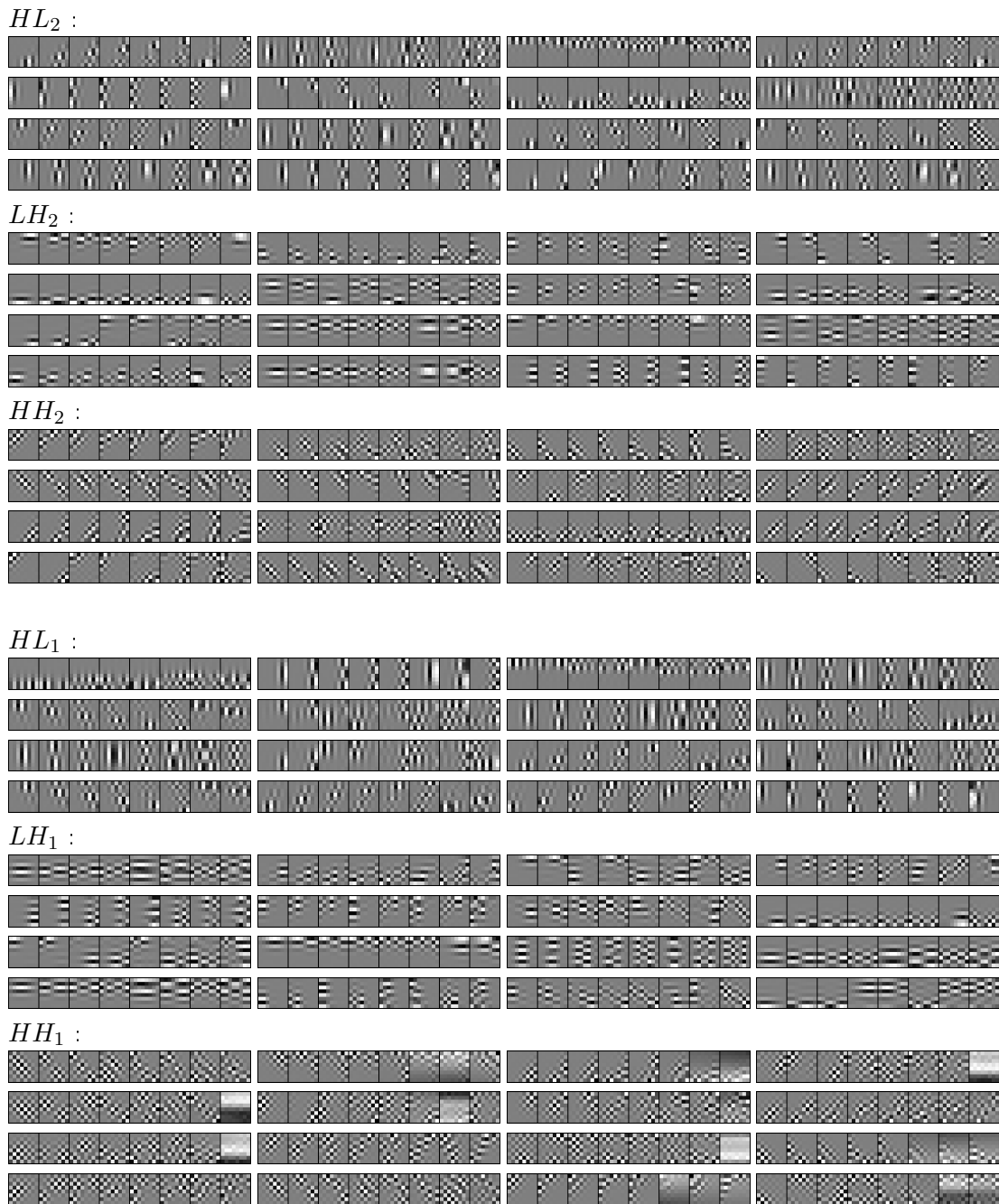


FIG. 6.8 – Approche intra sous-bande : premiers vecteurs de chaque base obtenue pour un apprentissage 16-KLT (sur la lame virtuelle 01 A.3) sur des blocs  $8 \times 8$  pour une décomposition dyadique sur 3 niveaux (2).

Les diverses approches présentées pourraient également s'appréhender sur plusieurs niveaux hiérarchiques : par exemple pour une décomposition en ondelettes sur 6 niveaux, plutôt que d'utiliser un apprentissage sur ces 6 niveaux (ce qui n'est pas vraiment envisageable), rien n'empêche d'effectuer un apprentissage sur les 3 premiers et un autre sur les 3 suivants par exemple, voir même d'essayer de décorréler les 3 niveaux suivants avec les coefficients les plus représentatifs des KLT des niveaux précédents (KLT imbriquée).

### 6.3.4 Résultats pour compression irréversible d'images anatomopathologiques

Les résultats qui vont suivre ont été obtenus à l'aide d'une implémentation visant à évaluer les performances des différentes approches (ADD, AG et AISB) dans un schéma de compression irréversible, car tous les calculs ont été effectués en virgule flottante (ici double précision), et scalable en qualité grâce aux algorithmes qui viennent d'être présentés. Leur utilisation dans un schéma réversible sera discutée dans la section 6.4.

Pour ces expérimentations, les apprentissages des  $K$ -KLT se sont appuyés sur les vecteurs formés à l'aide de l'ensemble des blocs extraits uniquement sur l'image 01 montrée en annexe A.3 (dimensions  $22000 \times 12267$ ), et dont la localisation spatiale chevauchait la ROI (11% des pixels), dans le domaine image ou transformé selon l'approche concernée. Pour les apprentissages dans le domaine transformé, afin de faciliter les comparaisons des différentes approches pour des blocs de taille  $8 \times 8$ , une décomposition en ondelettes sur 3 niveaux a été choisie (permet de produire des blocs hiérarchiques de la taille requise). Afin de remédier au manque de vecteurs d'apprentissage dans les sous-bande de faible résolution de l'AISB, les blocs ont été extraits de manière redondante, à l'aide d'une fenêtre glissante. Même si nous aurions pu nous amuser à optimiser la décomposition en ondelette comme au chapitre précédent, ce n'était pas vraiment le but de ce chapitre. Une décomposition dyadique a donc été arbitrairement fixée. Pour cette décomposition sur trois niveaux, les Sym(7,7) (symlettes à 7 moments nuls dont le support des filtres est de longueur 14) ont été retenues. Sur la famille des symlettes, elles permettent d'obtenir quasiment le meilleur rapport débit/PSNR, comme illustré dans la FIG. 6.9.

Afin de maximiser la qualité pour un débit quelconque, la base choisie pour transformer un bloc de données lors de la compression est celle qui minimise (6.5) sur le bloc en question.

#### 6.3.4.1 Approche directe

Avec l'approche directe (c.f. FIG. 6.10), pour une taille de blocs de  $8 \times 8$ , le coût de codage de l'indice de la classe, implique une chute des performances, croissante avec  $K$ , aux alentours de 0.1-0.2 bpp. Une fois l'indice de la classe codé, plus  $K$  est élevé, plus

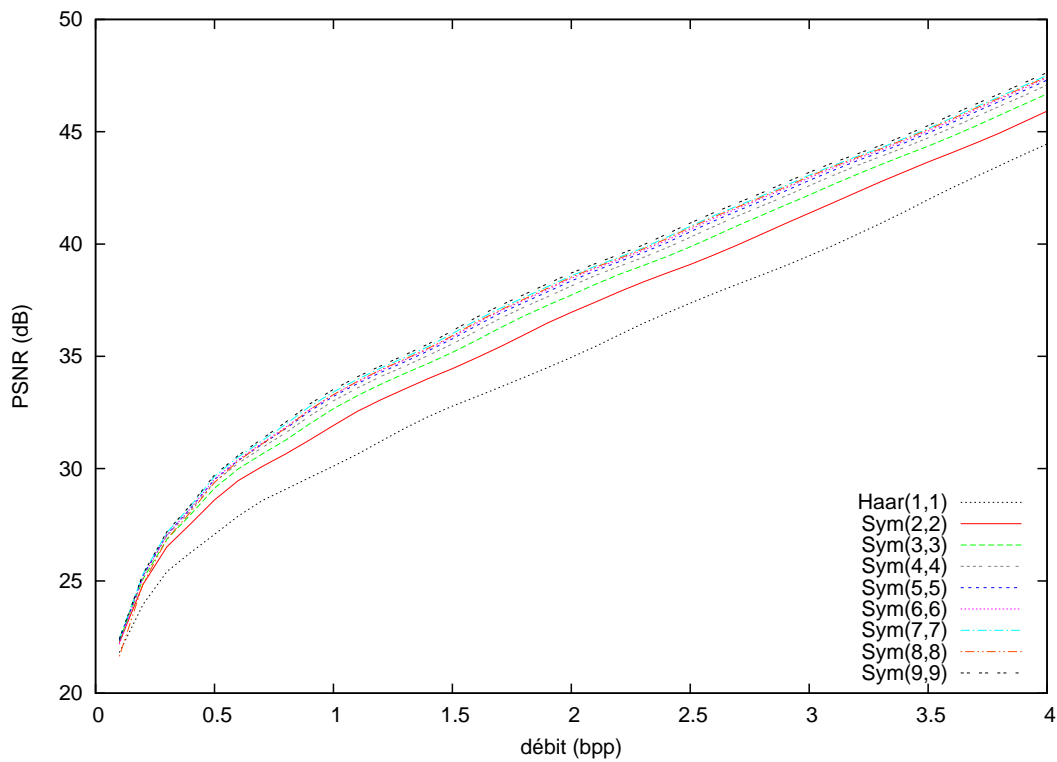


FIG. 6.9 – Courbes débit/PSNR obtenues avec le codeur SPIHT pour une décomposition dyadique sur 3 niveaux utilisant les symlettes. Ces résultats sont calculés sur l'ensemble des patches de test pour les lames virtuelles.

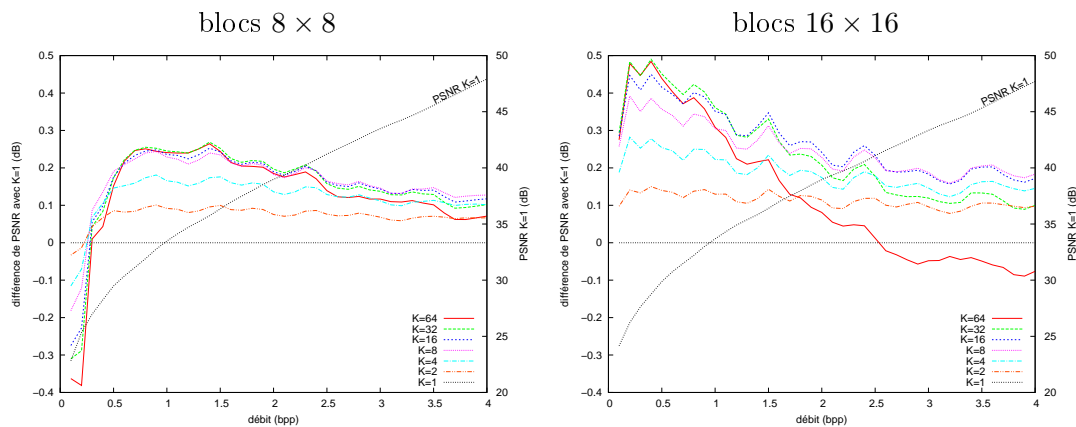


FIG. 6.10 – Approche directe (dans le domaine image) : différence de PSNR (sur l'ensemble des 120 échantillons de test) entre un apprentissage multi et un apprentissage mono KLT.

le gain sur la qualité de reconstruction (PSNR) est important entre 0.6 et 1.5 bpp, avec une asymptote aux alentours de  $K = 32$  pour ces images. Au contraire, au delà de 1.5 bpp, plus  $K$  est important, plus le gain a tendance à décroître. Avec les apprentissages réalisés,  $K = 8$  semblent être l'un des meilleurs compromis pour conserver une bonne qualité à bas et moyen débit.

En passant à une taille de blocs  $16 \times 16$ , le coût de codage de la classe (divisé par 4) altère moins les performances à bas débit. L'approximation de l'image originale est alors meilleure à bas débit lorsque  $K$  augmente, jusqu'à  $K = 32$ . Il peut ainsi être réalisé un gain de quasiment 0.5 dB à 0.2, 0.4 bpp en comparaison à un apprentissage mono KLT. Au delà de 0.4 bpp, plus le nombre de classe est élevé, plus l'amélioration tend à décroître. Ici  $K = 16$  semble être l'un des meilleurs compromis parmi les valeurs testées.

Si on prend deux apprentissages  $K$ -KLT  $A_1$  et  $A_2$ , le fait d'avoir un nombre de bases de décomposition  $K_1 > K_2$  rend théoriquement l'apprentissage de  $A_1$  plus spécifique et donc mieux adapté que  $A_2$  pour rendre l'énergie compacte. En s'appuyant sur les propriétés de la KLT (c.f. 6.1.2), une fois que l'indice de la classe est codé, si à un débit  $d_0$  donné l'erreur quadratique moyenne (MSE) pour  $A_1$  est inférieure à la MSE pour  $A_2$ , alors pour tout débit  $d \geq d_0$  (avec l'approche de codage utilisée, en régime strictement lossy), le PSNR pour  $A_1$  devrait intuitivement toujours rester supérieur ou égal à celui obtenu avec  $A_2$ . Hors ceci ne correspond clairement pas aux résultats obtenus. On peut supposer que la différence entre le comportement théorique et le comportement pratique provient principalement d'un apprentissage rendu de moins en moins efficace par l'augmentation de  $K$  et de  $D$ . En effet, quels que soient  $K$  et  $D$ , le nombre de vecteurs d'apprentissage utilisés est resté similaire. Ainsi l'estimation des KLT a été rendue de moins en moins robuste en augmentant la dimension du problème, avec  $D$ , et/ou en réduisant le nombre d'individus représentatifs d'une classe, avec l'enrichissement du nombre de classes  $K$ . On peut donc espérer rehausser les performances à haut débit avec une base d'apprentissage plus fournie, et en contrepartie une durée d'apprentissage plus longue. Une autre approche consisterait à chercher une méthode d'apprentissage plus robuste, peut-être en s'appuyant sur [CLMW09] par exemple.

#### 6.3.4.2 Approche globale

Pour l'approche globale (c.f. FIG. 6.11), avec une taille de blocs hiérarchique de  $8 \times 8$ , le coût de codage de l'indice de la classe, expliquant la chute des performances à 0.1-0.2 bpp, semble trop important pour permettre une amélioration significative du PSNR à plus haut débit. Les meilleurs résultats sont obtenus lorsque  $K = 2$  qui n'améliore que très légèrement ceux pour  $K = 1$  à partir de 0.4 bpp (jusqu'à +0.05 dB seulement). En passant à une taille de blocs hiérarchique  $16 \times 16$  (i.e. le bloc de référence dans la bande basse résolution à une taille  $2 \times 2$ ), le coût de codage de la classe (divisé par 4) altère beaucoup moins les performances à bas débit. Ainsi, jusqu'à environ  $K = 32$ , plus  $K$  augmente, meilleure est l'approximation de l'image originale à bas débit, avec un gain supérieur à 0.3 dB à 0.2, 0.4 bpp. Cependant, il peut également être constaté que plus le nombre de classe est élevé, plus l'amélioration tend à décroître rapidement au delà de

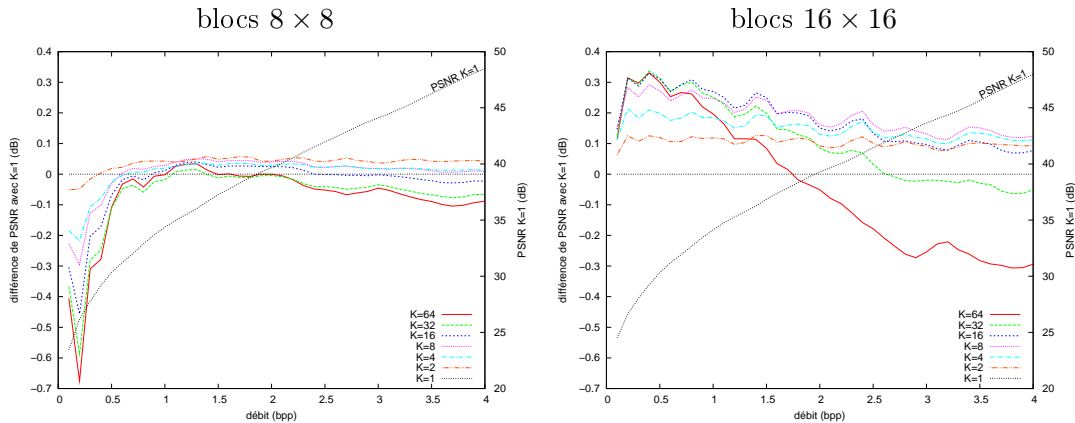


FIG. 6.11 – Approche globale : différence de PSNR (sur l’ensemble des 120 échantillons de test) entre un apprentissage multi et un apprentissage mono KLT.

0.4 bpp.  $K = 8$  ou 16 semblent être les meilleurs compromis parmi les valeurs testées.

Comme pour l’approche directe, les chutes de performances à haut débit lorsque  $K$  augmente sont probablement liées à un manque de données d’apprentissage (c.f. 6.3.4.1).

### 6.3.4.3 Approche intra sous-bande

Les résultats pour l’apprentissage intra sous-bande sont présentés dans la FIG. 6.12. Le PSNR augmente avec  $K$ , quel que soit le débit. Sur les lames virtuelles prises comme référence, une taille de bloc  $8 \times 8$  semble mieux adaptée avec cette approche qu’une taille  $16 \times 16$ , aussi bien à bas qu’à haut débit (en comparant les courbes de PSNR obtenues pour  $K = 1$ ), contrairement aux deux approches précédentes pour lesquelles les blocs  $16 \times 16$  conviennent mieux pour les bas débits. Pour l’AISB, ceci peut s’expliquer en partie par une décorrélation de la sous-bande basse résolution (composante continue) suffisante avec les blocs  $8 \times 8$  qui prennent déjà en considération l’équivalent d’une résolution spatiale de  $64 \times 64$  pixels en pleine résolution (sans compter le support des ondelettes). Alors que pour l’ADD et l’AG, la résolution spatiale prise en compte est égale à la taille des blocs utilisés, et une taille de  $8 \times 8$  semble insuffisante pour la haute définition des lames virtuelles.

### 6.3.4.4 Comparaison des meilleurs résultats

En utilisant les résultats comparatifs de la FIG. 6.13, qui prend comme référence les performances de la décomposition en ondelettes (sym(7,7)) sur 3 niveaux, il peut être constaté que la modélisation multi-KLT dans le domaine direct pour une représentation comparable (blocs  $8 \times 8$ ), même si légèrement pénalisée par le coût de codage de la classe vers 0.2 bpp, permet d’obtenir un meilleur compromis débit/distorsion dès 0.4 bpp et d’améliorer le PSNR de plus de 0.25 dB au delà de 1 bpp et 0.5 dB au delà de 1.5 bpp. L’approche globale montre que l’utilisation des  $K_i$ SET permet de mieux

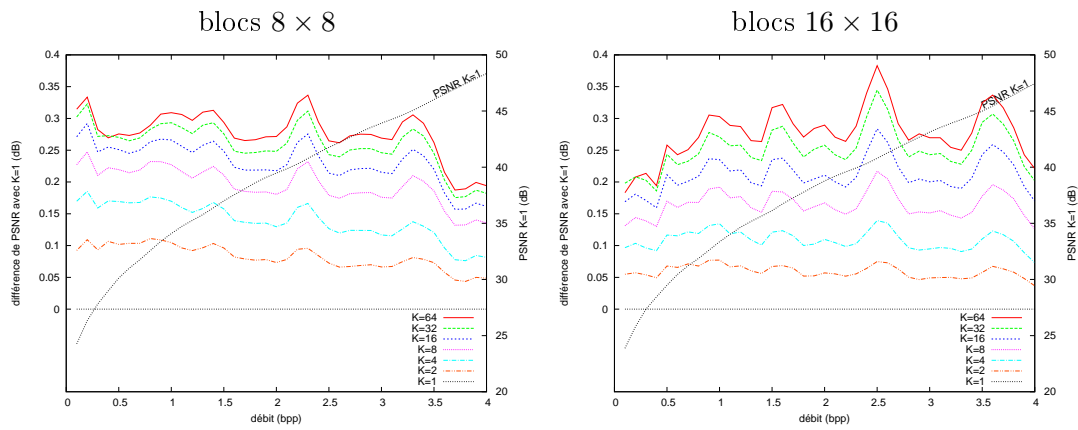


FIG. 6.12 – Approche intra : différence de PSNR (sur l'ensemble des 120 échantillons de test) entre un apprentissage multi et un apprentissage mono KLT.

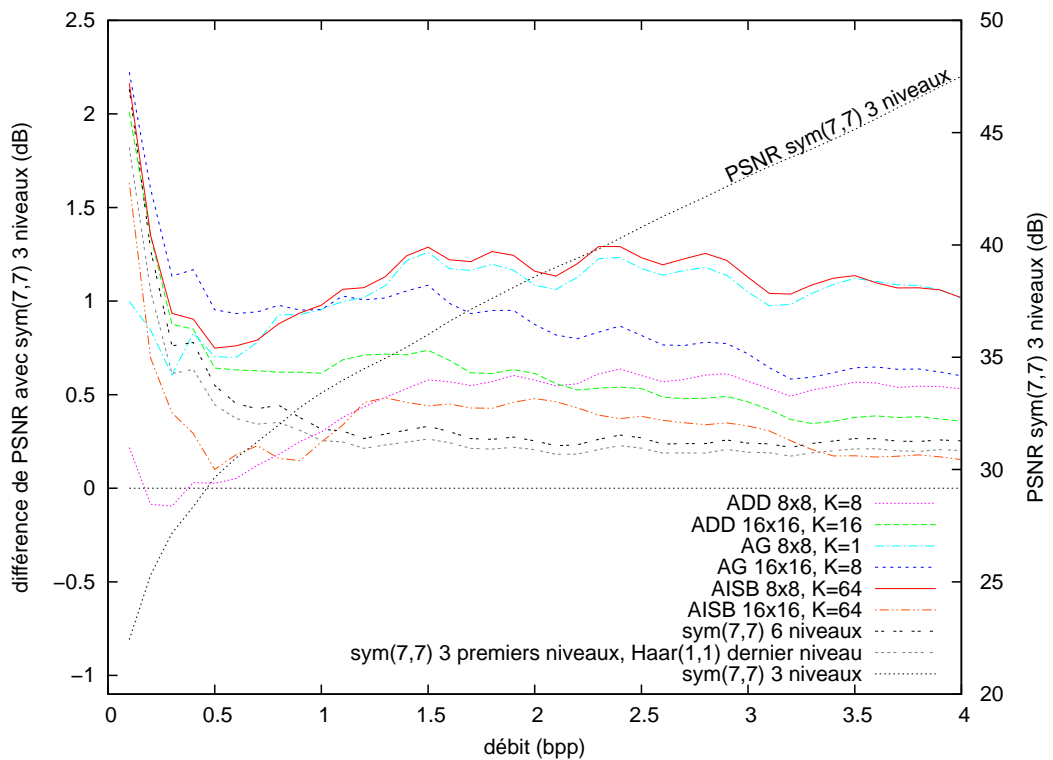


FIG. 6.13 – Meilleures performances : différence de PSNR (sur l'ensemble des 120 échantillons de test) entre un apprentissage multi KLT et la transformée en ondelettes.

organiser l'énergie dans le flux de données que les arbres hiérarchiques de SPIHT, en offrant quasiment les meilleurs résultats à moyen et haut débit pour des blocs  $(8 \times 8)$  pour une unique KLT, et à bas débit pour des blocs  $(16 \times 16)$  avec seulement 8 KLT (cette version peut être comparée à la décomposition en ondelettes sur 3 niveaux plus un dernier niveau de décomposition utilisant l'ondelette de Haar). La technique qui semble cependant être la plus efficace repose sur l'approche intra sous-bande (qui peut aussi être comparée à une décomposition en ondelettes sur 6 niveaux, concernant la concentration de l'énergie de la sous-bande basse résolution) en offrant presque la plus faible distorsion quel que soit le débit.

Pour toutes les approches, le passage à des blocs  $(16 \times 16)$  est assez peu encourageant, surtout à haut débit. Ces faibles performances peuvent en partie être attribuées à un apprentissage moins fiable, qui nécessiterait certainement beaucoup plus de données d'entraînement mais rendrait la durée d'optimisation des KLT plus critique. Cependant, comme il sera vu dans la section 6.5, le modèle utilisé pour le codage est sous-optimal et tend à pénaliser les approches  $16 \times 16$ . Lorsqu'un codage arithmétique est ajouté pour supprimer les redondances statistiques, leur efficacité est nettement améliorée.

Pour les images anatomopathologiques de test, les résultats montrent que la composante basse résolution d'une décomposition sur 3 niveaux peut être davantage décorrélée et ainsi permettre de gagner environ 2 dB à 0.1 bpp (sym(7,7) 6 niveaux). Cette décorrélation de la composante basse résolution explique en partie les meilleures performances à bas débit obtenues par l'ensemble des décompositions (hormis pour les approches directe et globale  $8 \times 8$  qui ne permettent pas d'exploiter cette corrélation).

Un exemple visuel de résultats obtenus sur le patch 05-01 est donné dans la FIG. 6.14 pour un débit de 0.5bpp et dans la FIG. 6.15 pour un débit de 1bpp.

## 6.4 Approximation des transformées pour une compression sans perte

Afin d'appliquer la transformée en ondelettes orthonormales dans un schéma de compression pouvant aller jusqu'au sans perte, celles-ci ont été préalablement factorisées en étapes de lifting en utilisant [DS98]. En fonction de la taille du support, il existe un nombre exponentiel de factorisations possibles. Celle retenue essaie, avec l'aide d'heuristiques locales pour élaguer l'arbre des possibilités de factorisations, de réduire les possibilités de débordements de capacités calculatoires qui pourraient survenir lors de l'application du lifting sur des valeurs entières.

### 6.4.1 Approximation réversible des KLT par factorisation PLUS

Pour pouvoir appliquer les KLT de manière réversible, toutes les bases  $(\kappa)_k$  sont factorisées en quatre matrices  $(\kappa)_k = P_k L_k U_k S_k$  dont la structure permet une approximation en étapes de lifting [HS01].  $P_k$  est une matrice de permutations (que l'on peut

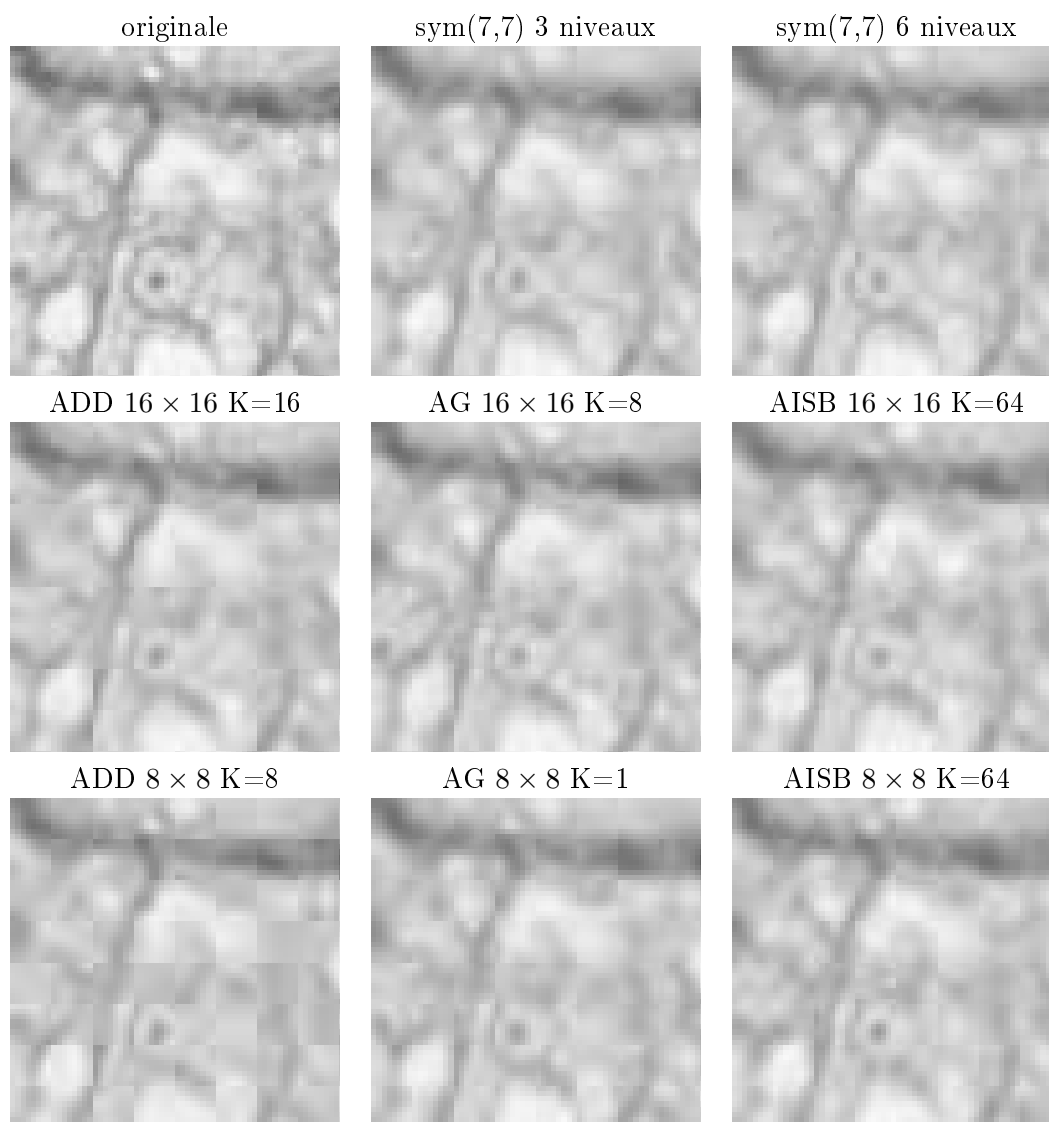


FIG. 6.14 – Exemple visuel (pour affichage sur écran) des distorsions pour une compression à 0.5bpp, blocs de taille  $64 \times 64$  extraits du patch 05-01.



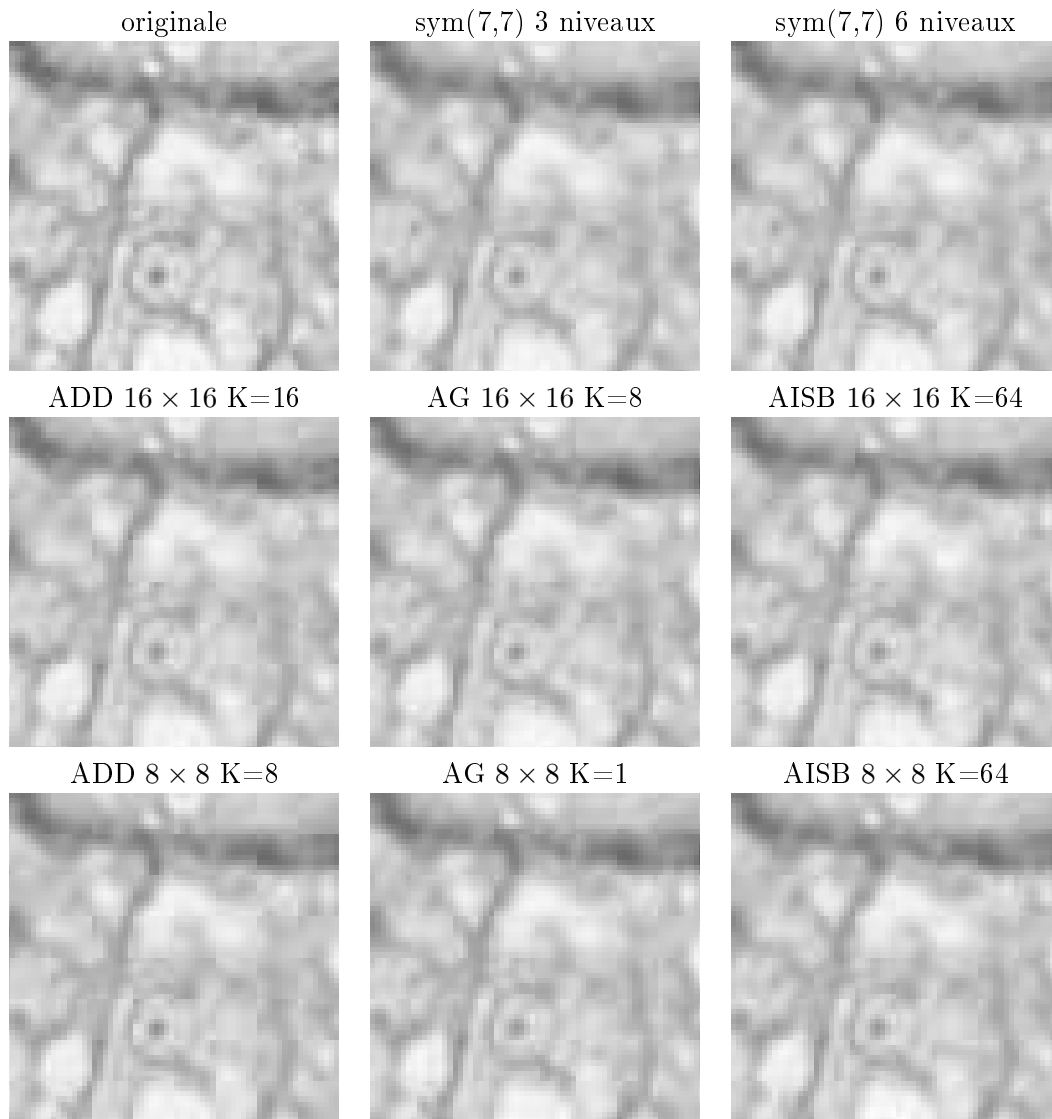


FIG. 6.15 – Exemple visuel (pour affichage sur écran) des distorsions pour une compression à 1bpp, blocs de taille  $64 \times 64$  extraits du patch 05-01.

chercher à construire selon divers critères),  $L_k$  une matrice triangulaire inférieure à diagonale unité,  $U_k$  une matrice triangulaire supérieure dont toutes les valeurs de la diagonale valent 1, sauf celle de la dernière ligne/colonne qui peut valoir  $\pm 1$ , et enfin  $S_k$  une matrice à diagonale unité pour laquelle toutes les autres valeurs sont nulles, sauf la dernière ligne.

Les  $P_k$  sont des matrices de permutations (forcément inversibles), et les étapes de lifting pour les autres matrices sont de la forme :

$$\hat{x}(i) \leftarrow \sigma_i \hat{x}(i) + \left\lfloor \sum_{j \neq i} \alpha_j \hat{x}(j) \right\rfloor, \quad (6.9)$$

avec  $\lfloor \cdot \rfloor$  l'arrondi à l'entier le plus proche,  $\hat{x}$  le vecteur contenant les valeurs intermédiaires du calcul de la transformée d'un vecteur  $x$ ,  $\sigma_i = \pm 1 = M(i, i)$ , où  $M$  est la matrice de lifting ( $L_k$ ,  $U_k$  ou  $S_k$ ), et  $\alpha_j$  les coefficients de lifting égaux à  $M(i, j)$  si on calcule  $\hat{x}M$  ou à  $M(j, i)$  si on calcule  $M\hat{x}$  (c.f. 6.4.1.1).

L'ordonnancement des étapes de lifting doit alors s'effectuer de  $i = 1$  à  $D$  pour les matrices triangulaires inférieures et de  $i = D$  à 1 pour les triangulaires supérieures dans le cas i) et dans l'ordre inverse dans le cas ii).

Il peut être noté que pour obtenir la transformée (réversible) inverse, il suffit de traiter toutes les étapes dans l'ordre inverse, et d'utiliser les schémas de lifting inverses :

$$\hat{x}(i) \leftarrow \sigma_i \left( \hat{x}(i) - \left\lfloor \sum_{j \neq i} \alpha_j \hat{x}(j) \right\rfloor \right). \quad (6.10)$$

#### 6.4.1.1 Choix de l'implémentation pour la transformée

Si on considère qu'une base orthonormée  $\beta$  de dimension  $D$  est formée par des vecteurs colonnes, on peut transformer un vecteur ligne  $x$  par la multiplication  $x\beta$ . Cependant, si  $\beta$  est factorisée en  $PLUS$ , l'approximation par lifting de  $xPLUS$  n'est pas spécialement idéale. En effet, la dernière multiplication :  $(xPLU)S$  nécessitera  $D-1$  étapes de lifting, chacune ne prenant en compte qu'un seul coefficient :  $\hat{x}_\beta(i) = \hat{x}_\beta(i) + \lfloor S(D, j)\hat{x}_\beta(D) \rfloor$ .

Même si  $P$  peut être construite de sorte à itérativement (sur  $j$ ) réduire au maximum l'amplitude des coefficients  $S(D, j)$ , nous avons pu constater que des erreurs d'approximations importantes pouvaient survenir lorsque des  $S(D, j)$  étaient inévitablement élevés. En effet, puisque les valeurs intermédiaires de  $\hat{x}_\beta$  sont déjà des approximations, l'erreur sur  $\hat{x}_\beta(D)$  est amplifiée d'un facteur  $S(D, j)$  durant l'étape de lifting. Donc, si  $S(D, j)$  possède une forte amplitude, un bruit de variance importante est ajouté sur  $\hat{x}_\beta(i)$  après lifting. Avec la méthode de codage des KLT présentée précédemment, ce bruit va perturber la compression, surtout si des erreurs importantes interviennent sur les derniers coefficients de la KLT.

Afin de contourner ce problème, l'approche retenue dans notre implémentation consiste à factoriser  $\beta^\top$ , de sorte à ce que la transformée soit effectuée par  $PLUSx$

( $x$  est ici considéré comme un vecteur colonne). La multiplication par  $S$  de  $x$  ne nécessite alors plus que d'une seule étape de lifting au lieu de  $D - 1$ , permettant au passage de réduire le bruit dû aux erreurs d'arrondis. Cette méthode simple nous a permis d'améliorer l'approximation des KLT et de rendre les résultats de la compression plus stables.

#### 6.4.1.2 Remarques sur la factorisation PLUS

L'approximation d'une transformée à valeurs réelles par une succession d'étapes de lifting n'est pas magique. Plus le nombre d'étapes de lifting sera important, moins l'approximation sera correcte à cause du bruit introduit par les erreurs d'arrondis.

De plus, il a pu être constaté lors des expérimentations que plus  $D$  était élevé, plus la factorisation *PLUS* des KLT pour une implémentation réversible devenait incertaine. En effet, plus  $D$  devient important, plus des coefficients d'amplitudes importantes tendent à apparaître dans les matrices *PLUS*. Ainsi, non seulement du bruit risque d'être généré, mais également des dépassements de capacité de représentation peuvent survenir pour les valeurs entières intermédiaires du calcul de la transformée. Par exemple, pour  $D = 256$  dans le cas d'une transformée par blocs  $16 \times 16$  sur des images monochromatiques 8 bits, une représentation entière sur 32 bits n'était pas toujours suffisante et des entiers sur 64 bits ont donc été requis. Dans cette même configuration ( $D = 256$ ) les erreurs d'approximation deviennent souvent problématiques : de faibles erreurs peuvent apparaître en particulier sur la diagonale de  $U$ , qui rendent l'implémentation par lifting inexacte (même si toujours réversible), et se répercutent principalement sur les valeurs transformées associées au  $D$  ième vecteur avant permutation. C'est pourquoi aucun résultat sans perte ne sera présenté pour des tailles de blocs supérieures à  $8 \times 8$ , valeur pour laquelle les erreurs d'approximation sur la diagonale sont encore négligeables. Une implémentation avec pertes, fonctionnant uniquement sur des valeurs flottantes double précision et utilisant la même factorisation PLUS dont seules les valeurs de la diagonale de  $U$  sont altérées en considérant qu'elles sont toutes sensées être exactement égales à 1 sauf la dernière égale à 1 ou  $-1$ , confirme ces résultats : des facteurs d'erreurs d'amplitudes similaires à une implémentation sans perte sont obtenus (c.f. FIG. 6.16).

Il pourrait être bénéfique de s'intéresser à l'optimisation de la factorisation dans le but de rendre l'estimation la plus fiable possible. Afin de réduire au maximum l'amplification du bruit généré par la quantification de l'opérateur d'arrondi, il pourrait être tenté de favoriser l'utilisation des coefficients  $\alpha$  élevés (si ceux-ci sont incontournables) pour les premières étapes de lifting (lorsqu'aucun bruit d'arrondi n'est encore présent) et tenter d'utiliser au maximum des coefficients faibles sur les dernières étapes. La prise en compte des bornes des valeurs potentiellement prise par les valeurs intermédiaires pourrait également intervenir lors de l'optimisation, afin de minimiser les possibilités de débordements de capacité pour leur représentation entière où de prévoir un nombre de bits suffisant pour garantir l'intégrité des calculs (les débordements n'empêchent pas une compression sans perte car les mêmes sont obtenus lors de la décompression, ce-

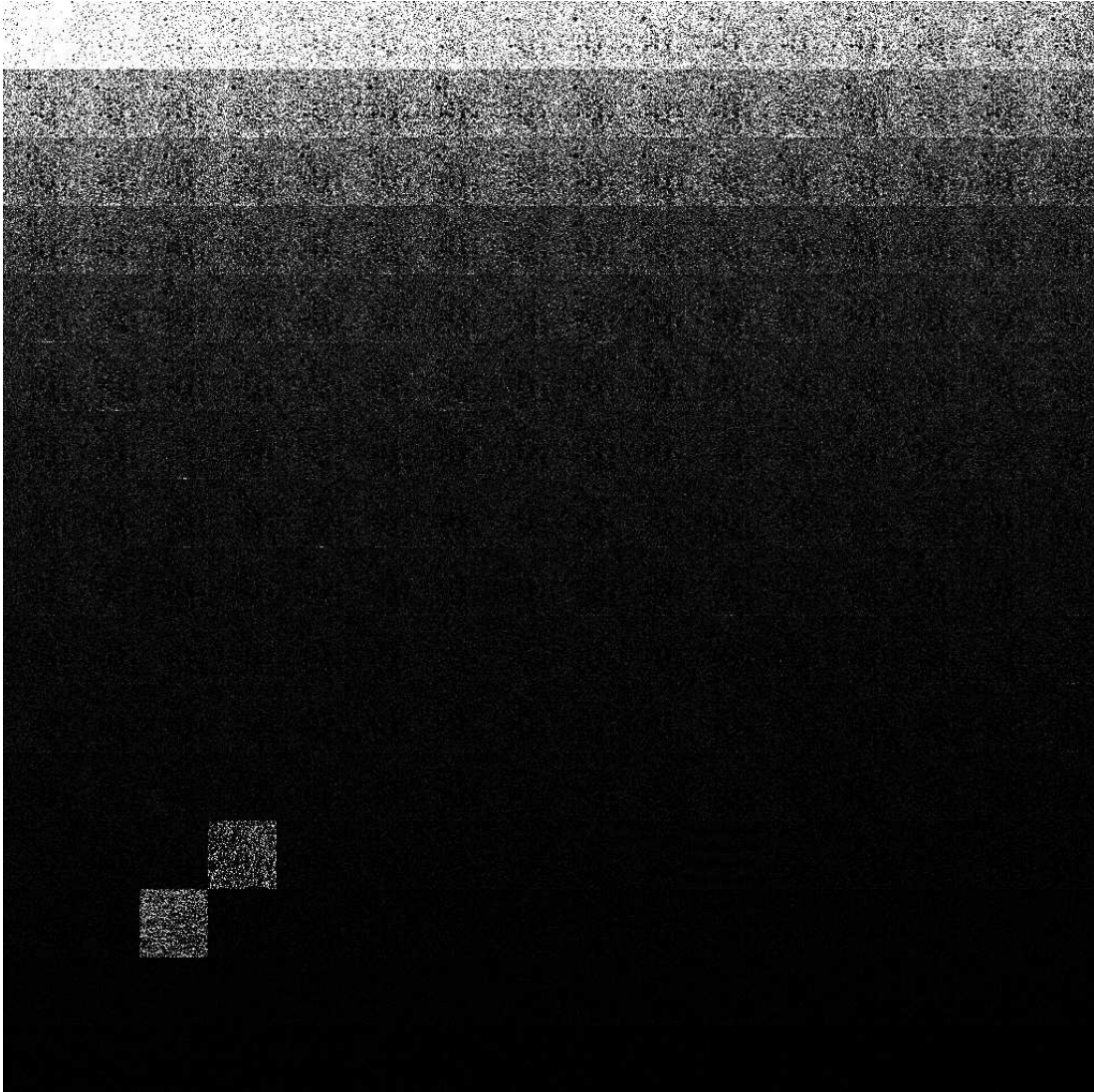


FIG. 6.16 – Exemple de problèmes pouvant être engendrés par les erreurs d'approximations de la factorisation PLUS lorsque la dimension  $D$  des bases est trop importante. Le patch de test 01-01 est ici transformée avec l'approche globale utilisant un apprentissage sur deux classes pour des blocs hiérarchiques  $16 \times 16$  obtenus à partir d'une décomposition dyadique sur 3 niveaux ( $D = 256$ ). Les coefficients sont réorganisés en sous-bandes en fonction de l'indice du vecteur propre. Les valeurs absolues des coefficients sont ajustées entre 0 et 64. Les 196 ièmes coefficients de la première base et les 211 ièmes de la seconde, calculées comme étant les 256 ièmes avant la permutation ( $P$ ), sont les valeurs problématiques. Ces coefficients, censés être de faible amplitude si la factorisation était exacte, engendrent de grosses pertes de performances pour la compression avec l'extension de SPIHT présentée.

pendant ils altèrent la qualité de l'approximation des KLT et peuvent diminuer les taux de compression). Enfin, il faudrait également prendre en considération des erreurs d'approximations, liées aux calculs en virgule flottante, intervenant lors de la construction des matrices PLUS. Il se pourrait que d'autres approches que la factorisation PLUS soient plus appropriées.

#### 6.4.2 Remarque concernant la normalisation réversible d'ondelettes bi-orthogonales

L'approche de transformation globale peut également permettre de normaliser les coefficients d'ondelettes de manière réversible lorsque la transformée est bi-orthogonale. Pour ce faire on peut utiliser un bloc de coefficients hiérarchiques  $2^{N_V} \times 2^{N_H}$  où  $N_H$  et  $N_V$  sont le nombre de décompositions horizontales et verticales utilisée pour obtenir la bande basse fréquence. Le facteur de normalisation utilisé en compression irréversible pour un coefficient d'ondelette est généralement  $1/\sqrt{\|\Phi_l^*\|^2}$  pour la bande basses fréquences ou  $1/\sqrt{\|\Psi_l^*\|^2}$  pour les autres composantes, afin de normaliser l'énergie des coefficients transformés en fonction de celle des filtres de synthèse pour toutes les bandes  $l$ . Cette normalisation est optimale pour des transformées orthogonales, ou si les erreurs de quantification ne sont pas corrélées. Bien que ces hypothèses soient généralement fausses, et afin de simplifier le processus de codage par plan de bits, il est souvent supposé que l'une des deux est presque vraie.

Ainsi, en considérant que les coefficients du bloc sont organisés dans un vecteur  $b$ , pour les normaliser il suffit de multiplier  $b$  par une matrice diagonale  $B_N$  qui associe à chacun des coefficients du bloc un facteur de normalisation. Afin de pouvoir factoriser  $B_N$  en étapes de lifting en utilisant [HS01], il faut que  $\det(B_N) = 1$ . Ceci n'est pas le cas mais rien n'empêche d'utiliser une matrice de normalisation  $B'_N = \beta B_N$ . Pour que  $\det(B'_N) = 1$ , on prend  $\beta = \sqrt[D]{\det(B_N)}$ , avec  $D = 2^{N_V+N_H}$  la dimension du bloc à « transformer ». L'énergie de toutes les bandes sera alors normalisée à un facteur  $\beta^2$  (commun) près.

Pour l'approche par KLT globale on peut combiner cette normalisation à la transformée en factorisant directement la matrice  $(B'_N \kappa)$  (l'apprentissage des KLT devra être fait sur des vecteurs normalisés).

Bien que pouvant sembler attractive, cette approche est assez limitée à cause des problèmes engendrés par la factorisation PLUS (c.f. 6.4.1.2), et par la dimension du problème  $D$  qui croît de manière exponentielle avec le nombre de niveaux de décomposition en ondelettes.

#### 6.4.3 Taux de compression sans perte

Les taux de compression sans perte sur les patches de test sont présentés dans la TAB. 6.1. Les résultats pour une décomposition en ondelettes sur 6 niveaux sont également inclus pour une comparaison plus réaliste avec l'approche intra sous-bande dont l'utilisation permet de générer un bande basses fréquences lors de la transformée par KLT de la sous-bande basse résolution (à l'aide du premier vecteur qui correspond à la

composante continue : DC) dont les dimensions sont identiques à la composante basse résolution d'une décomposition en ondelettes sur 6 niveaux.

Pour commencer, en comparant les résultats obtenus sur l'ensemble des patches à ceux obtenus en omettant les patches extraits de l'image d'apprentissage, il peut être constaté que bien que l'apprentissage n'ait été effectué que sur une seule image, les comportements de la compression restent similaires sur l'ensemble de la base de tests. Ainsi l'apprentissage semble pertinent et c'est pourquoi, dans le reste du chapitre, il n'y a pas de distinction entre les patches appartenant à l'image d'apprentissage et les autres.

L'impact du nombre de classe utilisées ( $K$ ) varie en fonction de l'approche utilisée. Avec l'ADD, lorsque  $K > 1$ , les résultats sont à peu près équivalents : les débits oscillent entre 5.123 et 5.129 bpp. Ainsi, la réduction de l'entropie obtenue par la compacité énergétique est reperdue par le coût de codage de l'indice de la classe. Pour l'AG, les meilleures performances sont obtenues lorsque  $K = 2$  et au delà de 8 classes elle tendent à chuter. L'apport entropique de l'AG est donc insuffisant en comparaison au coût de codage des index pour améliorer la compression sans perte, contrairement à l'ASB pour laquelle la compression progresse de manière asymptotique avec  $K$  (du moins jusqu'à 64 classes).

Avec l'ASB, on constate que la modélisation par blocs du contenu des sous-bandes pour un codage utilisant des  $K_iSET$  après une réorganisation de l'information à l'aide d'une unique KLT est moins efficace que l'exploitation, pour chaque coefficient, des relations hiérarchiques utilisées par SPIHT dans le domaine ondelettes. Ainsi, lorsque  $K = 1$  une perte de codage de l'ordre de 0.7% peut être perçue. Cependant, comme il vient d'être mentionné, plus  $K$  augmente, plus la réorganisation de l'information favorise l'ASB. Un gain de codage de presque 1% est alors obtenu lorsque le nombre de classe par sous-bande atteint 64. La compression sans perte est alors à peu près équivalente à celle de SPIHT pour l'utilisation d'une transformée avec six niveaux de décomposition.

Puisque le but recherché est d'avoir une représentation progressive en qualité efficace à bas et moyen débit, la sélection de la base est faite à l'aide d'un critère de compacité énergétique et non à partir d'un critère de coût de codage. Cependant, les résultats montrent que l'utilisation d'une approche par  $K$ -KLT peut permettre d'obtenir des taux de compression sans perte équivalents voire meilleurs qu'avec une décomposition en ondelettes uniquement. L'approche n'est donc pas défavorable à l'archivage.

Les meilleurs résultats sans perte de l'ADD, l'AG et l'ASB sont également très proches, avec une légère préférence pour l'AG.

#### 6.4.4 Scalabilité en qualité pour une compression *lossy-to-lossless*

Les résultats débit/PSNR des différentes représentations scalables sont présentés dans la FIG. 6.17. En comparaison à la compression irréversible (courbes de droite), les résultats sont à peu près identiques jusqu'à environ 1 bpp. Au delà, l'impact du bruit introduit par les étapes de lifting de la factorisation PLUS apparaît très clairement, et

	$K$	patchs 1-30 × 4		patchs 2-30 × 4	
		débit (bpp)	gain (%)	débit (bpp)	gain (%)
Sym(7,7) 3 niveaux	-	5.1701	0	5.1730	0
Approche Domaine Direct (ADD)	1	5.1637	0.12	5.1613	0.23
	2	5.1290	0.79	5.1275	0.88
	4	<b>5.1230</b>	<b>0.91</b>	<b>5.1224</b>	<b>0.98</b>
	8	5.1274	0.83	5.1279	0.87
	16	5.1268	0.84	5.1279	0.87
	32	5.1238	0.90	5.1253	0.92
	64	5.1276	0.82	5.1296	0.84
Approche Globale (AG)	1	5.1222	0.93	5.1238	0.95
	2	<b>5.1124</b>	<b>1.12</b>	<b>5.1141</b>	<b>1.14</b>
	4	5.1186	1.00	5.1204	1.02
	8	5.1178	1.01	5.1196	1.03
	16	5.1197	0.98	5.1215	1.00
	32	5.1283	0.81	5.1302	0.83
	64	5.1290	0.80	5.1309	0.81
Sym(7,7) 6 niveaux	-	5.1119	1.13	5.1148	1.13
Approche Intra Sous-Bandes (AISB)	1	5.2071	-0.72	5.2075	-0.67
	2	5.1765	-0.12	5.1772	-0.08
	4	5.1573	0.25	5.1583	0.29
	8	5.1426	0.53	5.1438	0.57
	16	5.1316	0.74	5.1330	0.77
	32	5.1245	0.88	5.1261	0.91
	64	<b>5.1204</b>	<b>0.96</b>	<b>5.1221</b>	<b>0.98</b>

TAB. 6.1 – Résultats de la compression sans perte (bpp) pour l’ensemble des patchs de test, et pour les patchs de test non extraits de la première image qui a été utilisée pour effectuer les apprentissages. Le gain d’espace est également indiqué en comparaison à l’approche ondelettes sur 3 niveaux. La taille de bloc utilisée pour toutes les approches est  $8 \times 8$ . Les meilleurs résultats de chaque approche sont en gras. Les résultats obtenus pour la décomposition en ondelettes sur 6 niveaux sont également donnés pour être mis en confrontation avec l’AISB.

engendre une perte de plus de 10 dB à haut débit, avant le codage du dernier plan de bits.

De même, l'impact de  $K$  diffère légèrement de l'approche irréversible au delà de 1 bpp, mais globalement les choix de  $K$ , pour un bon compromis débit/distorsion aux débits qui semblent intéressants, restent les mêmes :  $K = 8$  pour l'ADD,  $K = 1$  pour l'AG et  $K = 64$  pour l'ASB.

La transformée en ondelettes réversible souffre également du bruit produit par les erreurs d'arrondis (nécessaires à l'implémentation réversible) des étapes de lifting (c.f. FIG. 6.18). Les distorsions engendrées par l'approximation des ondelettes sont à peu près autant pénalisantes que celles générées par l'approximation de la factorisation PLUS pour les KLT. On peut en juger en comparant les résultats de l'ADD, qui n'utilise que la factorisation PLUS, avec ceux de la transformée en symlettes. Le comportement de leur différence de PSNR est relativement similaire à celui obtenu pour le codage irréversible. Cependant, si les ondelettes avaient été choisies avec un support plus compact, le nombre d'étapes de lifting nécessaire à les approximer aurait été moindre et donc les résultats probablement meilleurs. Pour l'AG et l'ASB, sont effectués non seulement une approximation de la transformée en ondelettes mais également une approximation de la factorisation PLUS, qui expliquent les résultats encore moins bons à haut débit.

## 6.5 Améliorations des résultats

Jusqu'à présent, les modèles de compression utilisés pour les différentes représentations multi-KLT se sont uniquement appuyés sur l'algorithme SPIHT de sorte à avoir une base algorithmique commune et ainsi rendre les résultats davantage comparables entre eux et avec les représentations ondelettes. Cependant ces extensions de SPIHT sont loin d'être optimales. Afin de pouvoir les rendre comparables avec les algorithmes développés dans les chapitres précédents ainsi qu'avec des logiciels de référence, les approches basées sur les  $K$ -KLT ont donc bénéficié de quelques améliorations. Dans cette section, un modèle un peu plus intéressant est utilisé pour améliorer la déquantification au décodage, et les algorithmes de compression sont optimisés en exploitant le codage arithmétique.

### 6.5.1 Déquantification

Dans JPEG-2000, lors de la reconstruction d'une image avec pertes (couche de qualité, ou compression avec pertes), si un coefficient quantifié avec un pas de quantification  $\Delta_b$  est décodé comme ayant une amplitude  $v_b$ , un signe  $\chi_b$ , la valeur du coefficient déquantifié  $y_b$  vaut [TM01]-10.5 :

$$y_b = \begin{cases} 0 & \text{si } v_b = 0 \\ \chi_b (v_b + 2^{p_b} \delta) \Delta_b & \text{si } v_b \neq 0 \end{cases}, \quad (6.11)$$

avec  $p_b$  correspondant au nombre de bits de  $v_b$  non décodés à cause de la troncature du flux.



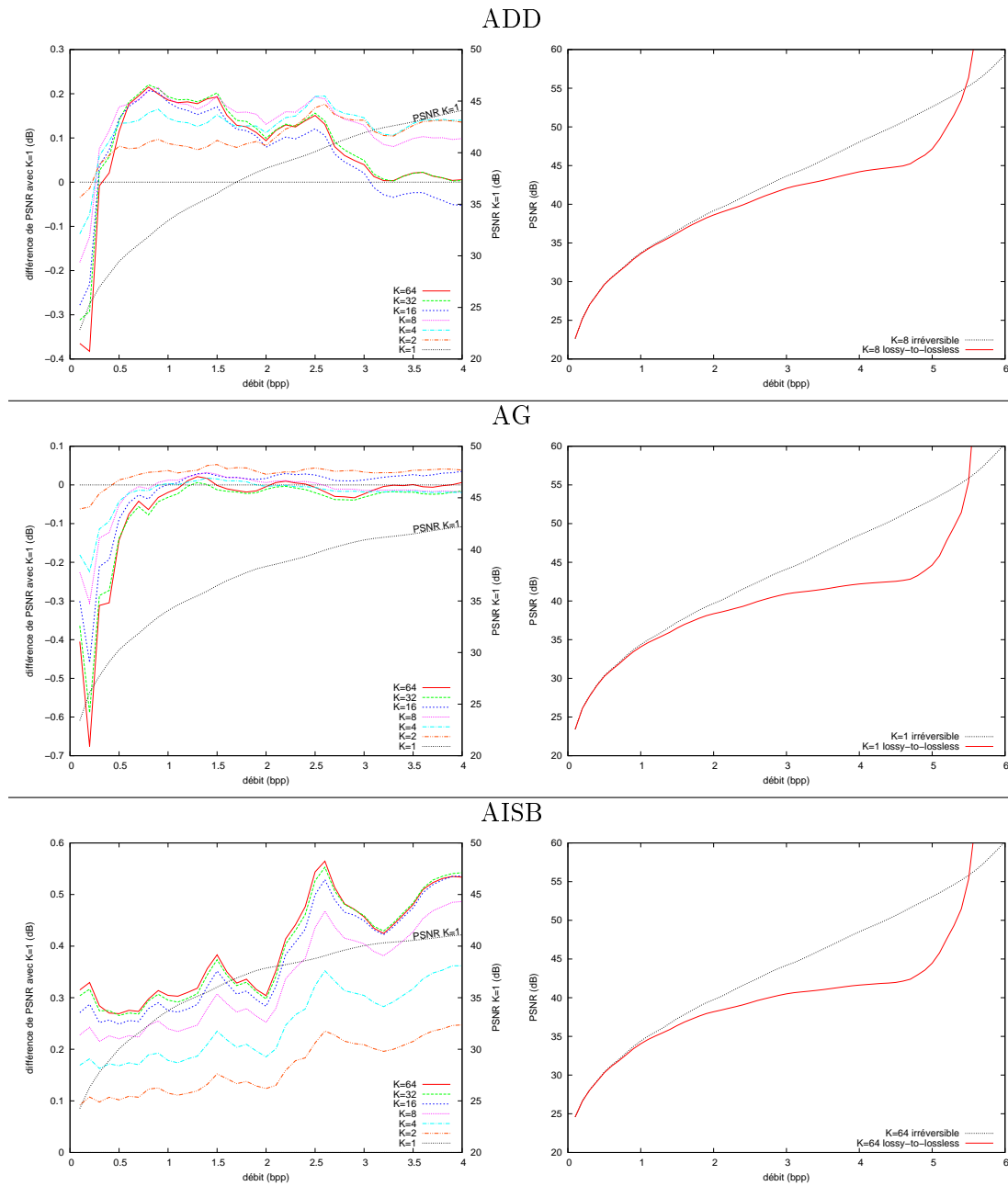


FIG. 6.17 – Différence de PSNR entre les apprentissages mono et multi KLT en mode réversible (à gauche), et comparaison entre les modes réversible et irréversible (à droite). Les résultats sont obtenus pour l'ensemble des 120 échantillons anatomopathologiques.

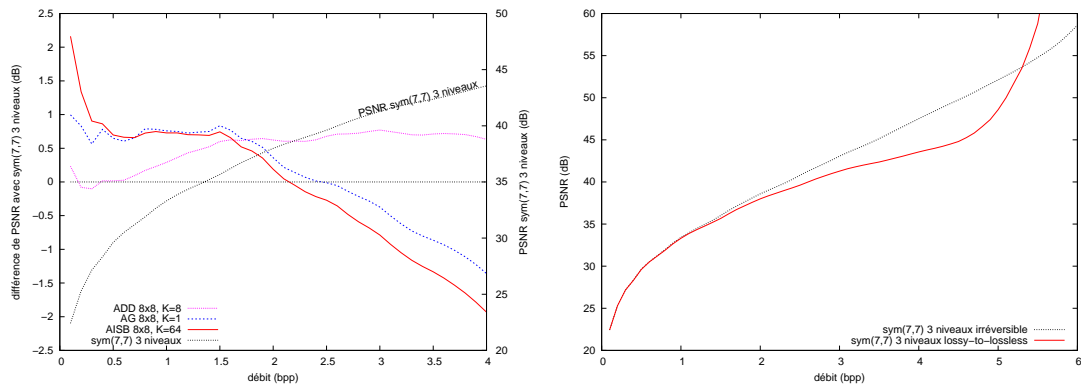


FIG. 6.18 – Différence de PSNR entre les apprentissages multi KLT et la transformée en ondelettes en mode réversible (à gauche), et comparaison entre les transformées en ondelettes réversible et irréversible (à droite). Les résultats sont obtenus pour l'ensemble des 120 échantillons de test.

La politique du choix de  $\delta$  pour la déquantification, permettant d'approximer le centroïde de la distribution, est laissée au décodeur, mais il est recommandé d'utiliser simplement  $\delta = 1/2$ . Ce choix de  $\delta$  est également celui qui a été utilisé avec les décompositions adaptatives en ondelettes (ODC) du chapitre précédent.

Dans les sections précédentes, en régime progressif, le décodage considérait que les coefficients à utiliser pour la reconstruction avaient pour valeurs celles qui étaient décodées lorsque la fin du flux de données tronqué était atteinte (i.e.  $\delta = 0$ ). Cependant, comme il vient d'être mentionné, cette valeur n'est pas optimale si l'on considère le modèle de quantification/déquantification induit par la troncation du flux. Pour les résultats qui vont suivre,  $\delta = 1/2$  sera utilisé, comme recommandé pour JPEG-2000. Cette valeur de  $\delta$  permet d'améliorer le PSNR de manière intéressante, comme illustré dans la FIG. 6.19-(a) pour la compression irréversible. Bien qu'il ne soit pas cherché à estimer un  $\delta$  optimal (qui dépendrait de  $p_b$  et éventuellement de  $v_b$ ), le PSNR est ainsi amélioré de presque 0.5 dB à bas débit (0.1 bpp) et jusqu'à plus de 2.5 dB à haut débit (4 bpp). Les améliorations sont très similaires, et ce, quelle que soit l'approche multi-KLT considérée.

### 6.5.2 Codage arithmétique

Dans les sections précédentes, afin d'éviter les ambiguïtés de sorte à ce que les différentes approches soient davantage comparables entre-elles, le codage de l'information n'exploitait pas certaines redondances statistiques. Afin d'améliorer les résultats de compression, les codeurs ont donc été étendus pour utiliser un codage arithmétique.

Dans le modèle utilisé (c.f. 6.3.1), l'un des points les plus critiques lors du codage d'un

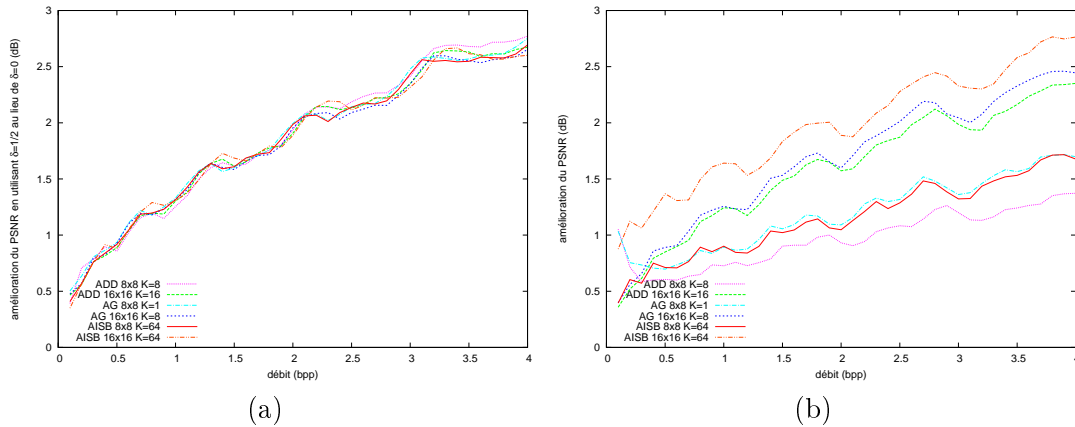


FIG. 6.19 – Gains sur le PSNR liés à (a) l'utilisation de  $\delta = 1/2$  au lieu de  $\delta = 0$ , (b) l'utilisation du codage arithmétique lorsque  $\delta = 1/2$ . Résultats obtenus en mode de compression irréversible sur l'ensemble des patches de test.

vecteur transformé par KLT est la signalisation de son premier coefficient significatif. En effet, avec les propriétés de la KLT, si un  $K_i$ SET devient significatif, il y a une très forte probabilité pour que ce soit son premier coefficient (le  $i$ -ième du vecteur transformé) qui devienne significatif. Ainsi l'utilisation d'un bit pour signaler cette information est statistiquement trop important. Afin de remédier à ce problème, une table  $L(j)$  indexée pour  $j = 1..D$  des statistiques de la signalisation de la significiance du  $j$ -ième coefficient est utilisée pour guider son codage arithmétique binaire. Pour l'approche intra-sous-bande, chaque sous-bande se voit associer une table indépendante. Cette technique simple est la principale source des gains de codage qui vont être obtenus.

La seconde source de gain de compression va au codage entropique de la signalisation de la significiance des coefficients contenus dans la LIP. Celle-ci est réalisée à l'aide d'un codeur arithmétique binaire contextuel qui choisit un contexte de codage en fonction du nombre de blocs contigus (0 à 8) ayant un coefficient de même indice déjà significatif. Enfin, le codage de la significiance des différents MaSET (précédemment spécifiés pour chacune des approches dans la section 6.3) est également codée arithmétiquement et engendre de faibles gains de compression.

Chaque table statistique guidant le codage est mise à jour après toute nouvelle signalisation l'utilisant, et est réinitialisée avant le codage de chaque nouveau plan de bit.

Les autres informations (index des bases de décomposition, signalisation du signe des coefficients et raffinement des coefficients déjà significatifs) sont considérées comme quasiment équiprobables, et le codage arithmétique n'est donc pas exploité.

Les améliorations du PSNR apportées par l'utilisation du codage arithmétique sont montrées dans la FIG. 6.19-(b). Le modèle de compression probabiliste permet de gagner entre 0.3 dB et 1.1 dB à 0.1 bpp et entre 1.8 et 2.7 dB à 4bpp selon l'approche.

Pour l'ensemble des algorithmes, la première source de gains est le modèle de codage arithmétique pour la signifiante dans un  $K_i$ SET. Cependant, lorsque des blocs  $16 \times 16$  sont utilisés, le nombre de bases étant assez restreint par rapport à leurs dimensions, il arrive assez souvent que des coefficients non significatifs apparaissent (les  $y_{\{i+s < i+S\}}$  dans 6.3.1) et soient placés dans la LIP. Le codage entropique de la signifiante des éléments de la LIP a donc un impact plus important en grande dimension et explique les gains plus élevés pour les blocs  $16 \times 16$ . Ceci est encore plus vrai pour le codage des KLT sur les sous-bandes d'ondelettes hautes fréquences de l'ASB. En effet, pour celles-ci, l'information est assez éparse et donc difficile à localiser dans un gros bloc, tandis que pour les KLT de la sous-bande basse résolution ou pour celles de l'ADD les relations géométriques sont assez bien exploitées pour extraire l'information principale. De même, les KLT de l'AG semblent (c.f. FIG. 6.4 et FIG. 6.5) s'appuyer assez efficacement sur les relations hiérarchiques (inter et intra-résolution). C'est pourquoi, sur l'ensemble des résultats, le codage arithmétique favorise davantage l'ASB  $16 \times 16$ , en particulier à bas débit car à haut débit la localisation de l'information restante (« les détails ») devient également plus difficile pour les autres représentations. La qualité de l'apprentissage peut également jouer un rôle, en provoquant une plus grosse confusion sur l'ordonnement et le contenu des derniers vecteurs propres de la KLT si les données d'apprentissage ne sont pas suffisamment représentatives des statistiques générales de la classe d'image à traiter.

Les améliorations plus importantes pour l'ADD et l'AG  $8 \times 8$  à bas débit non seulement du codage arithmétique de la signifiante des pixels de la LIP, mais également de celui de la signifiante des ensembles de la LIS (MaSET). En effet pour ces deux représentations, le nombre d'éléments se trouvant dans chacune de ces listes est plus important que pour les autres représentations (quatre fois plus en comparaison à l'ADD ou à l'AG  $16 \times 16$ , davantage en comparaison à l'ASB). Or, à bas débits, ceux-ci sont rarement significatifs et il existe donc, sans le codage arithmétique, une redondance sur les occurrences de bits nuls pour signaler l'insignifiante de ces éléments qui est également plus importante que pour les autres représentations.

### 6.5.3 Comparaisons expérimentales sur les patches des lames virtuelles

Les codecs pour les modèles multi-KLT ayant été un peu optimisés, leurs résultats vont pouvoir être comparés aux autres algorithmes introduits aux chapitres 4 et 5 ainsi qu'aux principales références en compression irréversible et réversible.

#### 6.5.3.1 Compression scalable irréversible des patches de test

Dans la FIG. 6.20, les résultats débit/PSNR obtenus, sur l'ensemble des patches de test, par le logiciel de référence de l'algorithme SPIHT (exploitant le codage arithmétique) pour la compression irréversible<sup>1</sup> sont comparés à ceux obtenus pour l'ADD,

<sup>1</sup>`codetree` et `decdtree` disponibles à l'adresse <http://www.cipr.rpi.edu/research/SPIHT/spiht3.html>, version utilisée : `SPIHT_linux_3.2.2.tar.gz`

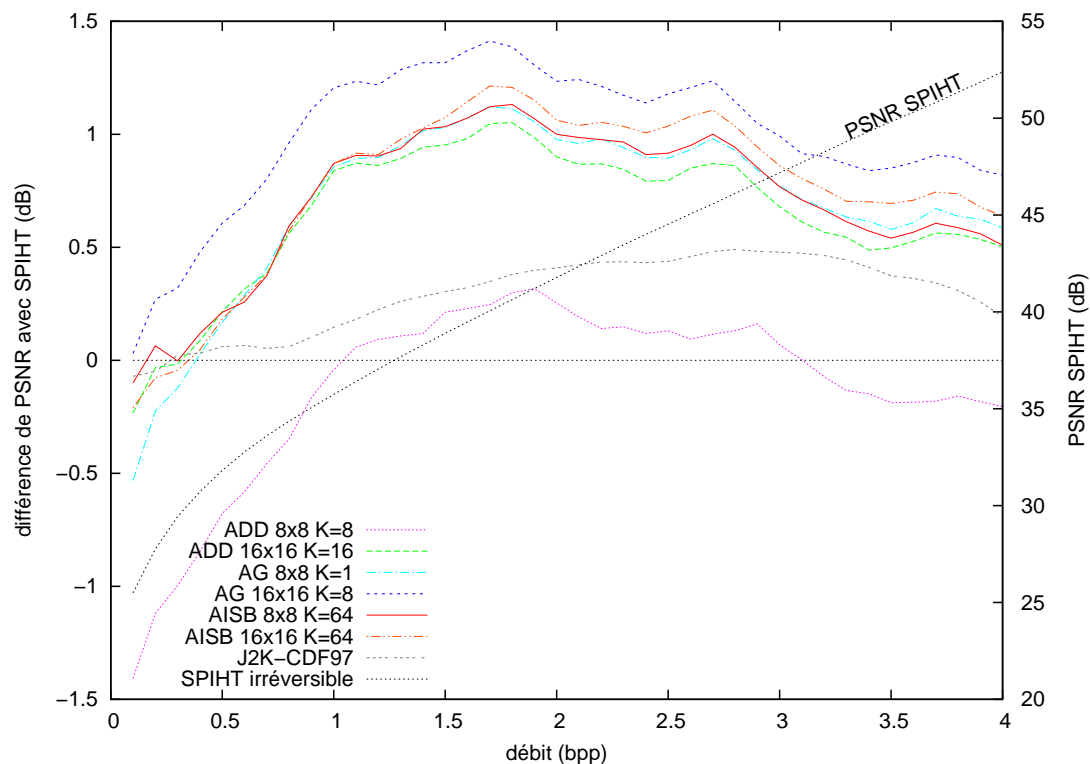


FIG. 6.20 – Différence de PSNR avec le logiciel de référence de l'algorithme SPIHT pour la compression irréversible.

l'AG et l'AISB ainsi qu'à ceux de la version 6.0 du logiciel kakadu<sup>2</sup> pour JPEG-2000 (J2K-CDF97).

Après amélioration des codecs pour les  $K$ -KLT, il peut être constaté que l'utilisation de blocs  $16 \times 16$  n'est plus pénalisante et devient même plutôt avantageuse. Il peut être clairement observé qu'une taille de bloc  $8 \times 8$  dans le domaine image n'est pas adaptée pour la haute définition les lames virtuelles étudiées. Ainsi l'ADD perd quasiment 1.2 dB à 0.1 bpp en comparaison à l'utilisation de blocs  $16 \times 16$ . L'AG est moins pénalisée, avec une différence plus faible de 0.5 dB à bas débit. Celle-ci peut s'expliquer par l'utilisation des ondelettes dont le support permet de prendre en considération une partie de l'information adjacente au bloc considéré et ainsi de compenser en partie la faible résolution spatiale.

Sans avoir tous les détails sur l'implémentation du logiciel de référence de SPIHT, il doit certainement utiliser une décomposition dyadique sur 4 ou 5 niveaux, comme JPEG-2000. Ceci permet alors d'expliquer les moins bons résultats obtenus par l'AG et l'ADD  $8 \times 8$  à bas débit.

<sup>2</sup>kakadu est disponible à l'adresse <http://www.kakadusoftware.com/>

Même si à très bas débit on ne note pas de grandes différences entre les autres algorithmes, pour les débits qui peuvent commencer à proposer une qualité visuelle intéressante (environ 0.4, 0.5 bpp) on peut noter l'apport de l'utilisation des KLT. Entre 0.5 et 1 bpp, l'ADD  $16 \times 16$ , l'AG  $8 \times 8$ , et les AISB ont des résultats équivalents en terme de PSNR, et permettent d'améliorer les résultats de 0.1 à 0.8 dB environ en comparaison à SPIHT. Au delà, leurs résultats se détachent légèrement, avec un petit mieux pour l'AISB  $16 \times 16$ , et permettent de dépasser les 1 dB d'amélioration à 1.5 bpp avant de progressivement redescendre jusqu'à 0.5 dB à 4bpp.

L'AG, qui, de par sa conception, peut sembler plus optimale puisque permettant de prendre en considération les corrélations des coefficients d'ondelettes dans l'ensemble des sous-bandes, fournit effectivement, pour les tailles de blocs  $16 \times 16$ , les meilleurs résultats pour tous les débits entre 0.1 et 4 bpp. A 0.5 bpp elle obtient déjà un PSNR supérieur à SPIHT d'environ 0.6 dB (soit 0.5 dB de mieux que les autres représentations multi-KLT) et à 1 bpp elle atteint les 1.2 dB d'amélioration et monte jusqu'à environ 1.4 dB à 1.7 bpp.

Notons que JPEG-2000 n'a pas été pris en considération jusqu'ici car le modèle de codage qu'il utilise ne permet pas de générer un flux scalable en qualité aussi efficace que SPIHT, pour tout point de troncature arbitrairement choisi. Cependant JPEG-2000 rend possible l'optimisation du codage pour des points de troncatures spécifiques (à des débits donnés), préalablement choisis, mais nécessite le codage d'informations supplémentaires si plusieurs débits cibles intermédiaires sont envisagés. La courbe pour JPEG-2000 dans la FIG. 6.20 correspond alors à chaque résultat de la compression irréversible obtenu spécifiquement pour le débit associé (et favorise donc les résultats en comparaison à SPIHT). Cependant, malgré cette optimisation, au delà de 0.4 bpp J2K ne propose pas de meilleurs résultats que les modèles multi-KLT (hormis l'ADD  $8 \times 8$ ) et n'améliore les résultats de SPIHT que de 0.2 dB à 1 bpp et d'à peine 0.3 dB à 1.5 bpp. L'approche par KLT reste donc plus avantageuse en terme de PSNR.

Au final, les approches par KLT et en particulier l'AG sont particulièrement compétitives en terme de compression irréversible, même si le codage pourrait certainement bénéficier d'autres améliorations. Un algorithme de type EBCOT pourrait également être utilisé afin d'optimiser encore davantage la qualité pour un débit cible. Des résultats visuels obtenus après amélioration du codec sont disponibles dans les FIG. 6.21 et FIG. 6.22

### 6.5.3.2 Compression scalable réversible des patches de test

Dans la FIG. 6.23, les résultats débit/PSNR obtenus, sur l'ensemble des patches de test, par le logiciel de référence de l'algorithme SPIHT (exploitant le codage arithmétique et la transformée en ondelettes S+P) pour la compression réversible<sup>3</sup> sont comparés à ceux obtenus pour l'ADD, l'AG et l'AISB ainsi qu'à ceux de la version 6.0

<sup>3</sup>progcode et progdecdd disponibles à l'adresse <http://www.cipr.rpi.edu/research/SPIHT/spiht3.html>, version utilisée : SPIHT\_linux\_3.2.2.tar.gz

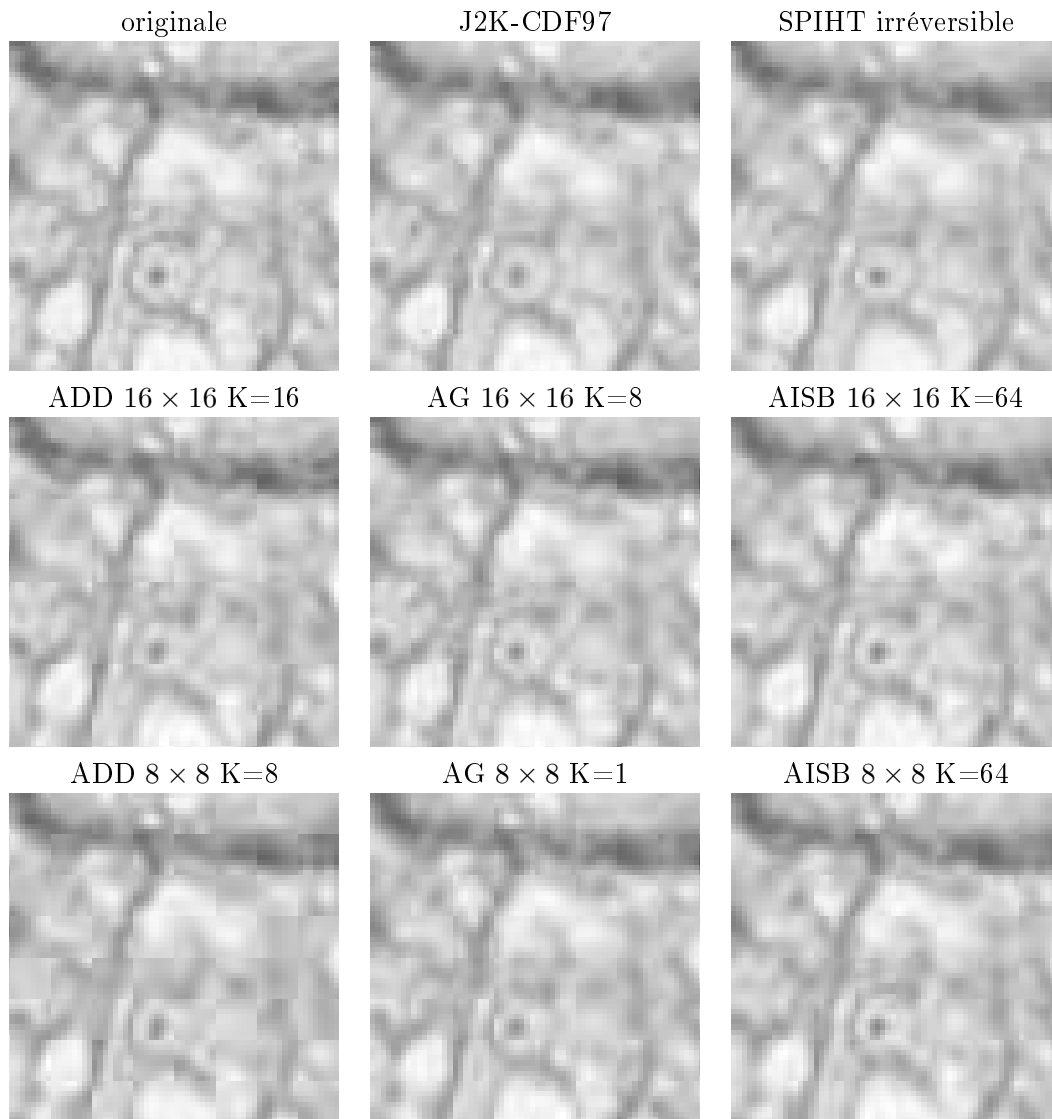


FIG. 6.21 – Exemple visuel (pour affichage sur écran) des distorsions pour une compression irréversible à 0.5bpp utilisant le codage arithmétique et  $\delta = 1/2$ , blocs de taille  $64 \times 64$  extraits du patch 05-01.

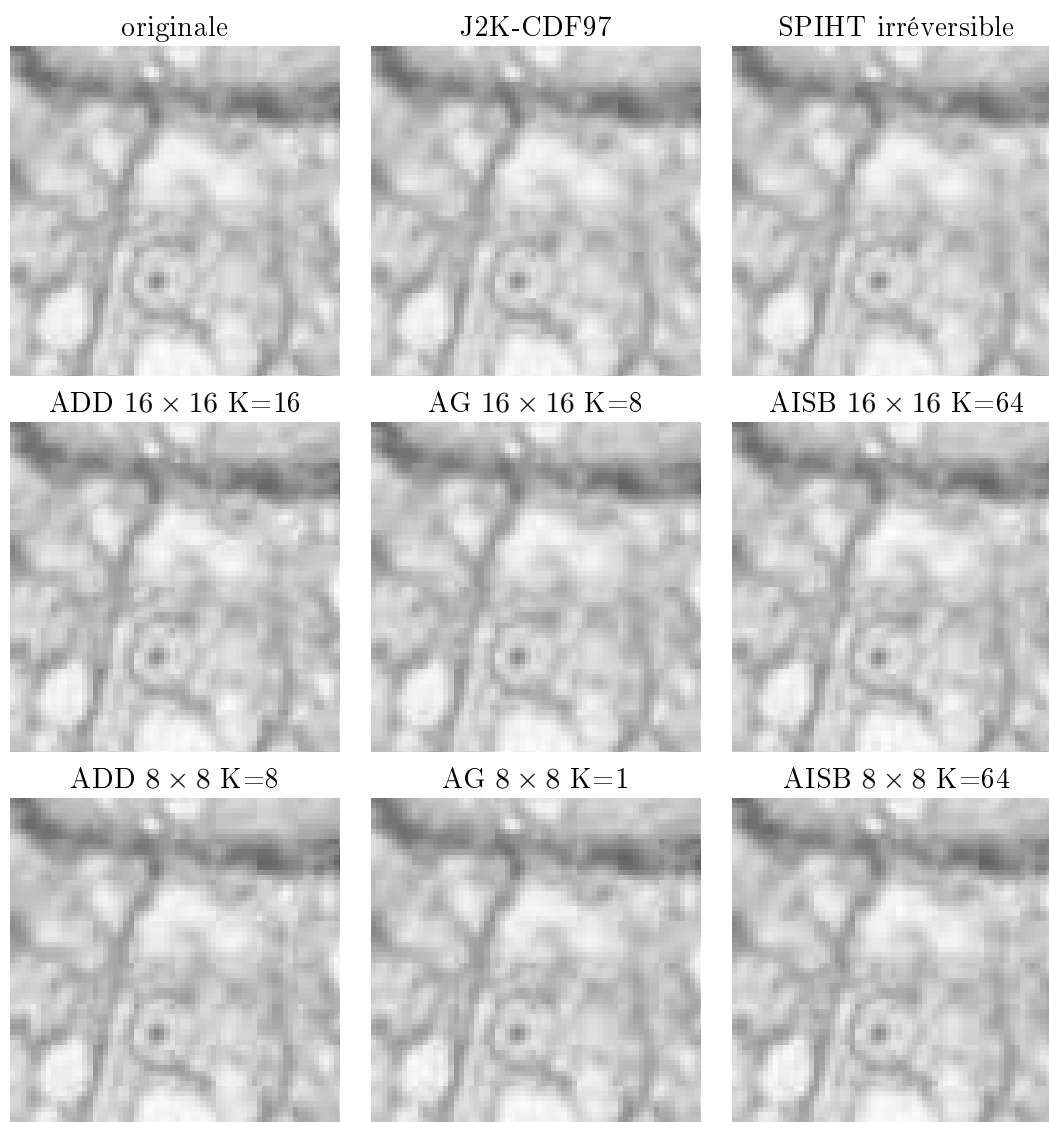


FIG. 6.22 – Exemple visuel (pour affichage sur écran) des distorsions pour une compression irréversible à 1bpp utilisant le codage arithmétique et  $\delta = 1/2$ , blocs de taille  $64 \times 64$  extraits du patch 05-01.



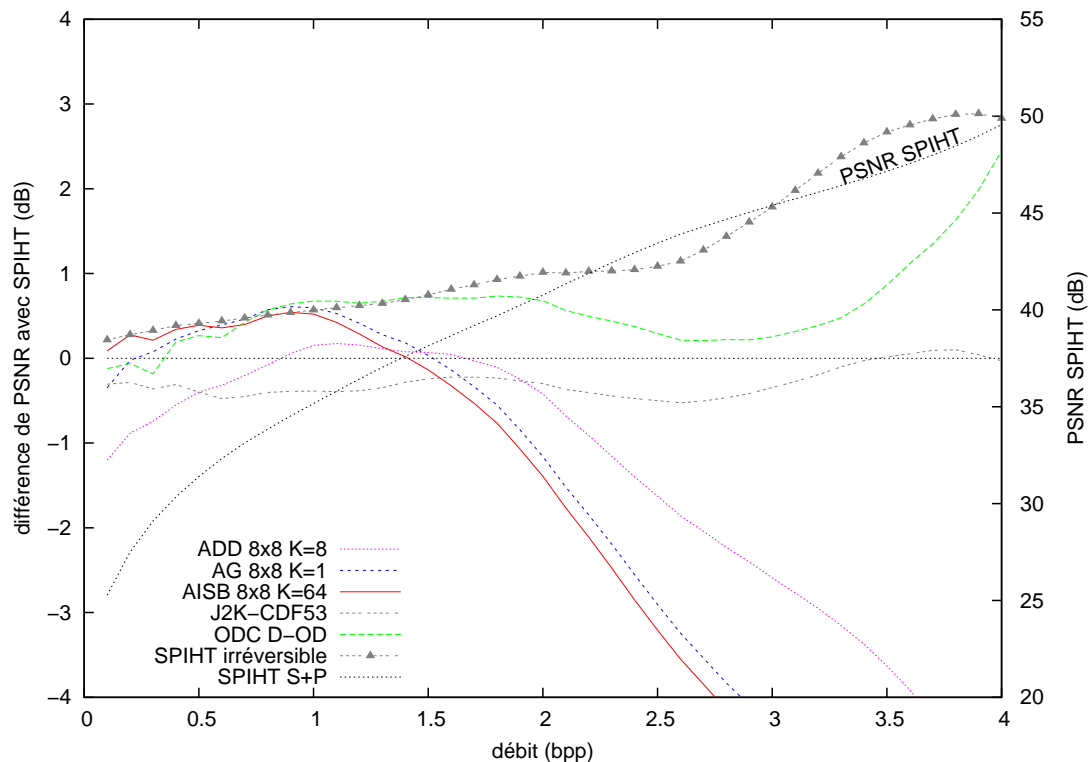


FIG. 6.23 – Différence de PSNR avec le logiciel de référence de l'algorithme SPIHT pour la compression sans perte (utilisant la transformée en ondelettes S+P).

du logiciel kakadu<sup>4</sup> pour JPEG-2000 (J2K-CDF53). Les résultats de SPIHT dans sa version irréversible sont également inclus afin de pouvoir comparer les résultats à ceux de la FIG. 6.20. Les résultats de la scalabilité obtenue à l'aide de l'optimisation de la décomposition en paquets d'ondelettes pour la compression sans perte vue au chapitre 5 utilisant les filtres de Deslauriers et Dubuc (ODC D-OD) y sont également comparés.

A bas débit, les résultats de l'AG et de l'AISB permettent une amélioration allant jusqu'à 0.5 dB à 1 bpp, débit jusqu'auquel l'ADD atteint des performances à peu près équivalentes à celles de SPIHT en mode irréversible. Cette amélioration s'atténue au delà, et à partir de 1.4, 1.5 bpp le bruit dû au lifting fait chuter les performances en dessous de celles de SPIHT réversible. En comparaison à sa version irréversible, SPIHT perd de manière quasiment linéaire de 0.2 dB à 0.1 bpp jusqu'à 1 dB à 2.5 bpp, puis davantage à cause de l'approximation réversible. Cependant cette perte semble assez convenable en comparaison à celles bien supérieures que peut obtenir JPEG-2000 en comparaison à son mode irréversible.

**En mode réversible, en comparaison à SPIHT et JPEG-2000, les résultats**

<sup>4</sup>kakadu est disponible à l'adresse <http://www.kakadusoftware.com/>

		Algorithmes de référence			
		J2K	4.7769		
		JLS	4.7061		
		SPIHT	4.6358		
		CALIC	4.5140		
<hr/>					
Prédiction Hiérarchique chapitre 4		Décomposition Optimisée chapitre 5		Apprentissage multi-KLT section 6.5	
HOP	4.6861	5/3-dy	4.5886	ADD 8×8 K=8	4.8179
HOP-LSE	4.4873	5/3-OD	4.5837	AISB 8×8 K=64	4.7446
HOP-LSE <sup>+</sup>	4.3620	L-dy	4.4266	AG 8×8 K=1	4.7343
		L-OD	4.4259		
		D-dy	4.4257		
		D-OD	4.4250		

TAB. 6.2 – Débits moyens (bpp) obtenus sur l'ensemble des patches de test lors de la compression sans perte à l'aide des principaux algorithmes proposés.

**en progressivité de l'AG et l' AISB sont donc plutôt satisfaisants jusqu'à 1 bpp environ. Cependant, l'optimisation des bases d'ondelettes (ODC), du chapitre précédent, apparait comme étant plus avantageuse en scalabilité réversible (grâce à un nombre d'étapes de lifting assez restreint), et obtient des résultats équivalentes à SPIHT-irréversible jusqu'à 1.5 bpp, et restant toujours plus performante que SPIHT-réversible au delà de 0.4 bpp.**

Il semble donc que si la qualité des images restituées pour des débits compris entre 0.5 et 1 bpp est suffisante pour la navigation, l'ADD et l'AG peuvent être compétitives pour une représentation scalable. Cependant, si le débit nécessaire est supérieur leur intérêt sera moindre. ODC se montre tout de même plus adapté pour la compression scalable réversible : en plus de fournir une scalabilité plus intéressante, il permet également un stockage des images sans perte plus économique à celui des approches par  $K$ -KLT (avec plus de 6% de gain d'espace). Ce chiffre est illustré dans la TAB. 6.5.3.2 au travers des résultats obtenus par les différents algorithmes de compression sans perte sur l'ensemble des patches de test.

Les résultats de cette TAB. 6.5.3.2 confirment également que ces images de haute définition sont assez douces puisqu'une taille de support de décorrélation trop court tend à pénaliser la compression. En effet, les faibles performances des prédicteurs de HOP sont largement contournées par l'optimisation des prédicteurs sur un support plus étendu (HOP-LSE et HOP-LSE<sup>+</sup>); et pour ODC c'est l'optimisation de la taille du support qui introduit le plus visiblement un gain de codage.

## Conclusion et perspectives

Dans ce chapitre a été introduit un algorithme de type  $K$ -Means pour l'optimisation multi-classe de bases de décomposition orthonormée : les  $K$ -KLT. Cette optimisation est effectuée dans le but de spécialiser un modèle pour des images ayant des caractéristiques similaires (modalité de contenu), afin d'améliorer leur compression dans un schéma scalable en qualité.

Trois approches pour la modélisation par  $K$ -KLT ont été considérées. La première est une approche classique dans le domaine direct (ADD) pour la décorrélation de blocs de pixels, et les deux autres s'appuient sur une transformée en ondelettes préalable afin de supprimer les artefacts en blocs causés par l'ADD, et sont donc de type post-transformées. L'approche globale (AG) tente d'exploiter la corrélation des coefficients d'ondelettes au travers des différentes sous-bandes pour une certaine zone spatiale et peut être vue comme une *overlapped orthonormal transform* optimisée. L'approche intra sous-bande (AISB) suit un modèle plus pratique à mettre en place, qui consiste à considérer indépendamment chacune des sous-bandes, et à faire un apprentissage différent pour chaque sous-bande afin de décorrélérer les coefficients par blocs. Chacune des approches dans le domaine transformé peut avoir ses avantages. La première peut tirer parti des relations hiérarchiques des coefficients d'ondelettes. Et la seconde permet de spécialiser chacun des apprentissages et de restreindre l'information structurelle aux orientations captées dans la sous-bande.

L'algorithme SPIHT a été étendu pour chacune des modélisations afin d'effectuer une compression scalable en qualité et d'avoir un socle commun pour pouvoir comparer les différentes approches. Le codeur a ensuite été optimisé en intégrant l'utilisation du codage arithmétique, et le décodeur amélioré en estimant un centroïde de déquantification dans le but de rendre le codec plus comparable aux autres algorithmes, tels que JPEG-2000. Les résultats en compression irréversible scalable sont intéressants. L'AG permet, par exemple, d'améliorer de plus de 1 dB le PSNR en comparaison à JPEG-2000 pour un débit de 1 bpp. Pour rendre la compression réversible une factorisation en étape de lifting a également été utilisée, mais les erreurs d'approximations liées à cette factorisation rendent l'approche moins avantageuse que l'optimisation en paquets d'ondelettes du chapitre 5.

Les approches présentées pourraient être étendues aux images couleurs ou multispectrales en intégrant directement les informations supplémentaires aux données à transformer pour chercher à être plus optimal. Cependant, comme pour l'augmentation du nombre de niveaux de décomposition en ondelettes, la dimension du problème risque de devenir rapidement trop importante. Et, à moins d'utiliser une parallélisation des calculs pour l'apprentissage et d'avoir une base d'image suffisamment conséquente pour pouvoir permettre une génération efficace des vecteurs de la KLT, des approches d'apprentissage plus robustes pourraient nécessiter d'être mises en place. Une autre solution serait d'utiliser une modélisation par KLT/transformatées imbriquées, afin de réduire localement la dimension du problème. Un exemple classique, est l'utilisation d'une simple décorrélation inter-canaux d'abord, puis la décorrélation indépendante

de chaque canal.

On pourrait s'intéresser à optimiser davantage l'AISB pour la classe d'images à traiter en sélectionnant le nombre de KLT le mieux adapté pour chacune des sous-bande.

Enfin, il pourrait être intéressant de voir comment réduire le nombre d'étapes de lifting pour tenter d'améliorer les mauvaises performances en compression réversible.

Un outil de navigation qui permettrait d'enregistrer les différentes étapes du parcours effectué par le médecin lors de son diagnostic, et éventuellement des durées d'observations couplées à un système d'*eye tracking*, peut sembler fort intéressant pour ces lames virtuelles de très grandes tailles. En effet, un tel parcours n'est pas exhaustif puisque chaque zone de l'image n'a pas le même intérêt. Une compression post-diagnostique exploitant cette information pour ne conserver que les données pertinentes (observées) de chaque niveau de résolution permettrait de réduire de manière drastique l'archivage des données et pourrait également servir de preuve juridique. Un tel outil pourrait aussi servir à étudier et analyser le comportement des spécialistes afin d'orienter des algorithmes d'aide au diagnostic. Enfin il pourrait avoir un attrait pédagogique, en permettant de cerner des erreurs commises par un étudiant, par exemple.



# Conclusion et perspectives générales

Le contexte général de cette étude a été présenté dans le premier chapitre. Les images médicales, et plus particulièrement certaines modalités d'acquisition dont celles ciblées pour cette thèse (TDM, IRM et les lames virtuelles) sont très volumineuses et nécessitent donc des algorithmes de compression efficaces pour leur stockage et leur transmission. La qualité de restitution des TDM et des IRM utilisées pour le diagnostic étant très importante au regard des radiologues, et devant être nécessairement identique à celle de l'archivage, la compression sans perte est majoritairement utilisée. Cependant, ces images médicales sont bien souvent très bruitées, et il ne faut donc pas espérer des gains d'espace spectaculaires avec une telle contrainte. Comme il a pu être vu dans les chapitres de contributions, seule la compression des images ayant un contenu doux (peu bruité, régulier) bénéficient de réelles améliorations en tirant parti de supports de décorrélation plus larges. Cependant, ce type de contenu est encore peu fréquent. Dans le but de réduire l'espace de stockage, la compression presque sans perte semble être une solution plus satisfaisante, pour respecter les clauses éthiques et juridiques liées au diagnostic médical, que la compression irréversible, encore faut-il convaincre les médecins d'accepter l'utilisation une telle approche. Pour les lames virtuelles la compression avec pertes semble indispensable et surtout dans le cas d'une navigation dans ces grandes images par le biais de réseaux à débit limité.

Dans le but de faciliter l'utilisation des images compressées, les techniques de compression introduites dans ces travaux proposent toutes un codage scalable en résolution et/ou en qualité. Tous les codecs ont été mis en place pour des images en niveaux de gris et pourraient être étendus aux images couleurs ou multi-spectrales. La solution la plus simple serait d'appliquer les algorithmes indépendamment sur chacun des canaux couleurs après les avoir précédemment décorrélés entre-eux (à l'aide d'une transformation de type YCrCb (éventuellement réversible, comme pour JPEG-2000) pour les images couleurs ou une KLT pour les images multispectrales, par exemple. Une autre solution pour les  $K$ -KLT, qui a été mentionnée dans le chapitre 6, serait d'intégrer la décorrélation des canaux directement dans le modèle d'apprentissage.

L'algorithme HOP, qui est introduit au chapitre 4 et est destiné à la compression d'images médicales 2D, a été développé pour prendre en considération la nature bruitée des TDM et des IRM ainsi que leur contenu contrasté. En compression sans perte d'images bruitées il est presque aussi efficace que les approches DPCM de ré-

férence mais offre en plus la scalabilité en résolution. Etant moins performant sur les images douces, des extensions exploitant une optimisation dynamique des prédicteurs au sens des moindres carrés pour un support de prédiction plus important (HOP-LSE et HOP-LSE<sup>+</sup>) ont été proposées et permettent d'améliorer encore davantage les résultats. L'algorithme peut être efficace pour l'archivage sans perte ou presque sans perte et facilite le *quick-view* basse résolution pour accéder rapidement à une image particulière.

Les différentes étapes de l'algorithme peuvent certainement être améliorées davantage, dans le but d'accélérer la vitesse d'exécution et/ou d'augmenter l'efficacité de la compression. L'utilisation d'une grille quinconce pour le sous-échantillonnage pourrait être intéressante afin de contourner la faible efficacité du prédicteur statique vertical ( $o = V$ ) qui est la principale source de pénalisation de HOP (en comparaison à IHINT par exemple) sur les images douces.

Le chapitre 5 a introduit un modèle d'optimisation simple pour construire un arbre de décomposition en paquets d'ondelettes adapté à la compression sans perte d'images 2D ou volumiques. Cette optimisation améliore fortement la compression d'images douces, et permet une compression sans perte efficace. L'optimisation, même si elle possède une complexité raisonnable, peut prendre du temps, en particulier sur les images douces pour lesquelles des tailles de support élevées sont sélectionnées pour les niveaux de décomposition des plus hautes résolutions. Même si une recherche un peu plus rapide de la taille du support optimal devrait pouvoir être mise en place, une solution sous optimale (pour les supports de taille importantes) serait suffisante pour obtenir de bons résultats. Cet algorithme se destine plutôt à de l'archivage, et le modèle basé ondelettes peut offrir une scalabilité en qualité ou en résolution pour faciliter la navigation. L'utilisation d'un algorithme de codage de type EBCOT couplé à un protocole de communication similaire à JPIP permettrait également de l'exploiter pour la navigation dans de grandes images. Comme il a pu être mentionné, le coût de codage du modèle optimisé étant assez faible, les résultats d'une optimisation de la décomposition par macro-blocs pour les grandes images (*tiles* de JPEG-2000) permettrait d'obtenir de meilleurs résultats de compression en s'adaptant à des caractéristiques plus locales de l'image. L'approche offre donc de bonnes perspectives pour la compression scalable (ou avec pertes) et pour la navigation à distance.

Dans le chapitre 6 est proposé un modèle exploitant l'apprentissage afin de spécialiser la compression pour des types d'images possédant des caractéristiques spécifiques (contenu caractéristique, modalité d'acquisition, etc.). Cet apprentissage est effectué selon trois approches différentes, une dans le domaine image et deux sur les coefficients d'ondelettes. Un schéma de compression scalable en qualité, dérivé de l'algorithme SPIHT, permet de montrer que ces approches par apprentissage peuvent être très compétitives pour la compression scalable en qualité irréversible. Leur utilisation pourrait être bénéfique pour des milieux spécialisés traitant un nombre de modalités d'images assez faible. Bien entendu, d'autres codeurs entropiques que SPIHT peuvent être exploités et encore une fois un algorithme de type EBCOT pourrait se révéler avantageux. Cependant, l'AG qui offre les meilleurs résultats n'est pas réellement adapté pour une

représentation scalable en résolution, qui peut s'avérer importante pour la navigation dans les lames virtuelles. L'AISB qui offre tout de même un gain en PSNR important en comparaison à JPEG-2000, ou une approche intra-résolution, serait dans ce cas mieux adaptée.

Les résultats obtenus dans ce chapitre proviennent d'apprentissages basés sur une seule image d'exemple. Si une variété plus importante de données avaient été utilisée, les performances auraient certainement pu être améliorées davantage. Dans le but d'accélérer l'apprentissage, il serait intéressant de tester des méthodes d'estimation des KLT robustes afin de réduire le nombre de données nécessaires (par sélection aléatoire dans la base d'apprentissage par exemple). Une autre approche pouvant être envisagée serait une parallélisation sur GPU.

Concernant la convergence, celle-ci est assurée, mais rien ne prouve qu'un optimum global soit atteint. Il faudrait chercher une méthode permettant de mieux s'en approcher.

Enfin, cette thèse n'a pu qu'effleurer l'introduction de deux aspects permettant d'accroître les performances de compression : d'une part la prise en compte de notions d'objets d'intérêts dans un contexte d'imagerie médicale, d'autre part celle d'une mesure objective de qualité subjective de restitution de telles modalités d'imagerie dans un contexte de diagnostic où la part d'interprétation et d'expertise reste essentielle.



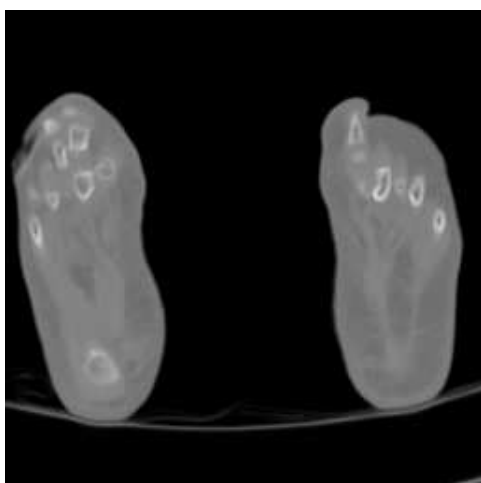


# Annexe A

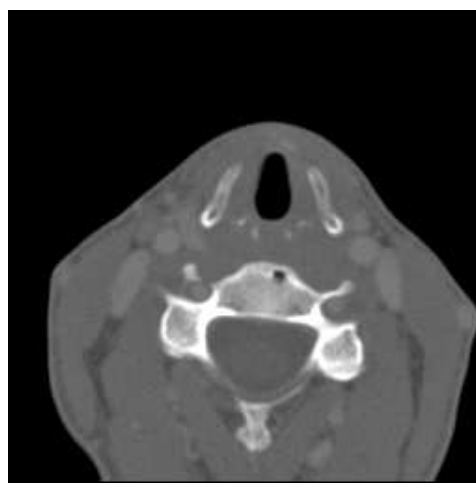
## Images

### A.1 TDM

Exemples d'images TDM. Pour les images 12 bits, la LUT de conversion pour l'affichage en 256 niveaux de gris est linéaire. La luminosité et le contraste sont réglés en fonction du contenu à afficher.



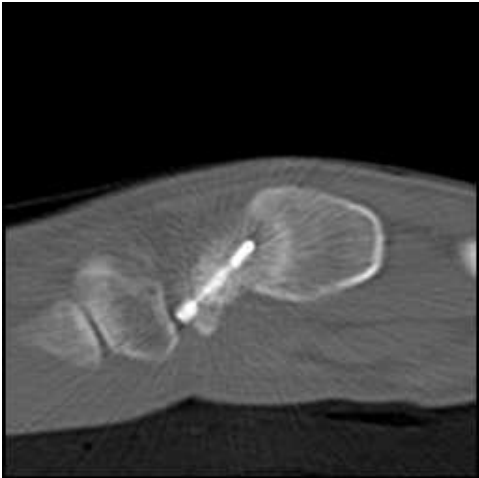
(a) cibr (8 bits) CT\_Aperts 70



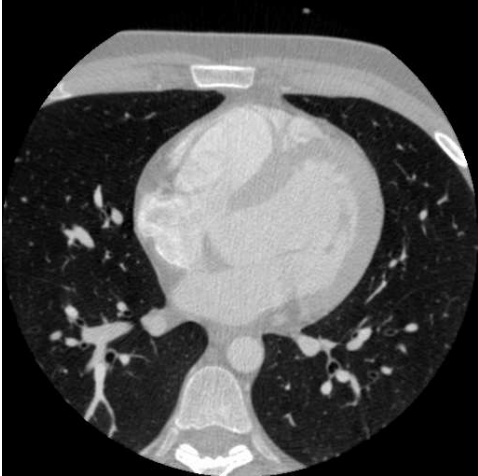
(b) cibr (8 bits) CT\_carotid 45



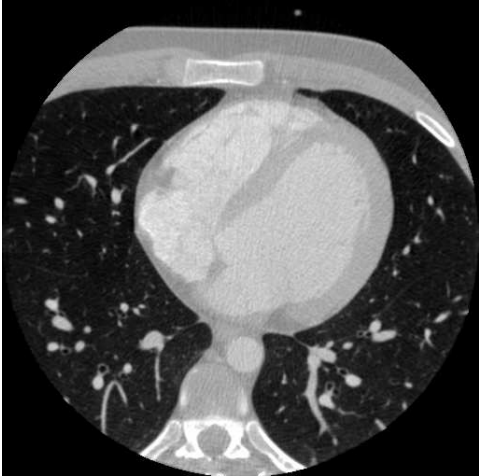
(c) cibr (8 bits) CT\_skull 25



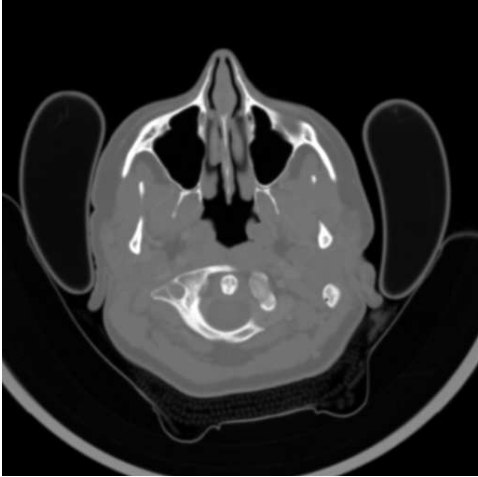
(d) cibr (8 bits) CT\_wrist 35



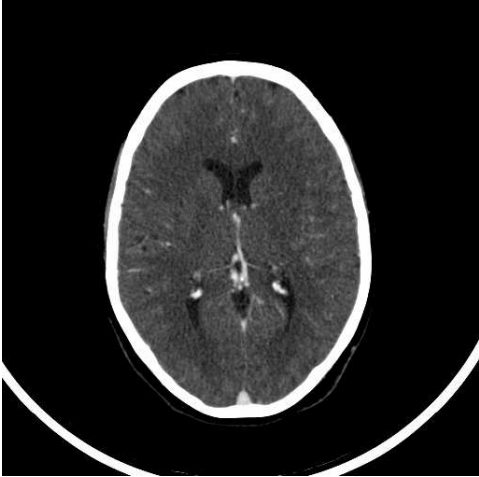
(e) MeDEISA CT\_datat\_1 91



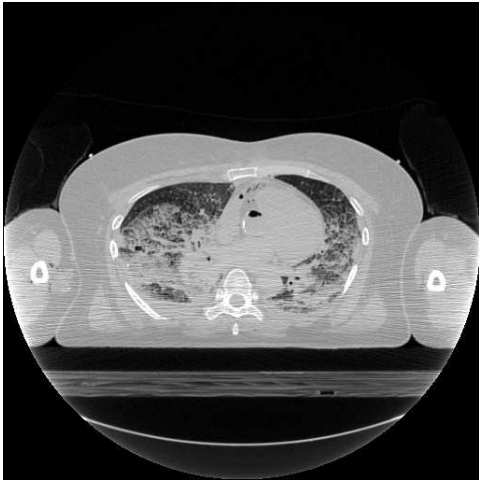
(f) MeDEISA CT\_datat\_1 105



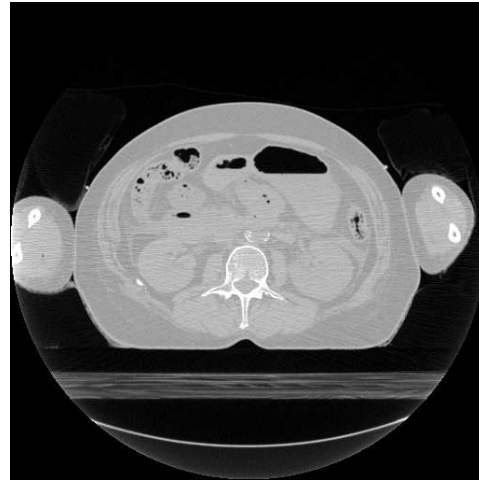
(g) PMR CT AVEC\_IV 30



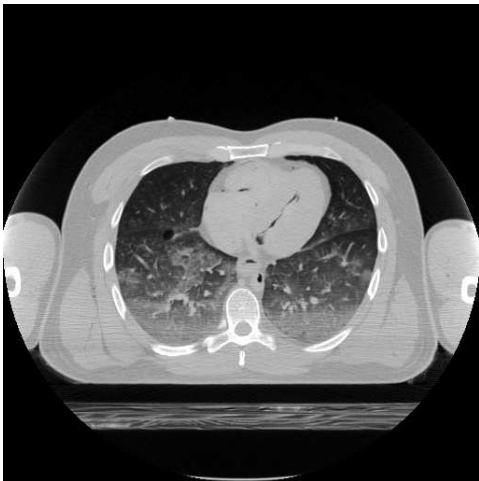
(h) PMR CT AVEC\_IV 118



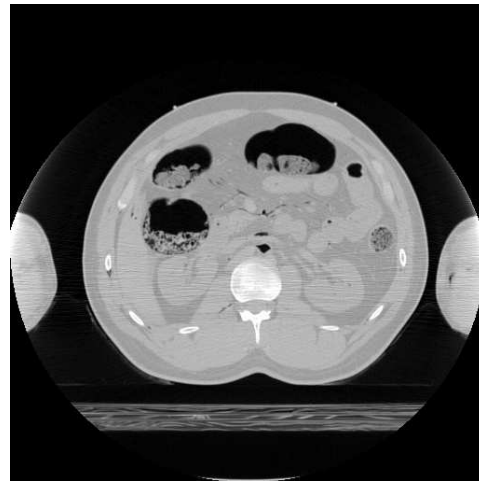
(i) VHP-Female normalCT 400



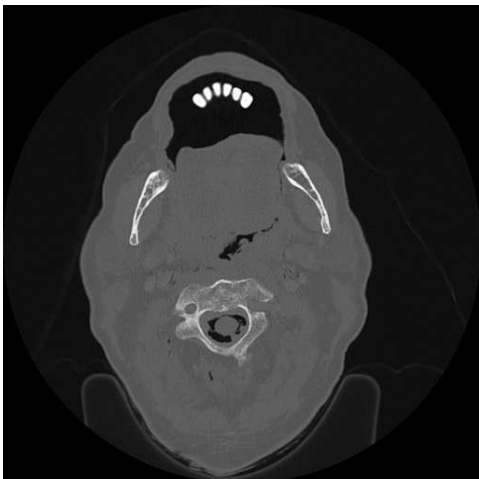
(j) VHP-Female normalCT 624



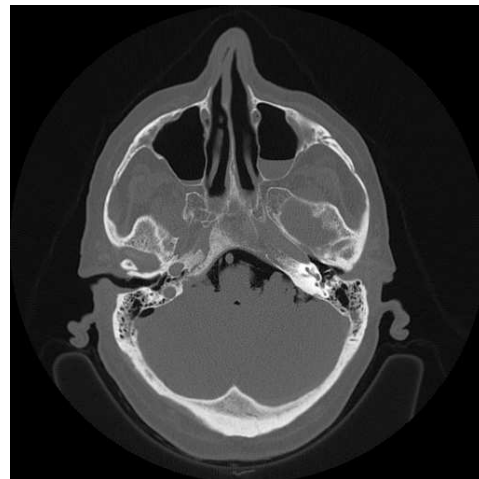
(k) VHP-Male normalCT 248



(l) VHP-Male normalCT 300



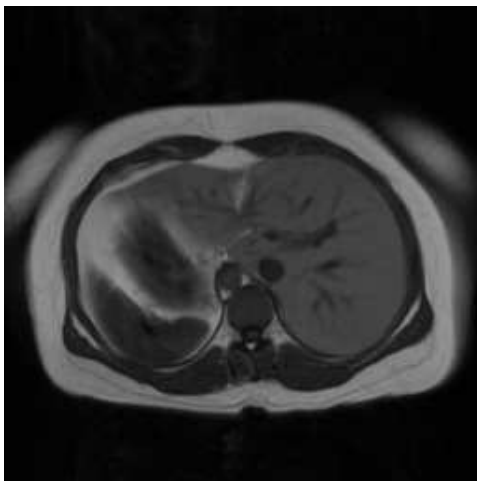
(m) VHP-Harvard CAT 111



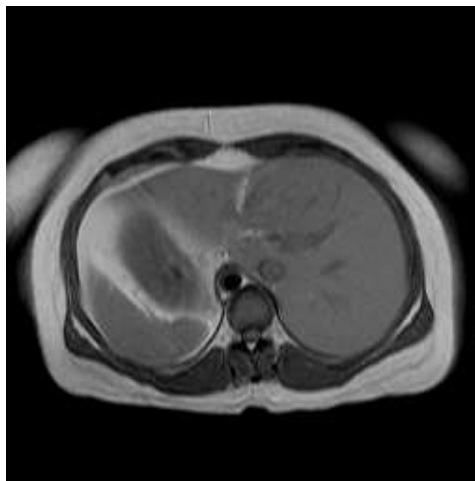
(n) VHP-Harvard CAT 212

## A.2 IRM

Exemples d'IRM. Pour les images 12 bits, la LUT de conversion pour l'affichage en 256 niveaux de gris est linéaire. La luminosité et le contraste sont réglés en fonction du contenu à afficher.



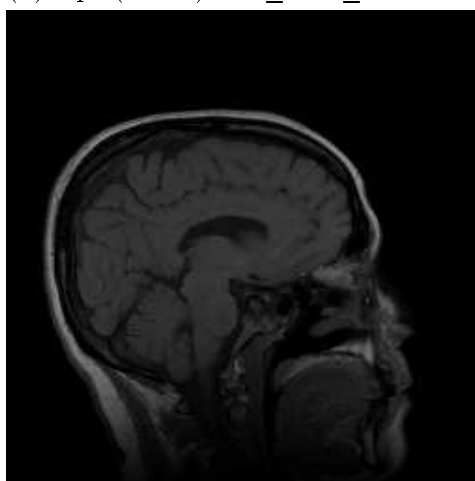
(a) cpr (8 bits) MR\_liver\_t1 40



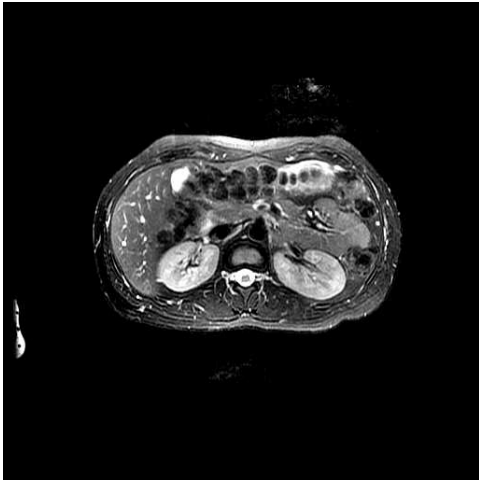
(b) cpr (8 bits) MR\_liver\_t2e1 40



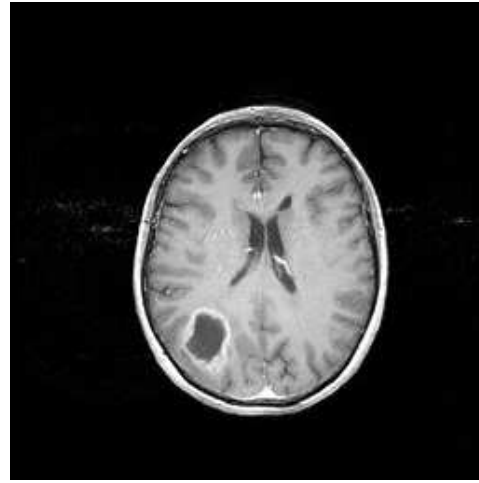
(c) cpr (8 bits) MR\_ped\_chest 40



(d) cpr (8 bits) MR\_sag\_head 27



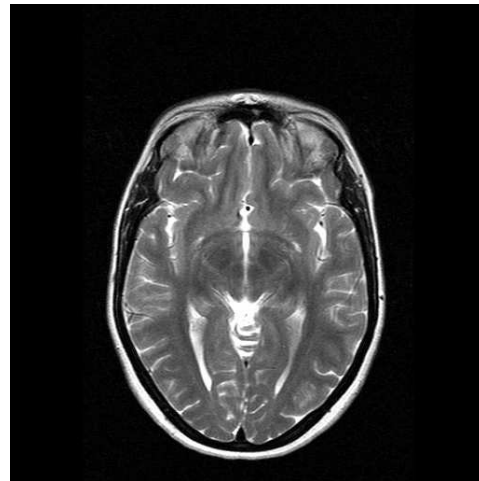
(e) MeDEISA MR2DLiver1 15



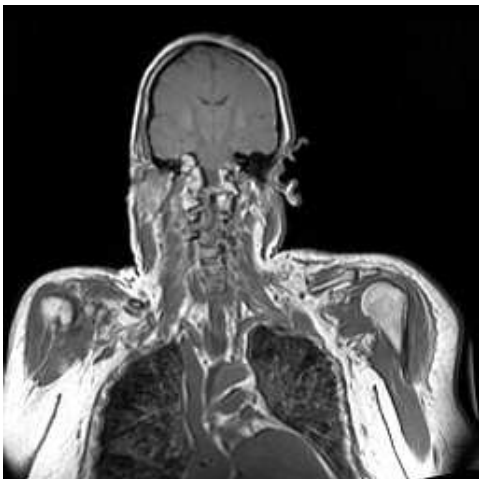
(f) MeDEISA MR3DBrain5 75



(g) PMR MRI-01 11



(h) PMR MRI-02 12



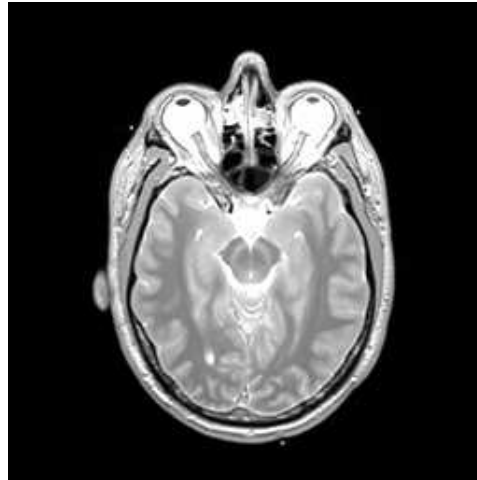
(i) VHP-Female mri-t1-3 30



(j) VHP-Female mri-pd-6 16



(k) VHP-Female mri-t2-7 20



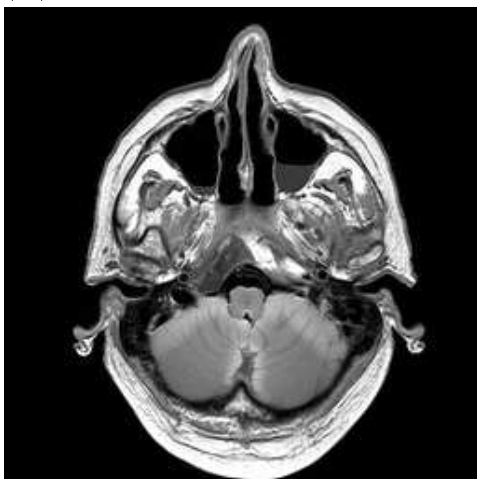
(l) VHP-Male mri-pd-1 21



(m) VHP-Male mri-t2-4 32



(n) VHP-Male mri-t1-5 32



(o) VHP-Harvard MRI-T1 32



(p) VHP-Harvard-3D MRI-T2\_512 115

### A.3 Lames virtuelles

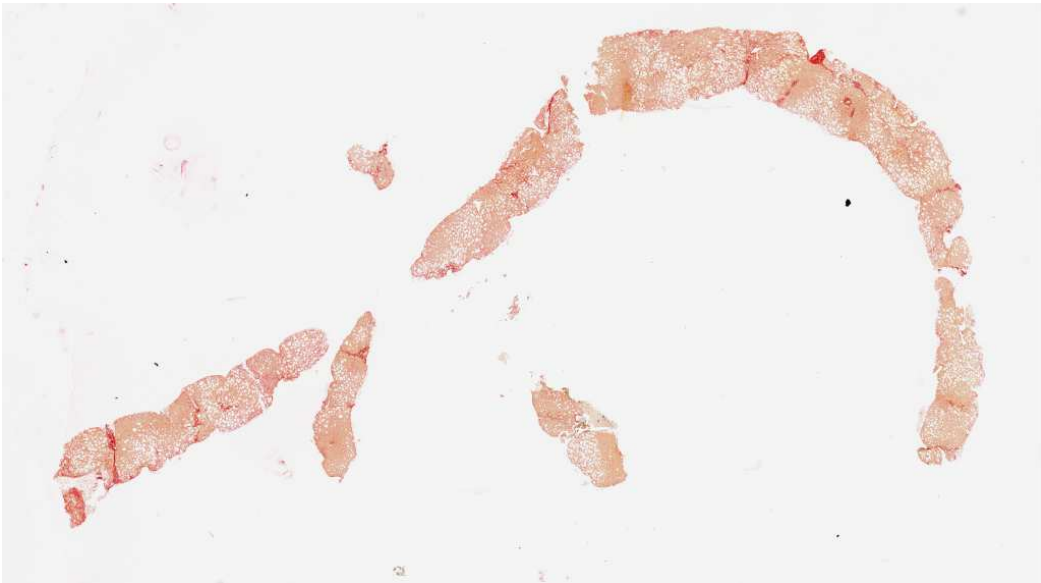


image 01 (22000×12267)

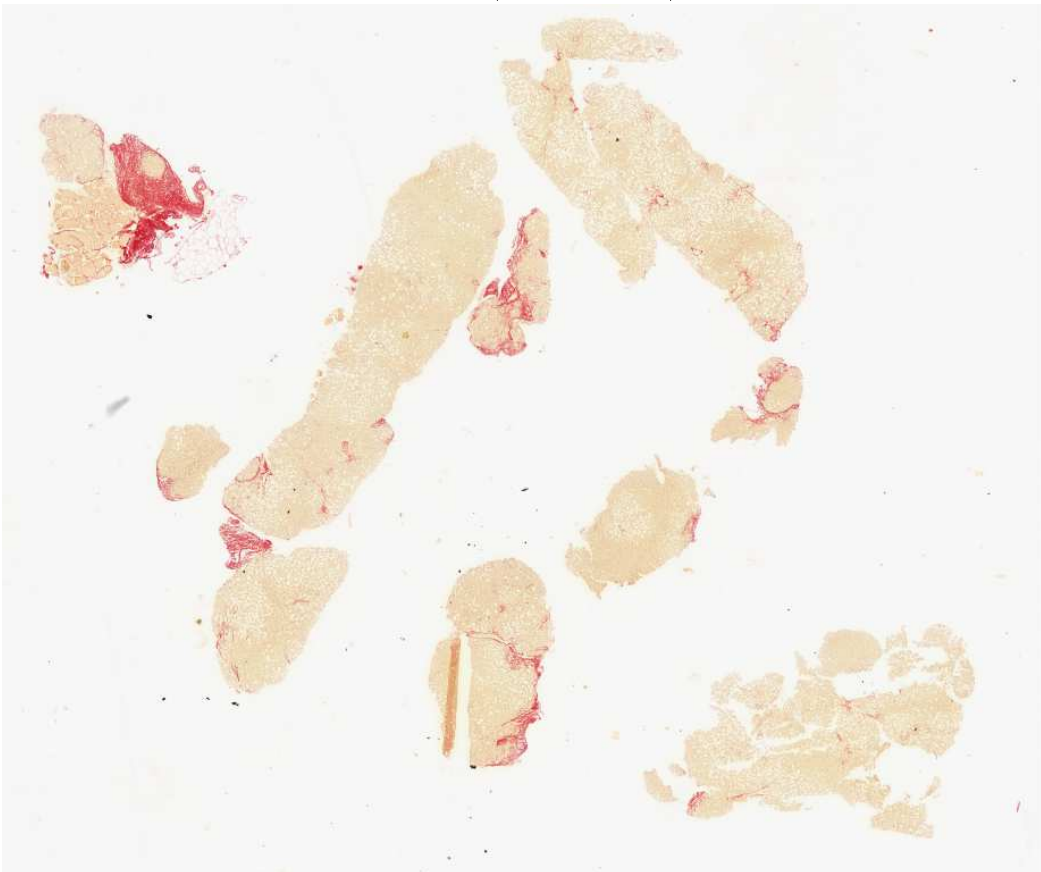


image 03 (19991×16712)





image 06 (19991×23796)



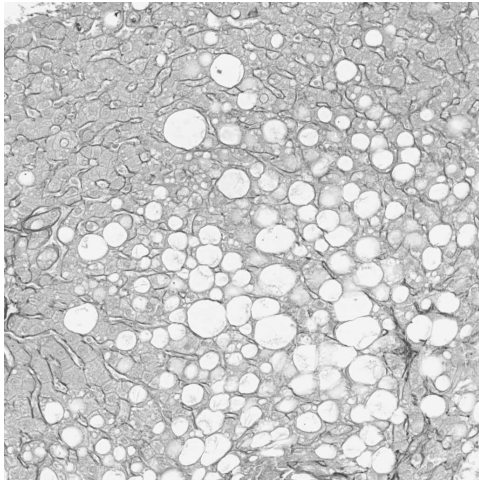
image 21 (22000×16016)



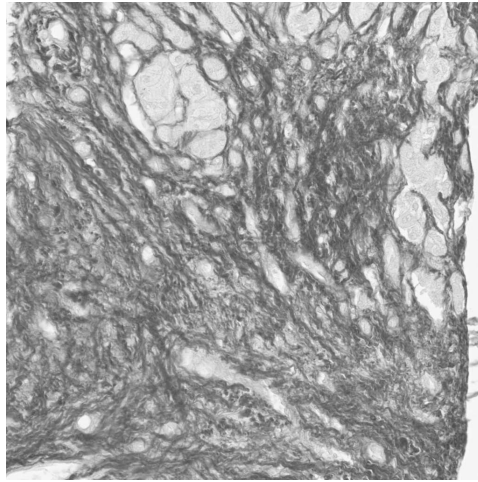
image 28 (23989×15591)

## A.4 Patches de lames virtuelles

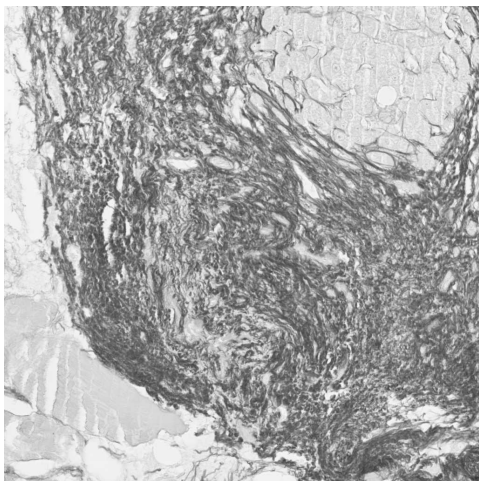
Exemples de patches utilisés pour les expérimentations sur les lames virtuelles. Chaque patch à une dimension de  $1024 \times 1024$ .



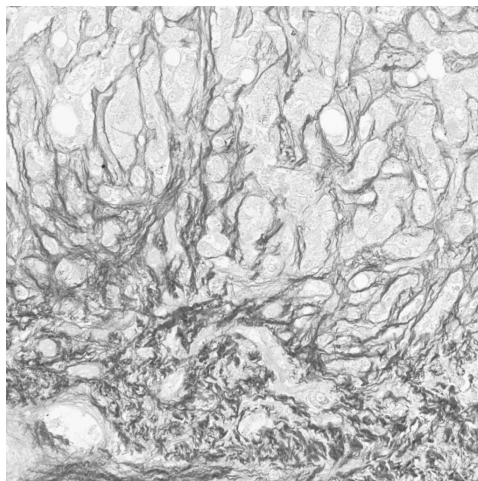
01-01



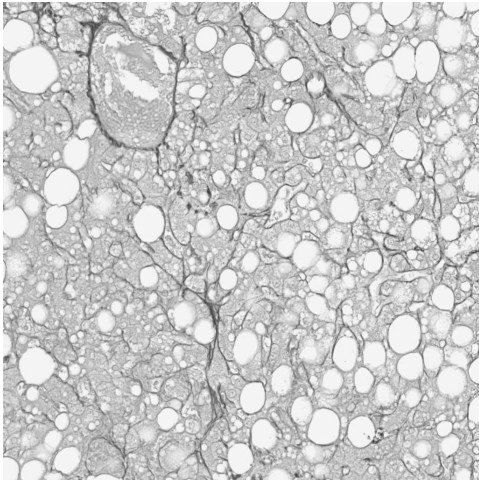
02-01



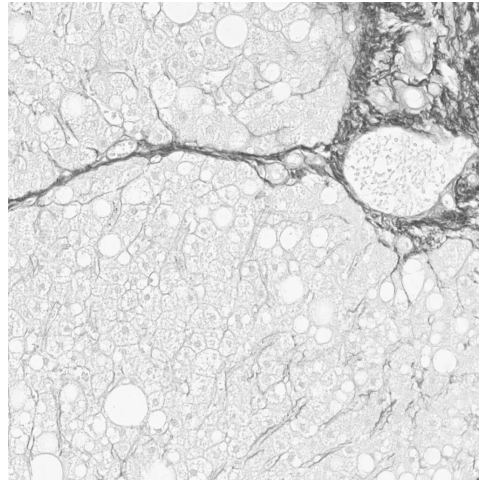
03-01



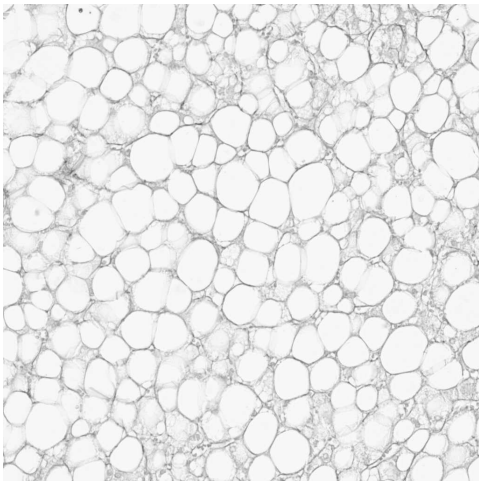
04-01



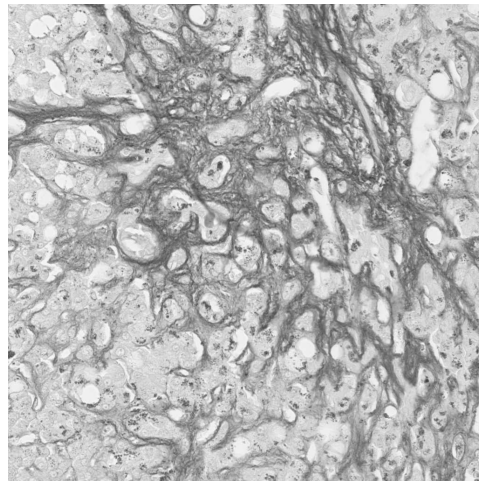
05-01



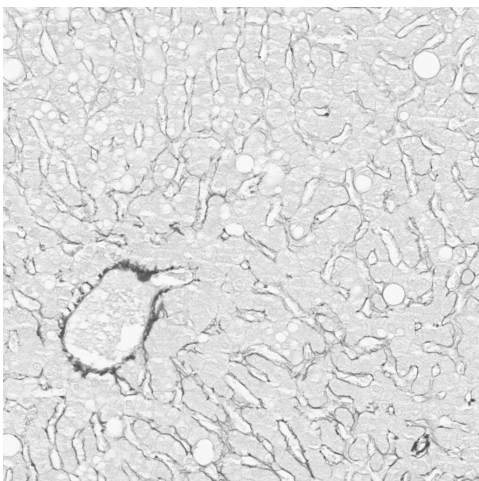
06-01



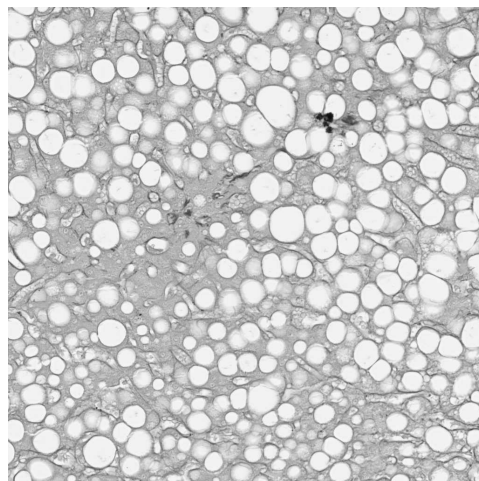
07-01



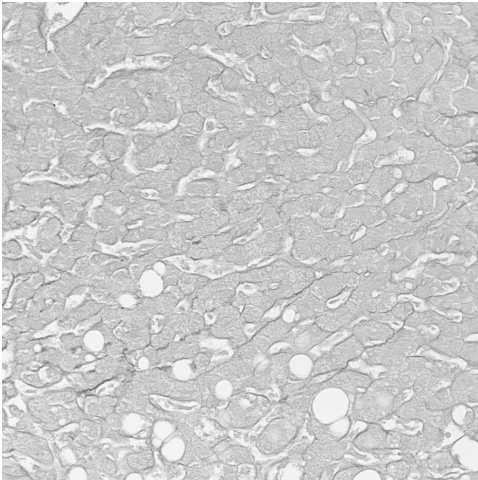
08-01



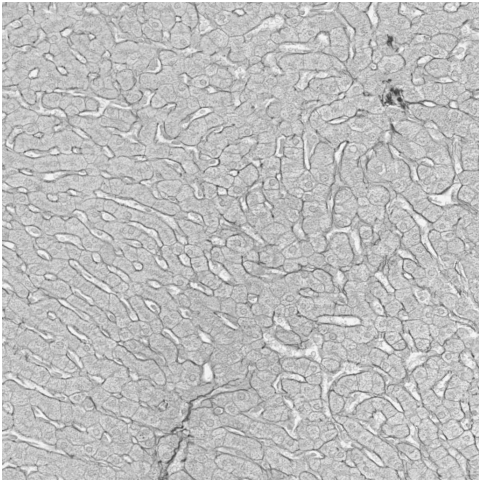
09-01



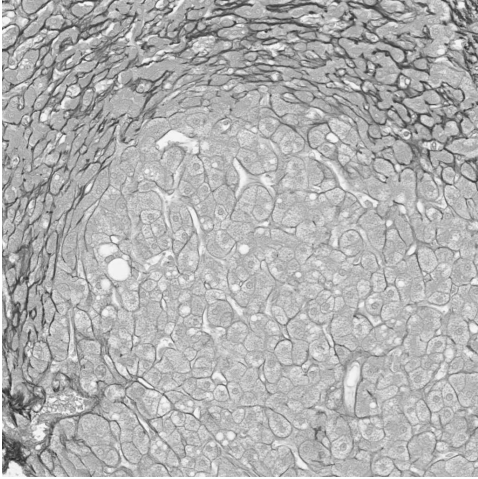
10-01



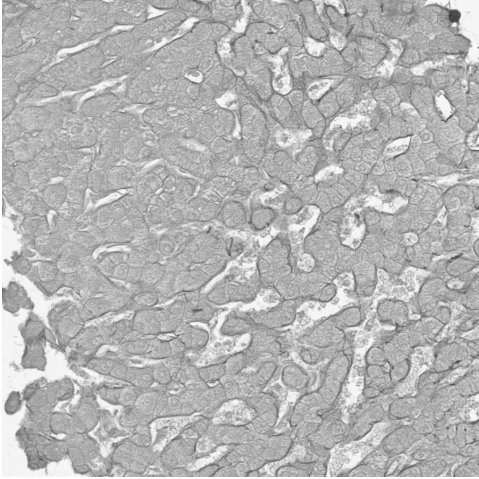
11-01



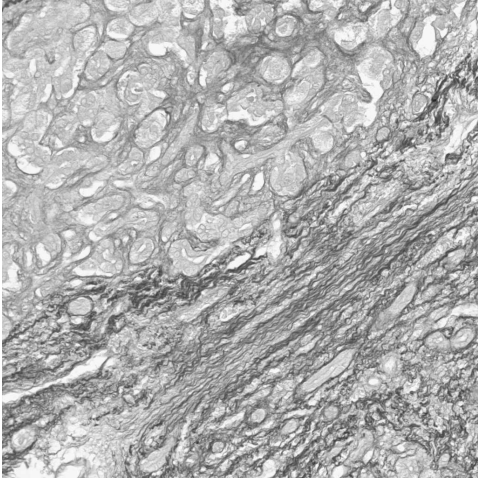
12-01



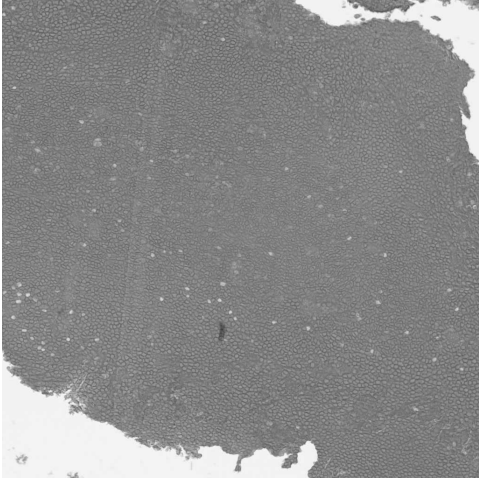
13-01



14-01



15-01



16-01

# Bibliographie

- [AAB97] Andrea Abrardo, Luciano Alparone, and Franco Bartolini. Encoding-interleaved hierarchical interpolation for lossless image compression. *Signal Processing*, 56(3) :321 – 328, 1997.
- [AAB02] B. Aiazzi, L. Alparone, and S. Baronti. Fuzzy logic-based matching pursuits for lossless predictive coding of still images. *Fuzzy Systems, IEEE Transactions on*, 10(4) :473–483, Aug 2002.
- [AABA96] B. Aiazzi, P.S. Alba, S. Baronti, and L. Alparone. Three-dimensional lossless compression based on a separable generalized recursive interpolation. In *International Conference on Image Processing (ICIP) 1996*, volume 1, pages 85–88 vol.1, Sep 1996.
- [AABL97] B. Aiazzi, L. Alparone, S. Baronti, and F. Lotti. Lossless image compression by quantization feedback in a content-driven enhanced laplacian pyramid. *Image Processing, IEEE Transactions on*, 6(6) :831–843, Jun 1997.
- [ABA01] B. Aiazzi, S. Baronti, and L. Alparone. Near-lossless compression of coherent image data. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 490–493 vol.3, 2001.
- [Abh03] G.C.K. Abhayaratne. Modifying integer wavelet transforms for scalable near-lossless image coding. In *Visual Communications and Image Processing (VCIP) 2003*, volume Proc. SPIE 5150, pages 1697–1708, 2003.
- [Abh07] G. Charith K. Abhayaratne. Reversible integer-to-integer mapping of n-point orthonormal block transforms. *Signal Processing*, 87(5) :950–969, 2007.
- [Ada05] Michael D. Adams. The JPEG-2000 still image compression standard. *Revision of the JPEG-2000 tutorial appeared in JPEG working group document WG1N1734*, December 03 2005.
- [AEB06] M. Aharon, M. Elad, and A. Bruckstein. K-SVD : An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *Signal Processing, IEEE Transactions on*, 54(11) :4311 –4322, nov. 2006.
- [AHS02] A. Abu-Hajar and R. Sankar. Integer-to-integer shape adaptive wavelet transform for region of interest image coding. In *Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop. Proceedings of 2002 IEEE 10th*, pages 94–97, Oct. 2002.

- [AHS04] A. Abu-Hajar and R. Sankar. Region of interest coding using partial-spiht. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 3, pages iii-657–60 vol.3, May 2004.
- [ALM09] F. Auli-Llinas and M.W. Marcellin. Distortion estimators for bitplane image coding. *Image Processing, IEEE Transactions on*, 18(8) :1772 – 1781, aug. 2009.
- [ALM11] F. Auli-Llina and M.W. Marcellin. Rate-distortion optimized adaptive scanning order for bitplane image coding engines. In *Data Compression Conference (DCC), 2011*, pages 163 –172, march 2011.
- [ALMSS09] F. Auli-Llinas, M.W. Marcellin, and J. Serra-Sagrista. Highly accurate distortion estimation for jpeg2000 through pdf-based estimators. In *Data Compression Conference, 2009. DCC '09.*, pages 391 –400, march 2009.
- [AMCS06] A. Alecu, A. Munteanu, J.P.H. Cornelis, and P. Schelkens. Wavelet-based scalable l-infinity-oriented compression. *Image Processing, IEEE Transactions on*, 15(9) :2499–2512, Sept. 2006.
- [AMSS02] I. Avcibas, N. Memon, B. Sankur, and K. Sayood. A progressive lossless/near-lossless image compression algorithm. *Signal Processing Letters, IEEE*, 9(10) :312–314, Oct 2002.
- [AMSS05] I. Avcibas, N. Memon, B. Sankur, and K. Sayood. A successively refinable lossless image-coding algorithm. *Communications, IEEE Transactions on*, 53(3) :445–452, March 2005.
- [And97] J. Andrew. A simple and efficient hierarchical image coder. In *Image Processing, 1997. Proceedings., International Conference on*, volume 3, pages 658–661 vol.3, Oct 1997.
- [ANR74] N. Ahmed, T. Natarajan, and K.R. Rao. Discret cosine transform. *IEEE Signal Processing Magazine*, 14(2) :24–41, 1974.
- [BA83] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *Communications, IEEE Transactions on*, 31(4) :532–540, Apr 1983.
- [Bas08] Atilla Baskurt. *State of the Art of Compression Methods*, chapter 2, pages 15–42. In Naït-Ali and Cavaro-Ménard [NACM08], 2008.
- [BBM<sup>+</sup>04] Martin Blaimer, Felix Breuer, Matthias Mueller, Robin M. Heidemann, Mark A. Griswold, and Peter M. Jakob. SMASH, SENSE, PILS, GRAPPA : how to choose the optimal method. *Topics in magnetic resonance imaging : TMRI*, 15(4) :223–236, August 2004.
- [BCK07] Eric Bodden, Malte Clasen, and Joachim Kneis. Arithmetic coding revealed - a guided tour from theory to praxis. Technical Report SABLE-TR-2007-5, Sable Research Group, School of Computer Science, McGill University, Montréal, Québec, Canada, May 2007. <http://www.bodden.de/legacy/arithmetic-coding/>.

- [BdP96] J. Browne and A.B. de Pierro. A row-action alternative to the em algorithm for maximizing likelihood in emission tomography. *Medical Imaging, IEEE Transactions on*, 15(5) :687–699, Oct 1996.
- [BDR05] M. Babel, O. Deforges, and J. Ronsin. Interleaved s+p pyramidal decomposition with refined prediction model. *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, 2 :II–750–3, Sept. 2005.
- [Ber99] Christophe Bernard. *Ondelettes et problèmes mal posés : la mesure du flot optique et l'interpolation irrégulière*. PhD thesis, Ecole Polytechnique, 1999.
- [BFG05] G. Boisson, E. François, and C. Guillemot. Twavix : une solution basée-ondelettes (t+2d) pour le codage vidéo scalable. In *Actes de la 10ème conférence sur la COmpression et la REprésentation des Signaux Audio-visuels, CORESA 2005*, Rennes, France, nov 2005.
- [BPBCG93] A. Baskurt, F. Peyrin, H. Benoit-Cattin, and R. Goutte. Coding of 3d medical images using 3d wavelet decompositions. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 5, pages 562 –565 vol.5, april 1993.
- [BRS<sup>+</sup>04] M. Bertram, G. Rose, D. Schafer, J. Wiegert, and T. Aach. Directional interpolation of sparsely sampled cone-beam ct sinogram data. In *Biomedical Imaging : Nano to Macro, 2004. IEEE International Symposium on*, pages 928–931 Vol. 1, April 2004.
- [BSS09] Ian Blanes and Joan Serra-Sagrilà. Clustered reversible-klt for progressive lossy-to-lossless 3d image coding. *Data Compression Conference*, 0 :233–242, 2009.
- [BvdE92] F.A.M.L. Bruekers and A.W.M. van den Enden. New networks for perfect inversion and perfect reconstruction. *Selected Areas in Communications, IEEE Journal on*, 10(1) :129 –137, jan 1992.
- [CAR10] L'Association canadienne des radiologistes CAR. Normes de la CAR en matière de compression irréversible pour l'imagerie numérique diagnostique en radiologie, Avril 2010.
- [CAR11] The Canadian Association of Radiologists CAR. CAR standards for irreversible compression in digital diagnostic imaging within radiology. [http://www.car.ca/uploads/standards\\_guidelines/201106\\_EN\\_Standard\\_Lossy\\_Compression.pdf](http://www.car.ca/uploads/standards_guidelines/201106_EN_Standard_Lossy_Compression.pdf), June 2011.
- [CBC<sup>+</sup>03] U. Catalyurek, M.D. Beynon, Chialin Chang, T. Kurc, A. Sussman, and J. Saltz. The virtual microscope. *Information Technology in Biomedicine, IEEE Transactions on*, 7(4) :230 –248, dec. 2003.
- [CD93] A Cohen and Ingrid Daubechies. Orthonormal bases of compactly supported wavelets iii. better frequency resolution. *SIAM Journal on Mathematical Analysis*, 24(2) :520–527, 1993.



- [CD04] Emmanuel J. Candès and David L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise  $c^2$  singularities. *Communications on Pure and Applied Mathematics*, 57(2) :219–266, 2004.
- [CDDY06] Emmanuel Candès, Laurent Demanet, David Donoho, and Lexing Ying. Fast discrete curvelet transforms. *Multiscale Modeling and Simulation*, 5(3) :861–899, 2006.
- [CDF90] A. Cohen, Ingrid Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communication on Pure and Applied Mathematics*, 45(5) :485–560, 1990.
- [Cha00] Anindya Chatterjee. An introduction to the proper orthogonal decomposition. *Current*, 78(7) :808–817, 2000.
- [Che07] Yao-Tien Chen. *Medical Image Compression and Segmentation based on Statistical Inferences*. PhD thesis, Institute of Computer Science and Information Engineering from National Central University, Taiwan, R.O.C., May 2007.
- [CKP04] Sungdae Cho, Dongyoun Kim, and William A. Pearlman. Lossless compression of volumetric medical images with improved three-dimensional spiht algorithm. *Journal of Digital Imaging*, 17(1) :57–63, March 2004.
- [CLMW09] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of ACM*, 58(1) :1–37, 2009.
- [Clu00] David A. Clunie. Lossless compression of grayscale medical images - effectiveness of traditional and state of the art approaches. In *Medical Imaging 2000 : PACS Design and Evaluation : Engineering and Clinical Issues*, volume 3980 of *Proceedings of SPIE*, pages 74–84, San Diego, Calif, USA, February 2000.
- [CMLCBT08] Christine Cavarro-Ménard, Patrick Le Callet, Dominique Barba, and Jean-Yves Tanguy. *Quality Assessment of Lossy Compressed Medical Images*, chapter 5, pages 101–128. In Naït-Ali and Cavarro-Ménard [NACM08], 2008.
- [CMNADB08] Christine Cavarro-Ménard, Amine Naït-Ali, Olivier Deforges, and Marie Babel. *Compression of 2D Biomedical Images*, chapter 7, pages 155–186. In Naït-Ali and Cavarro-Ménard [NACM08], 2008.
- [CMNAT<sup>+</sup>08] Christine Cavarro-Ménard, Amine Naït-Ali, Jean-Yves Tanguy, Elsa Angelini, Christel Le Bozec, and Jean-Jacques Le Jeune. *Specificities of Physiological Signals and Medical Images*, chapter 3, pages 43–76. In Naït-Ali and Cavarro-Ménard [NACM08], 2008.
- [CMW92] Ronald R. Coifman, Yves Meyer, and M. Victor Wickerhauser. Wavelet analysis and signal processing. *Wavelets and their applications*, pages 153–178, 1992.

- [CMZLC10] C. Cavaro-Menard, L. Zhang, and P. Le Callet. Diagnostic quality assessment of medical images : Challenges and trends. In *Visual Information Processing (EUVIP), 2010 2nd European Workshop on*, pages 277–284, July 2010.
- [Cor01] Emmanuel Cordonnier. Communication dans la santé. Vers la connectivité médicale multimédia. In Lavoisier, editor, *L'information médicale numérique*, volume 2 of *Les Cahiers du numérique*, pages 13–35. 2001.
- [CPS05] Yushin Cho, W.A. Pearlman, and A. Said. Low complexity resolution progressive image coding algorithm : progres (progressive resolution decompression). *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, 3 :III–49–52, Sept. 2005.
- [CR94] K. Chen and T.V. Ramabadran. Near-lossless compression of medical images through entropy-coded dpcm. *Medical Imaging, IEEE Transactions on*, 13(3) :538–548, sep 1994.
- [CTC06] Y.T. Chen, D.C. Tseng, and P.C. Chang. Wavelet-based image compression with polygon-shaped region of interest. In *2006 IEEE Pacific-Rim Symposium on Image and Video Technology (PSIVT'06)*, pages 878–887, 2006.
- [Dal93] Scott Daly. The visible differences predictor : an algorithm for the assessment of image fidelity. In Watson, editor, *Digital images and human vision*, pages 179–206. MIT Press, Cambridge, MA, USA, 1993.
- [Dal94] S. Daly. A visual model for optimizing the design of image processing algorithms. *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, 2 :16–20 vol.2, Nov 1994.
- [Dau88] Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communication on Pure and Applied Mathematics*, 41(7) :909–996, 1988.
- [Dau92] Ingrid Daubechies. *Ten Lectures on Wavelets (C B M S - N S F Regional Conference Series in Applied Mathematics)*. Soc for Industrial & Applied Math, December 1992.
- [Dau93] I Daubechies. Orthonormal bases of compactly supported wavelets ii. variations on a theme. *SIAM Journal on Mathematical Analysis*, 24(2) :499–519, 1993.
- [DBBR07] O. Deforges, M. Babel, L. Bedat, and J. Ronsin. Color lar codec : A color image representation and compression scheme based on local resolution adjustment and self-extracting region representation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(8) :974–987, Aug. 2007.
- [DBM06] Olivier Déforges, Marie Babel, and Jean Motsch. The RWHT+P for an improved lossless multiresolution coding. In *EUSIPCO'06*, Florence, Italy, Sept. 2006.

- [DBN<sup>+</sup>04] Olivier Déforges, Marie Babel, Nicolas Normand, Benoît Parrein, Joseph Ronsin, Jean-Pierre Guédon, and Laurent Bédard. Le LAR aux Mojettes. In *Proc. of CORESA'04*, pages 165–168, France, 2004. CORESA.
- [DCMT07] Xavier Delaunay, Marie Chabert, Géraldine Morin, and Carole Thiebaut. Post-transformée dans le domaine ondelettes appliquée à la compression d'images satellite. In *GRETSI, Groupe d'Etudes du Traitement du Signal et des Images*, 2007.
- [DD89] Gilles Deslauriers and Serge Dubuc. Symmetric iterative interpolation processes. *Constructive Approximation*, 5 :49–68, 1989. 10.1007/BF01889598.
- [Del08] Xavier Delaunay. *Compression d'images satellite par post-transformées dans le domaine ondelettes*. PhD thesis, Université de Toulouse, 2008.
- [DGSWAL05] M. Díez-García, F. Simmross-Wattenberg, and C. Alberola-López. A lossless compression algorithm based on predictive coding for volumetric medical datasets. In *European Signal Processing Conference, EUSIPCO-05*, Antalya, Turkey, Sept. 2005.
- [Don95] D.L. Donoho. De-noising by soft-thresholding. *Information Theory, IEEE Transactions on*, 41(3) :613–627, May 1995.
- [Don00] R. D. Dony. *Karhunen-Loève Transform*, chapter 1. In Rao and Yip [RY01], Sept. 2000.
- [dQ00] Ricardo de Queiroz. *Lapped Transforms for Image Compression*, chapter 5. In Rao and Yip [RY01], Sept. 2000.
- [dRP99] D. de Rycke and W. Philips. Lossless non-linear predictive coding of video data through context matching. In M. Torres, B. Sanchez, and D. Langlois, editors, *The 5th International Conference on Information Systems Analysis and Synthesis (ISAS'99)*, volume 6, pages 42–49, Orlando, USA, Aug. 1999.
- [DRPV06] C. Delgorge, C. Rosenberger, G. Poisson, and P. Vieyres. Towards a new tool for the evaluation of the quality of ultrasound compressed images. *Medical Imaging, IEEE Transactions on*, 25(11) :1502–1509, Nov. 2006.
- [DS98] Ingrid Daubechies and Wim Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4(3) :247–249, May 1998.
- [dSS08] Rafael A.P. dos Santos and Jacob Scharcanski. Lossless and near-lossless digital angiography coding using a two-stage motion compensation approach. *Computerized Medical Imaging and Graphics*, 32(5) :379–387, 2008.
- [Duc79] Claude E. Duchon. Lanczos filtering in one and two dimensions. *Journal of Applied Meteorology*, 18(8) :1016–1022, 1979.
- [DV05] M.N. Do and M. Vetterli. The contourlet transform : an efficient directional multiresolution image representation. *Image Processing, IEEE Transactions on*, 14(12) :2091–2106, dec. 2005.

- [DvAPL97] Koen Denecker, Steven van Assche, Wilfried Philips, and Ignace Lemahieu. State of the art concerning lossless medical image coding. In *in Proceedings of the PRORISC IEEE Benelux Workshop on Circuits, Systems and Signal Processing*, pages 129–136, 1997.
- [DWW<sup>+</sup>07] Wenpeng Ding, Feng Wu, Xiaolin Wu, Shipeng Li, and Houqiang Li. Adaptive directional lifting-based wavelet transform for image coding. *Image Processing, IEEE Transactions on*, 16(2) :416–427, Feb. 2007.
- [EC91] W.H.R. Equitz and T.M. Cover. Successive refinement of information. *Information Theory, IEEE Transactions on*, 37(2) :269–275, Mar 1991.
- [ELK95] O. Egger, Wei Li, and M. Kunt. High compression image coding using an adaptive morphological subband decomposition. *Proceedings of the IEEE*, 83(2) :272–287, Feb 1995.
- [Gau06] Yann Gaudeau. *Contributions en compression d’images médicales 3D et d’images naturelles 2D*. PhD thesis, Université Henri Poincaré, Nancy 1, 2006.
- [GBH70] R. Gordon, R. Bender, and G. T. Herman. Algebraic reconstruction techniques (art) for three dimensional electron microscopy and x-ray photography. *Journal of Theoretical Biology*, 29(3) :471–481, 1970.
- [GC08] Bernard Gibaud and Joël Chabriaïs. *Standards in Medical Image Compression*, chapter 4, pages 77–100. In Naït-Ali and Cavaro-Ménard [NACM08], 2008.
- [GEVK00] R. Gruter, O. Egger, J.M. Vesin, and M. Kunt. Rank-order polynomial subband decomposition for medical image compression. *Medical Imaging, IEEE Transactions on*, 19(10) :1044–1052, Oct. 2000.
- [GG91] Allen Gersho and Robert M. Gray. *Vector quantization and signal compression*. Kluwer Academic Publishers, Norwell, MA, USA, 1991.
- [Gil72] Peter Gilbert. Iterative methods for the three-dimensional reconstruction of an object from projections. *Journal of Theoretical Biology*, 36(1) :105 – 117, 1972.
- [GM09] Yann Gaudeau and Jean-Marie Moureaux. Lossy compression of volumetric medical images with 3d dead-zone lattice vector quantization. *Annals of Telecommunications*, 64 :359–367, 2009. 10.1007/s12243-008-0079-5.
- [GN05] JeanPierre Guédon and Nicolas Normand. The mojette transform : The first ten years. In Eric Andres, Guillaume Damiand, and Pascal Lienhardt, editors, *Discrete Geometry for Computer Imagery*, volume 3429 of *Lecture Notes in Computer Science*, pages 79–91. Springer Berlin / Heidelberg, 2005.
- [Gol66] S. W. Golomb. Run length encodings. In *IEEE Transactions on Information Theory IT-12*, pages 399–401, July 1966.
- [GSP05] N. Gupta, M.N.S. Swamy, and E. Plotkin. Despeckling of medical ultrasound images using data and rate adaptive lossy compression. *Medical Imaging, IEEE Transactions on*, 24(6) :743–754, June 2005.

- [Gué09] Jeanpierre Guédon. *The Mojette transform. Theory and applications*. ISTE-WILEY, January 2009.
- [GvV75] R. Gallager and D. van Voorhis. Optimal source codes for geometrically distributed integer alphabets (corresp.). *Information Theory, IEEE Transactions on*, 21(2) :228 – 230, mar 1975.
- [Hao04] Pengwei Hao. Customizable triangular factorizations of matrices. *Linear Algebra and its Applications*, 382 :135 – 154, 2004.
- [HCL03] Wen-Jyi Hwang, Ching-Fung Chine, and Kuo-Jung Li. Scalable medical data compression and transmission using wavelet transform for telemedicine applications. *Information Technology in Biomedicine, IEEE Transactions on*, 7(1) :54–63, March 2003.
- [HL94] H.M. Hudson and R.S. Larkin. Accelerated image reconstruction using ordered subsets of projection data. *Medical Imaging, IEEE Transactions on*, 13(4) :601–609, Dec 1994.
- [HLL75] Gabor T. Herman, Arnold Lent, and Peter H. Lutz. Iterative relaxation methods for image reconstruction. In *ACM 75 : Proceedings of the 1975 annual conference*, pages 169–174, New York, NY, USA, 1975. ACM.
- [HMa] Denis Hoa and Antoine Micheau. e-MRI : MRI physics interactive tutorial. web site : <http://www.imaios.com/en/e-Courses/e-MRI>.
- [HMb] Denis Hoa and Antoine Micheau. L'IRM pas à pas, cours interactif sur l'imagerie par résonance magnétique. site web : <http://www.imaios.com/fr/e-Cours/e-MRI>.
- [HR73] G.T. Herman and S.W. Rowland. Three methods for reconstructing objects from x rays : A comparative study. *CGIP*, 2(2) :151–178, October 1973.
- [HS01] Pengwei Hao and Qingyun Shi. Matrix factorizations for reversible integer mapping. *Signal Processing, IEEE Transactions on*, 49(10) :2314–2324, oct 2001.
- [HS03] P. Hao and Q. Shi. Reversible integer KLT for progressive-to-lossless compression of multiple component images. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 1, pages I – 633–6 vol.1, sept. 2003.
- [Hsi01] Shih-Ta Hsiang. Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling. *Data Compression Conference, 2001. Proceedings. DCC 2001.*, pages 83–92, 2001.
- [Huf52] D.A. Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9) :1098–1101, Sept. 1952.
- [HW01] Shih-Ta Hsiang and John W. Woods. Embedded video coding using invertible motion compensated 3-d subband/wavelet filter bank. *Signal Processing : Image Communication*, 16(8) :705 – 724, 2001.

- [HW02] Shih-Ta Hsiang and John W. Woods. Highly scalable and perceptually tuned embedded subband/wavelet image coding. In *VCIP*, pages 1153–1164, 2002.
- [ITU05] ITU-T Recommendation T.851. ITU-T T.81 (JPEG-1)-based still-image coding using an alternative arithmetic coder. Technical report, International Telecommunication Union, September 2005.
- [JZK10a] C. Guillemot J. Zepeda and E. Kijak. Image compression using sparse representations and the iteration-tuned and aligned dictionary. *IEEE Transactions on Image Processing (submitted)*, 2010.
- [JZK10b] C. Guillemot J. Zepeda and E. Kijak. The iteration-tuned and aligned dictionary and its application to image compression and denoising. *IEEE Transactions on Image Processing (submitted)*, 2010.
- [JZK10c] C. Guillemot J. Zepeda and E. Kijak. Iteration-tuned dictionaries for sparse representations and their tree-structured variant. *IEEE Transactions on Signal Processing (submitted)*, 2010.
- [KA08] Andrew Kingston and Florent Atrousseau. Lossless image compression via predictive coding of discrete radon projections. *Signal Processing : Image Communication*, 23 :313–324, 2008.
- [KBB<sup>+</sup>09] D. Koff, P. Bak, P. Brownrigg, D. Hosseinzadeh, A. Khademi, A. Kiss, L. Lepanto, T. Michalak, H. Shulman, and A. Volkening. Pan-canadian evaluation of irreversible compression ratios ("lossy" compression) for development of national guidelines. *Journal of Digital Imaging*, 22(6) :569–578, 2009.
- [KJML05] Xie Kai, Yang Jie, Zhu Yue Min, and Li Xiao Liang. Hvs-based medical image compression. *European Journal of Radiology*, 55(1) :139–145, July 2005.
- [KL05] Lih-Jen Kau and Yuan-Pei Lin. Adaptive lossless image coding using least squares optimization with edge-look-ahead. *Circuits and Systems II : Express Briefs, IEEE Transactions on*, 52(11) :751–755, Nov. 2005.
- [KM95] Ligang Ke and M.W. Marcellin. Near-lossless image compression : minimum-entropy, constrained-error dpcm. In *Image Processing, 1995. Proceedings., International Conference on*, volume 1, pages 298 –301 vol.1, oct 1995.
- [KM98] Ligang Ke and M.W. Marcellin. Near-lossless image compression : minimum-entropy, constrained-error dpcm. *Image Processing, IEEE Transactions on*, 7(2) :225 –228, feb 1998.
- [KOK<sup>+</sup>98] Juha Kivijarvi, Tiina Ojala, Timo Kaukoranta, Attila Kuba, László G. Nyúl, and Olli Nevalainen. A comparison of lossless compression methods for medical images. *Computerized Medical Imaging and Graphics*, 22 :323–339, 1998.

- [Kri02] A. Krivoulets. Efficient entropy coding for image compression. Technical Report TR-2002-13, IT University of Copenhagen, Feb. 2002.
- [Kri03a] A. Krivoulets. A method for progressive near-lossless image compression. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, volume 2, pages II-185-8 vol.3, Sept. 2003.
- [Kri03b] A. Krivoulets. Progressive near-lossless coding of medical images. In *Image and Signal Processing and Analysis, 2003. ISPA 2003. Proceedings of the 3rd International Symposium on*, volume 1, pages 202-207 Vol.1, Sept. 2003.
- [LBG80] Y. Linde, A. Buzo, and R. Gray. An algorithm for vector quantizer design. *Communications, IEEE Transactions on*, 28(1) :84-95, Jan 1980.
- [LBL87] Kenneth Lange, Mark Bahn, and Roderick Little. A theoretical study of some maximum likelihood algorithms for emission and transmission tomography. *Medical Imaging, IEEE Transactions on*, 6(2) :106-114, June 1987.
- [LF97] J. Lubin and D. Fibush. Sarnoff JND vision model. Technical report, T1A1.5 Working Group Document #97-612, T1 Standards Committee, 1997.
- [LGT88] D. Le Gall and A. Tabatabai. Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 761-764 vol.2, Apr 1988.
- [LHX06] Tony Lin, Pengwei Hao, and Shufang Xu. Matrix factorizations for reversible integer implementation of orthonormal m-band wavelet transforms. *Signal Process.*, 86 :2085-2093, August 2006.
- [LL00] Shipeng Li and Weiping Li. Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 10(5) :725-743, Aug 2000.
- [LLF03] S.-C.B. Lo, Huai Li, and M.T. Freedman. Optimization of wavelet decomposition for image compression and feature preservation. *Medical Imaging, IEEE Transactions on*, 22(9) :1141-1151, Sept. 2003.
- [LLK<sup>+</sup>05] Kyoung Ho Lee, Hak Jong Lee, Jae Hyoung Kim, Heung Sik Kang, Kyung Won Lee, Helen Hong, Ho Jun Chin, and Kyoo Seob Ha. Managing the ct data explosion : Initial experiences of archiving volumetric datasets in a mini-pacs. *Journal of Digital Imaging*, 18(3) :188-195, Sept. 2005.
- [Llo82] S. Lloyd. Least squares quantization in pcm. *Information Theory, IEEE Transactions on*, 28(2) :129-137, Mar 1982.
- [LLZ07] Chuanfeng Lv, Zhiwen Liu, and Qiangfu Zhao. A Flexible Non-linear PCA Encoder for Still Image Compression. In *Computer and Information Technology, 2007. CIT 2007. 7th IEEE International Conference on*, pages 645 -650, oct. 2007.

- [LN08] Yu Liu and King Ngi Ngan. Weighted adaptive lifting-based wavelet transform for image coding. *Image Processing, IEEE Transactions on*, 17(4) :500–511, April 2008.
- [LO01] Xin Li and Michael T. Orchard. Edge-directed prediction for lossless compression of natural images. *IEEE Transactions on Image Processing*, 10 :813–817, 2001.
- [LPM05] E. Le Pennec and S. Mallat. Sparse geometric image representations with bandelets. *Image Processing, IEEE Transactions on*, 14(4) :423–438, april 2005.
- [Lub93] Jeffrey Lubin. The use of psychophysical data and models in the analysis of display system performance. In Watson, editor, *Digital images and human vision*, pages 163–178. MIT Press, Cambridge, MA, USA, 1993.
- [Lub95] Jeffrey Lubin. A visual discrimination model for imaging system design and evaluation. In E. Peli, editor, *Vision Models for Target Detection and Recognition*, volume 2 of *Series on Information Display*, pages 245–283. World Scientific Publishing, 1995.
- [Lub97] J. Lubin. A human vision system model for objective picture quality measurements. *Broadcasting Convention, 1997. International*, pages 498–503, Sep 1997.
- [LXSH06] Tony Lin, Shufang Xu, Qingyun Shi, and Pengwei Hao. An algebraic construction of orthonormal m-band wavelets with perfect reconstruction. *Applied Mathematics and Computation*, 172(2) :717 – 730, 2006. Special issue for The Beijing-HK Scientific Computing Meetings.
- [LZ05a] Chuanfeng Lv and Qiangfu Zhao. An extended lbg algorithm for constructing universal image encoders. In *Electronic, Control and Signal Processing, 2005. 4th WSEAS International Conference on*, pages 251–256, Miami, Florida, USA, nov. 2005.
- [LZ05b] Chuanfeng Lv and Qiangfu Zhao. A universal pca for image compression. In Laurence Yang, Makoto Amamiya, Zhen Liu, Minyi Guo, and Franz Rammig, editors, *Embedded and Ubiquitous Computing – EUC 2005*, volume 3824 of *Lecture Notes in Computer Science*, pages 910–919. Springer Berlin / Heidelberg, 2005.
- [Mal06] H.S. Malvar. Adaptive run-length/golomb-rice encoding of quantized generalized gaussian sources with unknown statistics. *Data Compression Conference, 2006. DCC 2006. Proceedings*, pages 23–32, March 2006.
- [Mal08] Stéphane Mallat. *A Wavelet Tour of Signal Processing, Third Edition : The Sparse Way*. Academic Press, 3 edition, December 2008.
- [Max60] J. Max. Quantizing for minimum distortion. *Information Theory, IRE Transactions on*, 6(1) :7–12, March 1960.
- [MC04] Shaou-Gang Miaou and Shih-Tse Chen. Automatic quality control for wavelet-based compression of volumetric medical images using distortion-



- constrained adaptive vector quantization. *Medical Imaging, IEEE Transactions on*, 23(11) :1417–1429, Nov. 2004.
- [Men00] Gloria Menegaz. *Model-based coding of multi-dimensional data with applications to medical imaging*. PhD thesis, école polytechnique fédérale de Lausanne, 2000.
- [MK06] A.A. Moinuddin and E. Khan. Wavelet based embedded image coding using unified zero-block-zero-tree approach. *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, 2 :II–II, May 2006.
- [MMI00] I. Matsuda, H. Mori, and S. Itoh. Lossless coding of still images using minimum-rate predictors. In *Image Processing, 2000. Proceedings. 2000 International Conference on*, volume 1, pages 132–135 vol.1, 2000.
- [Mou07] Jean-Marie Moureaux. *Quantification vectorielle algébrique : un outil performant pour la compression et le tatouage d’images fixes*. Hdr, Université Henri Poincaré - Nancy I, December 2007.
- [MOUI05] Ichiro Matsuda, Nau Ozaki, Yuji Umezu, and Susumu Itoh. Lossless coding using variable block-size adaptive prediction optimized for each image. In *13th European Signal Processing Conference (EUSIPCO 2005)*, Sep. 2005.
- [MSW00] N. Merhav, G. Seroussi, and M.J. Weinberger. Optimal prefix codes for sources with two-sided geometric distributions. *Information Theory, IEEE Transactions on*, 46(1) :121–135, jan 2000.
- [MT97] B. Meyer and P. Tischer. TMW - a New Method for Lossless Image Compression. In *International Picture Coding Symposium PCS97*, sept. 1997.
- [MT98] B. Meyer and P. Tischer. Extending tmw for near lossless compression of greyscale images. In *Data Compression Conference, 1998. DCC '98. Proceedings*, pages 458–470, Mar-1 Apr 1998.
- [MT01] B. Meyer and P. Tischer. Glicbawls - grey level image compression by adaptive weighted least squares. In *Proc. Data Compression Conference*, page 503, Snowbird, Utah, USA, Mar 2001. IEEE Computer Society.
- [MT02] G. Menegaz and J.-P. Thiran. Lossy to lossless object-based coding of 3-d mri data. *Image Processing, IEEE Transactions on*, 11(9) :1053–1061, Sep 2002.
- [MT03] G. Menegaz and J.-P. Thiran. Three-dimensional encoding/two-dimensional decoding of medical data. *Medical Imaging, IEEE Transactions on*, 22(3) :424–440, March 2003.
- [Mul09] Rikus Muller. *Applying the MDCT to Image Compression*. PhD thesis, Stellenbosh University, South Africa, March 2009.

- [MYRP07] Byungjun Min, Sook Yoon, Jungjin Ra, and Dong Sun Park. Enhanced renormalization algorithm in mq-coder of jpeg2000. *Information Technology Convergence, 2007. ISITC 2007. International Symposium on*, pages 213–216, Nov. 2007.
- [NACM07] Amine Naït-Ali and Christine Cavaro-Ménard, editors. *Compression des images et des signaux médicaux*. LAVOISIER, 2007.
- [NACM08] Amine Naït-Ali and Christine Cavaro-Ménard, editors. *Compression of Biomedical Images and Signals*. ISTE / WILEY, 2008.
- [NCOC11] Binh P. Nguyen, Chee-Kong Chui, Sim-Heng Ong, and Stephen Chang. An efficient compression scheme for 4-d medical images using hierarchical vector quantization and motion compensation. *Computers in Biology and Medicine*, 41(9) :843 – 856, 2011.
- [NMOL96] A. Nosratinia, N. Mohsenian, M.T. Orchard, and B. Liu. Interframe coding of magnetic resonance images. *Medical Imaging, IEEE Transactions on*, 15(5) :639–647, Oct 1996.
- [OBBO06] P.M.A. van Ooijen, P.J.M. ten Bhomer, A. Broekema, and M. Oudkerk. Shifting storage requirements due to modality changes in six years of pacs. In *The 24th International EuroPACS Conference*, Trondheim, Norway, June 15th to 17th 2006.
- [OBO05] P.M.A. van Ooijen, P.J.M. ten Bhomer, and M. Oudkerk. Pacs storage requirements-influence of changes in imaging modalities. *International Congress Series 1281*, pages 888–893, 2005.
- [OP09] N. Ouarti and G. Peyre. Best basis denoising with non-stationary wavelet packets. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 3825 –3828, nov. 2009.
- [OZONA08] Azza Ouled Zaid, Christian Olivier, and Amine Naït-Ali. *Compression of Dynamic and Volumetric Medical Sequences*, chapter 8, pages 187–210. In Naït-Ali and Cavaro-Ménard [NACM08], 2008.
- [PBDB08] François Pasteau, Marie Babel, Olivier Déforges, and Laurent Bédard. Interleaved s+p scalable coding with inter-coefficient classification methods. In *Proc. 16th European Signal Processing Conference*, Lausanne, Switzerland, August 25-29 2008.
- [Pia08] Oleg S. Pinykh. *Digital Imaging and Communications in Medicine (DICOM) – A Practical Introduction and Survival Guide*. Springer, 2008.
- [PINS04] W.A. Pearlman, A. Islam, N. Nagaraj, and A. Said. Efficient, low-complexity image coding with a set-partitioning embedded block coder. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(11) :1219–1235, Nov. 2004.
- [PJB87] J. Princen, A. Johnson, and A. Bradley. Subband/transform coding using filter bank designs based on time domain aliasing cancellation. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '87.*, volume 12, pages 2161–2164, Apr 1987.

- [PMLA88] W. B. Pennebaker, J. L. Mitchell, G. G. Langdon, Jr., and R. B. Arps. An overview of the basic principles of the q-coder adaptive binary arithmetic coder. *IBM J. Res. Develop.*, 32(6) :717–726, 1988.
- [PPT<sup>+</sup>03] M. Penedo, W.A. Pearlman, P.G. Tahoces, M. Souto, and J.J. Vidal. Region-based wavelet coding methods for digital mammography. *Medical Imaging, IEEE Transactions on*, 22(10) :1288–1296, Oct. 2003.
- [PR07] B. Prabhakar and M. Ramasubba Reddy. Hvs scheme for dicom image compression : Design and comparative performance evaluation. *European Journal of Radiology*, 63(Issue 1) :128–135, July 2007.
- [PTMO07] B. Penna, T. Tillo, E. Magli, and G. Olmo. Transform coding techniques for lossy hyperspectral data compression. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(5) :1408–1421, May 2007.
- [PvAdRD01] W. Philips, S. van Assche, D. de Rycke, and K. Denecker. State-of-the-art techniques for lossless compression of 3d medical image sets. *Computerized Medical Imaging and Graphics*, 25(2) :173–185, 3 2001.
- [RGI07] E. Romero, F. Gómez, and M. Iregui. Virtual microscopy in medical images : a survey. In A. Méndez-Vilas and J. Díaz, editors, *Modern Research and Educational Topics in Microscopy*, pages 996–1006. FORMATEX, 2007.
- [Ric03] Iain Richardson. *H.264 and MPEG-4 video compression : Video coding for Next-generation MultiMedia*. WILEY, 2003.
- [RP71] R. Rice and J. Plaunt. Adaptive variable-length coding for efficient compression of spacecraft television data. *Communication Technology, IEEE Transactions on*, 19(6) :889 –897, december 1971.
- [RR01] K. Rose and S.L. Regunathan. Toward optimality in scalable predictive coding. *Image Processing, IEEE Transactions on*, 10(7) :965 –976, jul 2001.
- [RV93] P. Roos and M.A. Viergever. Reversible 3-D decorrelation of medical images. *Medical Imaging, IEEE Transactions on*, 12(3) :413–420, Sep 1993.
- [RVvDP88] P. Roos, M.A. Viergever, M.C.A. van Dijke, and J.H. Peters. Reversible intraframe compression of medical images. *Medical Imaging, IEEE Transactions on*, 7(4) :328–336, Dec 1988.
- [RY01] K. R. Rao and P. C. Yip, editors. *The Transform and Data Compression Handbook*. Electrical Engineering & Applied Signal Processing Series. CRC Press, Boca Raton, FL, USA, 2001.
- [Sai04] Amir Said. Introduction to arithmetic coding - theory and practice. Technical Report HPL-2004-76, Imaging Systems Laboratory, HP Laboratories Palo Alto, April 2004. <http://www.hpl.hp.com/techreports/2004/HPL-2004-76.pdf>.

- [SAN10] V. Sanchez, R. Abugharbieh, and P. Nasiopoulos. 3-d scalable medical image compression with optimized volume of interest coding. *Medical Imaging, IEEE Transactions on*, 29(10) :1808–1820, oct. 2010.
- [SCE01] Athanassios Skodras, Charilaos Chritopolos, and Touradj Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, pages 36–58, September 2001.
- [Sha48] C. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27 :379–423, 623–656, July,October 1948.
- [Sha93] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *Signal Processing, IEEE Transactions on*, 41(12) :3445–3462, Dec 1993.
- [SMB<sup>+</sup>03] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto, and J. Cornelis. Wavelet coding of volumetric medical datasets. *Medical Imaging, IEEE Transactions on*, 22(3) :441–458, March 2003.
- [SP93] Amir Said and William A. Pearlman. Reversible image compression via multiresolution representation and predictive coding. In *Visual Communications and Image Processing*, number 2094 in SPIE, pages 664–674, Nov. 1993.
- [SP96a] A. Said and W.A. Pearlman. An image multiresolution representation for lossless and lossy compression. *Image Processing, IEEE Transactions on*, 5(9) :1303–1310, Sep 1996.
- [SP96b] A. Said and W.A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *Circuits and Systems for Video Technology, IEEE Transactions on*, 6(3) :243–250, Jun 1996.
- [SR05] R. Srikanth and A.G. Ramakrishnan. Contextual encoding in uniform and adaptive mesh-based lossless compression of mr images. *Medical Imaging, IEEE Transactions on*, 24(9) :1199–1206, Sept. 2005.
- [ST02] D. Stefanoiu and I. Tabus. Euclidean lifting schemes for i2i wavelet transform implementation. *Studies in Informatics and Control*, 11(2) :255–270, 2002.
- [SV82] L. A. Shepp and Y. Vardi. Maximum likelihood reconstruction for emission tomography. *Medical Imaging, IEEE Transactions on*, 1(2) :113–122, Oct. 1982.
- [Swe96a] W. Sweldens. Wavelets and the lifting scheme : A 5 minute tour. *Z. Angew. Math. Mech.*, 76 (Suppl. 2) :41–44, 1996.
- [Swe96b] Wim Sweldens. The lifting scheme : A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, 3(2) :186 – 200, 1996.
- [Tau00] D. Taubman. High performance scalable image compression with ebcot. *Image Processing, IEEE Transactions on*, 9(7) :1158–1170, Jul 2000.

- [TJLBF08] Jean-Yves Tanguy, Pierre Jallet, Christel Le Bozec, and Guy Frija. *Relevance of Biomedical Data Compression*, chapter 1, pages 1–14. In Naït-Ali and Cavarro-Ménard [NACM08], 2008.
- [TK03] E Tanaka and H Kudo. Subset-dependent relaxation in block-iterative algorithms for image reconstruction in emission tomography. *Physics in Medicine and Biology*, 48(10) :1405–1422, 2003.
- [TL10a] J. Taquet and C. Labit. Optimized decomposition basis using lanczos filters for lossless compression of biomedical images. In *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*, pages 122–127, 2010.
- [TL10b] Jonathan Taquet and Claude Labit. Une introduction à la compression d’images médicales volumiques. Rapport de recherche RR-7324, INRIA, June 2010.
- [TL11] J. Taquet and C. Labit. Compression sans perte et avec pertes contrôlées d’images de très haute définition : les lames virtuelles. In *XXIIIeme Colloque du Groupe de Recherche et d’Etudes du Traitement du Signal et des Images (GRETSI 2011)*, BORDEAUX France, Sept. 2011.
- [TM01] David Taubman and Michael Marcellin. *JPEG2000 : Image Compression Fundamentals, Standards and Practice*. Springer, 2001.
- [TSS<sup>+</sup>08] Chengjie Tu, Sridhar Srinivasan, Gary J. Sullivan, Shankar Regunathan, and Henrique S. Malvar. Low-complexity hierarchical lapped transform for lossy-to-lossless image coding in JPEG XR / HD Photo. In Andrew G. Tescher, editor, *Applications of Digital Image Processing XXXI*, volume 7073, pages 70730C–1–70730C–12. SPIE, 2008.
- [TZ94] D. Taubman and A. Zakhor. Multirate 3-d subband coding of video. *Image Processing, IEEE Transactions on*, 3(5) :572–588, sep 1994.
- [US08] G. Ulacha and R. Stasinski. A new simple context lossless image coding algorithm based on adaptive context arithmetic coder. *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on*, pages 45–48, June 2008.
- [Vae99] Vincent Vaerman. *Multi-dimensional object modeling with application to medical image coding*. PhD thesis, Thèse sciences techniques Ecole polytechnique fédérale de Lausanne EPFL, no 2047 (1999), Faculté informatique et communications IC, Section de systèmes de communication, Institut de génie électrique et électronique IEL (Laboratoire de traitement des signaux 5 LTS5). Dir. : Murat Kunt, Lausanne, 1999.
- [VGP02] J. Viéron, C. Guillemot, and S. Pateux. Motion compensated 2d+t wavelet analysis for low rate fgs video compression. In *of the International Thyrrhenian workshop on digital communications 2002 (invited paper)*, Capri, Italy, September 2002.
- [VMT99] V. Vaerman, G. Menegaz, and J.-P. Thiran. A parametric hybrid model used for multidimensional object representation. In *Image Processing*,

1999. *ICIP 99. Proceedings. 1999 International Conference on*, volume 1, pages 163–167 vol.1, 1999.
- [WB97] Xiaolin Wu and P. Bao. Near-lossless image compression by combining wavelets and calic. In *Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on*, volume 2, pages 1427–1431 vol.2, Nov 1997.
- [WB00] Xiaolin Wu and P. Bao. L-infinity constrained high-fidelity image compression via adaptive context modeling. *Image Processing, IEEE Transactions on*, 9(4) :536–542, Apr 2000.
- [WBSS04] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment : from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4) :600–612, April 2004.
- [WC02] J.W. Woods and P.S. Chen. Improved mc-ezbc with quarter-pixel motion vectors. ISO/IEC/JTC1 SC29/WG11 doc no. m8366, Fairfax, May 2002.
- [WDJ06] K. Wahid, V. Dimitrov, and G. Jullien. New encoding of 8x8 dct to make h.264 lossless. In *Circuits and Systems, 2006. APCCAS 2006. IEEE Asia Pacific Conference on*, pages 780–783, Dec. 2006.
- [WLS07] Zhou Wang, Qiang Li, and Xinli Shang. Perceptual image coding based on a maximum of minimal structural similarity criterion. *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 2 :II–121–II–124, 16 2007-Oct. 19 2007.
- [WM97] Xiaolin Wu and Nasir Memom. Context-based, adaptative, lossless image coding. *IEEE TRANSACTIONS ON COMMUNICATIONS*, 45(4) :437–444, APRIL 1997.
- [WMS95] Xiaolin Wu, Nasir Memon, and Khalid Sayood. A context-based, adaptive, lossless/nearly-lossless coding scheme for continuous-tone images. A proposal submitted in response to the Call for Contributions for ISO/IEC JTC 1.29.12, 1995.
- [WQ05] Xiaolin Wu and Tong Qiu. Wavelet coding of volumetric medical images for high throughput and operability. *Medical Imaging, IEEE Transactions on*, 24(6) :719–727, June 2005.
- [WSB03] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. *Signals, Systems and Computers, 2003. Conference Record of the Thirty-Seventh Asilomar Conference on*, 2 :1398–1402 Vol.2, Nov. 2003.
- [WSJN07] Jiaji Wu, Guangming Shi, Licheng Jiao, and Yi Niu. Directional wavelet transform and block-set partitioning for image coding. In *Intelligent Signal Processing and Communication Systems, 2007. ISPACS 2007. International Symposium on*, pages 404–407, 28 2007-Dec. 1 2007.
- [WSS96] M. Weinberger, G. Seroussi, and G. Sapiro. LOCO-I : A Low Complexity, Context-Based, Lossless Image Compression Algorithm. In *Proc. IEEE Data Compression Conference*, Snowbird, Utah, March-April 1996.

- [WSS00] M.J. Weinberger, G. Seroussi, and G. Sapiro. The LOCO-I lossless image compression algorithm : principles and standardization into JPEG-LS. *Image Processing, IEEE Transactions on*, 9(8) :1309–1324, Aug 2000.
- [WT01] Yung-Gi Wu and Shen-Chuan Tai. Medical image compression by discrete cosine transform spectral similarity strategy. *Information Technology in Biomedicine, IEEE Transactions on*, 5(3) :236–243, Sept. 2001.
- [WWJ<sup>+</sup>08] L. Wang, J. Wu, L.C. Jiao, L. Zhang, and G.M. Shi. Lossy to lossless image compression based on reversible integer dct. In *ICIP08*, pages 1037–1040, 2008.
- [XWYP98] Zixiang Xiong, X. Wu, D.Y. Yun, and W.A. Pearlman. Progressive coding of medical volumetric data using three-dimensional integer wavelet packet transform. *Multimedia Signal Processing, 1998 IEEE Second Workshop on*, pages 553–558, Dec 1998.
- [YCP05] S. Yea, S. Cho, and W.A. Pearlman. Integrated lossy, near-lossless, and lossless compression of medical volumetric data. In Proc. SPIE 5685, editor, *Image and Video Communications and Processing 2005, IS&T/SPIE Symposium on Electronic Imaging*, volume 2, pages 151–159, Jan. 2005.
- [YDD03] Hua Ye, Guang Deng, and John C. Devlin. A weighted least squares method for adaptive prediction in lossless image compression. In *Picture Coding Symposium*, pages 489–493, Saint-Malo, France, 2003.
- [YP06] S. Yea and W.A. Pearlman. A wavelet-based two-stage near-lossless coder. *Image Processing, IEEE Transactions on*, 15(11) :3488–3500, Nov. 2006.
- [YSM10] Guoshen Yu, Guillermo Sapiro, and Stéphane Mallat. Solving inverse problems with piecewise linear estimators : From gaussian mixture models to structured sparsity. *CoRR*, abs/1006.3056, 2010.
- [ZL77] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *Information Theory, IEEE Transactions on*, 23(3) :337–343, May 1977.
- [ZVR08] Zhengbing Zhang, Radhika Veerla, and K. R. Rao. Modified advanced image coding. *Complexity and Intelligence of the Artificial and Natural Complex Systems, Medical Applications of the Complex Systems, Biomedical Computing, International Conference on*, 0 :110–116, 2008.

# Publications

- [1] J. Taquet et C. Labit, “Une introduction à la compression d’images médicales volumiques”. Rapport de Recherche RR-7324, INRIA, Juin 2010.
- [2] J. Taquet and C. Labit, “Near-lossless and scalable compression for medical imaging using a new adaptive hierarchical oriented prediction”. In *ICIP’10*, Sept. 2010.
- [3] J. Taquet and C. Labit, “Optimized decomposition basis using Lanczos filters for lossless compression of biomedical images”. In *MMSP’10*, Oct. 2010.
- [4] J. Taquet et C. Labit, “Compression sans perte et presque sans perte d’images médicales à l’aide d’un prédicteur hiérarchique orienté et adaptatif”. Dans *CORESA 2010*, Oct. 2010.
- [5] J. Taquet et C. Labit, “Compression sans perte et avec pertes contrôlées d’images de très haute définition : les lames virtuelle”. Dans *GRETSI 2011*, Sept. 2011.
- [6] J. Taquet and C. Labit, “Hierarchical Oriented Predictions for Resolution Scalable Lossless and Near-lossless Compression of CT and MRI Biomedical Images”. Submitted to *IEEE Transactions on Image Processing*, 2011.





# Table des figures

1.1	Représentations : transverse (1), sagittale (2), coronale (3) . . . . .	18
1.2	Acquisition tomодensitométrique (a) et sinogramme (b). . . . .	19
1.3	Exemple de coupe TDM . . . . .	20
1.4	Exemple de coupe IRM . . . . .	22
1.5	Exemple de lame virtuelle . . . . .	24
3.1	modèle DPCM . . . . .	62
3.2	schéma de lifting pour la prédiction hiérarchique . . . . .	67
3.3	exemple de modèle de prédiction hiérarchique . . . . .	68
3.4	Décomposition dyadique . . . . .	70
4.1	Un niveau de la prédiction hiérarchique de l'algorithme IHINT . . . . .	93
4.2	Un niveau de la prédiction hiérarchique de l'algorithme HOP . . . . .	93
4.3	Motif de prédiction contextuel (à gauche) et liens entre les pixels utilisés pour l'estimation du gradient (à droite) . . . . .	94
4.4	Ensembles de pixels causaux $\sigma_o$ utilisés pour HOP-LSE (à gauche), et pour HOP-LSE <sup>+</sup> (à gauche et à droite). . . . .	96
4.5	Réorganisation des valeurs résiduelles . . . . .	103
4.6	Codeur en deux phases . . . . .	104
4.7	Résultats débit/distorsion pour la compression presque sans perte. . . . .	111
5.1	Exemple de fonctions $L_\Theta$ à minimiser pour l'optimisation de la longueur du filtre et la sélection de la dimension de décomposition . . . . .	118
5.2	Une coupe tomographique (a) de MeDEISA (CT_data_1(105)), les valeurs de l'image entre 0 et 2272 sont mises à l'échelle pour l'affichage en 255 niveaux de gris ; et le résultat de ses décompositions (b) avec la base optimisée L-OD et (c) avec la transformée dyadique en ondelettes 5/3. Les valeurs des coefficients sont tronquées entre -128 et 127 et mises à l'échelle pour l'affichage en 255 niveaux de gris. . . . .	124
5.3	Une coupe IRM-3D (a) de VHP-Harvard-3D (T2_512(115)), les valeurs de l'image entre 0 et 1032 sont mises à l'échelle pour l'affichage en 255 niveaux de gris ; et le résultat de ses décompositions (b) avec la base optimisée L-OD et (c) avec la transformée dyadique en ondelettes 5/3. Les valeurs des coefficients sont tronquées entre -64 et 63 et mises à l'échelle pour l'affichage en 255 niveaux de gris. . . . .	125

5.4	Courbes débit/distorsion obtenues pour la représentation progressive en qualité « <i>lossy to lossless</i> » de coupes IRM et tomographiques, douces et natives, après l'optimisation de la décomposition pour la compression sans perte. (a) MeDEISA (CT_data_1(105)), (b) VHP-Harvard-3D (T2_512(115)), (c) NLM-VHP Male (normalCT(248)), (d) NLM-VHP Male (PD-1(21)).	126
5.5	Problème de bruit induit par l'utilisation de la décomposition optimisée pour la compression sans perte, et appliquée en compression presque sans perte.	131
5.6	Illustration du problème pouvant survenir après une quantification inappropriée de la bande basse résolution pour la compression presque sans perte. Exemple de résultat obtenus sur l'image A.2-(l) pour des PAE allant de 1 à 16.	132
5.7	Résultats de la compression presque sans perte pour des PAE allant de 1 à 16 i) d'une IRM bruitée A.2-(l), ii) d'une image TDM douce A.1-(e), et iii) d'une IRM douce A.2-(p). La méthode d'optimisation de recentrage pour la quantification de l'image basse résolution a été appliquée.	133
6.1	Domaine direct : vecteurs (ordonnés ligne par ligne, colonne par colonne) de la base obtenue pour un apprentissage mono KLT (sur la lame virtuelle 01 A.3) sur des blocs $16 \times 16$ .	146
6.2	Domaine direct : 16 premiers vecteurs (ordonnés de gauche à droite) de chaque base obtenue pour un apprentissage 16-KLT (sur la lame virtuelle 01 A.3) sur des blocs $16 \times 16$ obtenus sur une décomposition dyadique sur 3 niveaux.	147
6.3	Arbre hiérarchique des coefficients d'ondelettes mis en relation par leur position spatiale pour une décomposition dyadique, pouvant permettre de construire un bloc de coefficients hiérarchiques (ici $8 \times 8$ ).	148
6.4	Approche globale : vecteurs (ordonnés ligne par ligne, colonne par colonne) de la base obtenue pour un apprentissage mono KLT (sur la lame virtuelle 01 A.3) sur des blocs $16 \times 16$ obtenus sur une décomposition dyadique sur 3 niveaux.	149
6.5	Approche globale : 24 premiers vecteurs (ordonnés ligne par ligne, colonne par colonne) de chaque base obtenue pour un apprentissage 8-KLT (sur la lame virtuelle 01 A.3) sur des blocs $16 \times 16$ obtenus sur une décomposition dyadique sur 3 niveaux.	150
6.6	Ensembles de type <i>A</i> et <i>B</i> utilisés par SPIHT et pour le codage de l'approche intra-bande : le type <i>A</i> contient $2 \times 2$ coefficients et 1 ensemble de type <i>B</i> ; le type <i>B</i> contient $2 \times 2$ ensembles de type <i>A</i> .	151
6.7	Approche intra sous-bande : premiers vecteurs de chaque base obtenue pour un apprentissage 16-KLT (sur la lame virtuelle 01 A.3) sur des blocs $8 \times 8$ pour une décomposition dyadique sur 3 niveaux (1).	153
6.8	Approche intra sous-bande : premiers vecteurs de chaque base obtenue pour un apprentissage 16-KLT (sur la lame virtuelle 01 A.3) sur des blocs $8 \times 8$ pour une décomposition dyadique sur 3 niveaux (2).	154

6.9	Courbes débit/PSNR obtenues avec le codeur SPIHT pour une décomposition dyadique sur 3 niveaux utilisant les symlettes. Ces résultats sont calculés sur l'ensemble des patchs de test pour les lames virtuelles. . . . .	156
6.10	Approche directe (dans le domaine image) : différence de PSNR (sur l'ensemble des 120 échantillons de test) entre un apprentissage multi et un apprentissage mono KLT. . . . .	156
6.11	Approche globale : différence de PSNR (sur l'ensemble des 120 échantillons de test) entre un apprentissage multi et un apprentissage mono KLT. . . . .	158
6.12	Approche intra : différence de PSNR (sur l'ensemble des 120 échantillons de test) entre un apprentissage multi et un apprentissage mono KLT. . . . .	159
6.13	Meilleures performances : différence de PSNR (sur l'ensemble des 120 échantillons de test) entre un apprentissage multi KLT et la transformée en ondelettes. . . . .	159
6.14	Exemple visuel (pour affichage sur écran) des distorsions pour une compression à 0.5bpp, blocs de taille $64 \times 64$ extraits du patch 05-01. . . . .	161
6.15	Exemple visuel (pour affichage sur écran) des distorsions pour une compression à 1bpp, blocs de taille $64 \times 64$ extraits du patch 05-01. . . . .	162
6.16	Exemple de problèmes pouvant être engendrés par les erreurs d'approximations de la factorisation PLUS lorsque la dimension $D$ des bases est trop importante. Le patch de test 01-01 est ici transformée avec l'approche globale utilisant un apprentissage sur deux classes pour des blocs hiérarchiques $16 \times 16$ obtenus à partir d'une décomposition dyadique sur 3 niveaux ( $D = 256$ ). Les coefficients sont réorganisés en sous-bandes en fonction de l'indice du vecteur propre. Les valeurs absolues des coefficients sont ajustées entre 0 et 64. Les 196 ièmes coefficients de la première base et les 211 ièmes de la seconde, calculées comme étant les 256 ièmes avant la permutation ( $P$ ), sont les valeurs problématiques. Ces coefficients, censés être de faible amplitude si la factorisation était exacte, engendrent de grosses pertes de performances pour la compression avec l'extension de SPIHT présentée. . . . .	165
6.17	Différence de PSNR entre les apprentissages mono et multi KLT en mode réversible. . . . .	170
6.18	Différence de PSNR entre les apprentissages multi KLT et la transformée en ondelettes en mode réversible. . . . .	171
6.19	Gains sur le PSNR liés à (a) l'utilisation de $\delta = 1/2$ au lieu de $\delta = 0$ , (b) l'utilisation du codage arithmétique lorsque $\delta = 1/2$ . Résultats obtenus en mode de compression irréversible sur l'ensemble des patchs de test. . . . .	172
6.20	Différence de PSNR avec le logiciel de référence de l'algorithme SPIHT pour la compression irréversible. . . . .	174
6.21	Exemple visuel (pour affichage sur écran) des distorsions pour une compression irréversible à 0.5bpp utilisant le codage arithmétique et $\delta = 1/2$ , blocs de taille $64 \times 64$ extraits du patch 05-01. . . . .	176

6.22	Exemple visuel (pour affichage sur écran) des distorsions pour une compression irréversible à 1bpp utilisant le codage arithmétique et $\delta = 1/2$ , blocs de taille $64 \times 64$ extraits du patch 05-01. . . . .	177
6.23	Différence de PSNR avec le logiciel de référence de l'algorithme SPIHT pour la compression sans perte (utilisant la transformée en ondelettes S+P). . . . .	178



**Résumé** La compression d'images biomédicales est un enjeu économique important, surtout pour les TDM, les IRM et les lames virtuelles (LV), et en particulier pour leur archivage et leur transmission. Cette thèse dresse un panorama des besoins et des solutions existantes et cherche à proposer de nouveaux algorithmes efficaces en comparaison aux références standardisées.

Pour les TDM et IRM, un archivage de très bonne qualité est imposé. Ces travaux se sont donc focalisés sur la compression sans perte (SP) et presque sans perte (PSP). Il est proposé de i) fusionner la prédiction hiérarchique par interpolation avec un schéma DPCM adaptatif pour fournir une représentation scalable en résolution efficace en SP et en PSP, ii) optimiser pour chaque image une décomposition en paquets d'ondelettes. Les résultats des deux contributions montrent qu'il existe encore une marge de progression pour la compression des images les plus régulières et les moins bruitées.

Pour les LV, la lame physique peut être conservée, la problématique concerne donc plus le transfert pour la consultation à distance que l'archivage. De par leur contenu, une approche basée sur l'apprentissage des spécificités structurelles de ces images semble intéressante. Cette troisième contribution vise donc une optimisation hors-ligne de  $K$  transformées orthonormales optimales pour la décorrélation (K-KLT). Cette méthode est notamment utilisée pour construire des post-transformées sur une décomposition en ondelettes. Leur application dans un modèle de compression scalable en qualité montre que l'approche peut permettre d'obtenir des gains de qualité intéressants en terme de PSNR.

**Abstract** Compression of biomedical images, especially computed tomography (CT), magnetic resonance imaging (MRI) and virtual microscopy (VM), is an important economic issue, particularly for archival and transmission. This thesis provides an overview of the medical needs and of the existing compression solutions. It also aims, in this context, at providing new digital compression algorithms that are efficient in comparison with the state of the art standards.

For CT and MRI, a really good quality archival is imposed. Therefore this work has focused on lossless and near-lossless compression. It is proposed to i) combine the hierarchical interpolation predictive model with the adaptive DPCM one to provide a resolution scalable model that is effective for lossless and especially for near-lossless compression, ii) rely on an optimization for lossless compression of a wavelet packet decomposition structure, that is specific to the image content. The results of both contributions shows that there is still a room for improving the compression on most regular/smooth and less noisy images.

For VM, the physical slides can be stored, so the issue more concerns the transfer for remote access than the archival. By the nature of their content, the learning of the properties of these images seems to be interesting. So, this third contribution aims to optimize offline  $K$  orthonormal transforms that are optimal for the decorrelation of training data (K-KLT). This method is applied to learn in particular post-transforms for a wavelet decomposition. Their application in a quality scalable compression scheme shows that the approach can yield rather interesting quality gains in terms of PSNR.