Ondelettes et décompositions spatio-temporelles avancées; application au codage vidéo scalable

Grégoire Pau

TELECOM PARIS
école nationale supérieure des télécommunications

Thèse

présentée pour obtenir le grade de docteur
de l'Ecole nationale supérieure des télécommunications

Spécialité : **Signal et Images**

Grégoire PAU

Ondelettes et décompositions spatio-temporelles avancées ; application au codage vidéo scalable

Soutenue le 15 juin 2006 devant le jury composé de :

Christine Guillemot	Rapporteurs
Riccardo Leonardi	
Benoît Macq	Examinateurs
Nicolas Moreau	
Gemma Piella	Membre invité
Béatrice Pesquet-Popescu	Directeurs de thèse
Basarab Matei	

Remerciements

Je remercie avant tout Béatrice Pesquet-Popescu et Basarab Matei pour avoir assuré la direction de ma thèse avec enthousiasme, disponibilité et rigueur scientifique. Qu'ils trouvent ici l'expression de toute ma gratitude pour leurs conseils, leurs suggestions avisées et pour la confiance qu'ils m'ont accordée tout au long de mes recherches.

Je tiens de plus à remercier Benoît Macq d'avoir accepté de présider mon jury de thèse, Christine Guillemot et Riccardo Leonardi d'avoir bien voulu rapporter ce manuscrit. Je remercie aussi Nicolas Moreau et Gemma Piella d'avoir accepté de faire partie du jury.

Le laboratoire de Traitement du Signal de l'ENST est un endroit vivant et agréable, dans lequel j'ai passé trois années intenses, chargées de travail et de stress mais aussi de joies et d'instants mémorables. Je ne saurais trop remercier les doctorants de l'équipe, Christophe, Dora, Maria, Lionel, Tize, Yol, Séb, Loïs, Laurence, Ioana, Caroline, Cléo, Slim, Chloé, Rémi, Thomas, Julie, Roland, Simoné, Valentin, Nancy, Cyril, Mathieu, Miguel, Eduardo, Sylvia et tous les autres, pour tous ces moments passés à errer dans les cafés de la Butte aux Cailles et pour toutes ces soirées tardives précédant les échéances de soumissions aux conférences...

Les marges de cette page sont trop étroites pour remercier aussi tous mes amis et mes proches pour m'avoir supporté durant cette période. Merci en particulier à Laurent, pour sa perpétuelle bonne humeur et ses qualités de relecteur !

Enfin, je souhaite remercier mon père, Éléonore et Judith pour m'avoir soutenu dans ce projet professionnel. Je dédie cette thèse à la mémoire de ma mère Hélène, et à Josselin, pour qui les ondelettes n'ont désormais plus de secrets.

Table des matières

Table des matières 5

Glossaire 9

Introduction 11

I Ondelettes et codage vidéo scalable : un état de l'art 13

1 Ondelettes dyadiques et nouvelles représentations multirésolution 15
1.1 Introduction aux représentations multirésolution . . . 15
1.2 Ondelettes dyadiques . . . 18
1.2.1 Bases d'ondelettes . . . 18
1.2.2 Ondelettes et bancs de filtres . . . 19
1.2.3 Panorama d'ondelettes dyadiques utilisées en codage d'image . . . 24
1.2.4 Compression d'image par transformée en ondelettes . . . 26
1.3 Nouvelles représentations multirésolution . . . 30
1.3.1 Structure lifting . . . 30
1.3.2 Ondelettes géométriques non-adaptatives . . . 35
1.3.3 Ondelettes géométriques adaptatives . . . 37
1.3.4 Autres représentations . . . 38
1.4 Conclusion . . . 39

2 Codage vidéo scalable : un état de l'art 41
2.1 Codage vidéo hybride scalable . . . 41
2.1.1 Schéma de principe d'un codeur vidéo hybride . . . 41
2.1.2 Panorama des codecs MPEG et H.26X . . . 43
2.1.3 Scalabilité et l'extension MPEG-4 FGS . . . 44
2.1.4 SVC ou l'extension scalable de H.264 . . . 45
2.2 Codage vidéo scalable par ondelettes . . . 46
2.2.1 Premières approches . . . 46
2.2.2 Schéma de codage vidéo t+2D . . . 46
2.2.3 Premiers filtres temporels . . . 48
2.2.4 Exemple de schéma de codage t+2D : le codec MC-EZBC . . . 51
2.2.5 Améliorations apportées au schéma t+2D . . . 55
2.3 Conclusion . . . 58

II Mise en œuvre d'un codec vidéo scalable t+2D 59

3 Filtrage temporel 5/3 61
3.1 Filtrage temporel 5/3 compensé en mouvement . . . 61

3.1.1 Notations . . . 61
3.1.2 Lifting temporel . . . 62
3.1.3 Construction d'une transformée 5/3 compensée en mouvement . . 63
3.1.4 Traitement au fil de l'eau . . . 70
3.2 Résultats expérimentaux . . . 73
3.2.1 Efficacité de codage . . . 74
3.2.2 Scalabilité temporelle . . . 75
3.3 Conclusion . . . 76

4 Optimisation du filtrage temporel 77
4.1 Optimisation des vecteurs impliqués dans la prédiction . . . 78
4.1.1 Présentation du problème . . . 78
4.1.2 Prédiction itérative bidirectionnelle jointe . . . 80
4.1.3 Prédiction bidirectionnelle à vecteur de mouvement unique . . . 82
4.1.4 Résultats expérimentaux . . . 83
4.1.5 Conclusion . . . 91
4.2 Transformée temporelle 5/3 de sens uniforme . . . 92
4.2.1 Artefacts fantômes et mise à jour . . . 93
4.2.2 Transformée temporelle 5/3 de sens de mouvement uniforme . . . 94
4.2.3 Prédiction bidirectionnelle optimale des zones découvertes . . . 98
4.2.4 Résultats expérimentaux . . . 101
4.2.5 Conclusion . . . 104
4.3 Modération de la latence . . . 104
4.3.1 Introduction, latence et délais . . . 105
4.3.2 Analyse des délais créés par différents filtres temporels . . . 105
4.3.3 Construction d'un filtre temporel flexible à délai contraint . . . 109
4.3.4 Résultats expérimentaux . . . 111
4.3.5 Conclusion . . . 113
4.4 Transformée Daubechies-4 compensée en mouvement . . . 114
4.4.1 Description et mise en œuvre . . . 114
4.4.2 Résultats expérimentaux . . . 117
4.5 Conclusion . . . 118

5 Bancs de filtres M-bandes et filtrage spatial 121
5.1 Bancs de filtres M-bandes ; rappels . . . 121
5.1.1 Définition . . . 122
5.1.2 Transformées en blocs . . . 123
5.1.3 Transformées à recouvrement . . . 125
5.2 Codage spatial par bancs de filtres M-bandes . . . 127
5.2.1 Caractéristiques des sous-bandes temporelles . . . 127
5.2.2 Construction d'un banc de filtres 4-bandes adapté . . . 130
5.2.3 Étude de différents bancs de filtres . . . 135
5.3 Scalabilité fractionnaire . . . 138
5.3.1 Motivation . . . 139
5.3.2 Modification du banc de synthèse . . . 140
5.3.3 Complexité théorique . . . 143
5.3.4 Résultats expérimentaux . . . 144
5.4 Conclusion . . . 152

6 Filtrage spatial par lifting adaptatif **155**
6.1 Mise à jour adaptative avec critère de seuil binaire 156
6.1.1 Motivation 156
6.1.2 Décomposition avec mise à jour adaptative et critère TC 156
6.2 Comparaison de deux seminormes 159
6.2.1 Résultats principaux 159
6.2.2 Un cas d'étude : la seminorme pondérée $p(\boldsymbol{v}) = |\boldsymbol{a}^T\boldsymbol{v}|$ 161
6.3 Comparaison de N seminormes 165
6.4 Combinaison de deux seminormes et du critère TC 169
6.5 Résultats expérimentaux 171
6.5.1 Protocole expérimental 171
6.5.2 Détail des expérimentations 172
6.5.3 Efficacité de codage sans perte 178
6.6 Conclusion 183

Conclusion générale **185**

Annexe A : Preuves **189**

Annexe B : Filtrage temporel M-bandes et codage H.264 **193**

Publications **203**

Table des figures **205**

Bibliographie **209**

Index **219**

Glossaire

4CIF : *Four CIF* – Format de résolution mesurant 704×576 pixels.
CIF : *Common Interchange Format* – Format de résolution mesurant 352×288 pixels.
DCT : *Discrete Cosine Transform* – Transformée en cosinus discrète.
Dyadique : Relatif à une puissance de deux.
EZBC : *Embedded Zeroblock coding based on Context modeling* – Codec d'image fixe.
GOF : *Group of Frames* – Groupe d'images consécutives.
GOP : *Group of Pictures* – Groupe d'images consécutives.
H.26X : Famille d'algorithmes de codage vidéo normalisés par l'ITU.
HD : Haute Définition.
ISO : *International Organization for Standardization* – Organisme de normalisation.
ITU : *International Telecommunications Union* – Organisme de normalisation.
JPEG : *Joint Photographic Experts Group* – Groupe de travail de l'ISO.
JPEG : Algorithme de codage d'image fixe créé par le groupe JPEG.
JPEG-2000 : Algorithme de codage d'image fixe scalable.
JSVM : *Joint Scalable Video Model* – Logiciel de référence associé à la norme SVC.
Lifting : Structure de décomposition inversible.
MC-EZBC : *Motion-Compensated EZBC* – Algorithme de codage vidéo scalable.
MPEG : *Moving Picture Experts Group* – Groupe de travail de l'ISO.
MPEG-X : Famille d'algorithmes de codage vidéo normalisés par le groupe MPEG.
PSNR : *Peak Signal Noise Ratio* – Rapport de signal à bruit crête à crête.
QCIF : *Quarter CIF* – Format de résolution mesurant 176×144 pixels.
Scalable : Qui peut être représenté sur différents niveaux de précision.
Sous-bande : Signal résultant de la décomposition d'un signal par un banc de filtres.
SVC : *Scalable Video Coding* – Extension scalable de la norme H.264.
Vidwav : Groupe d'exploration MPEG sur les ondelettes.
Vidwav : Désigne aussi le logiciel de référence associé au groupe éponyme.
YSNR : Rapport de signal à bruit de crête, mesuré sur la composante de luminance Y.

Introduction générale

Introduction

Les progrès récents sur les schémas de codage vidéo par ondelettes ont permis l'apparition d'une nouvelle génération de codeurs vidéos scalables dont l'efficacité est comparable à celle des meilleurs codecs hybrides. Ces schémas sont qualifiés de $t+2D$ et reposent sur l'utilisation d'une transformée en ondelettes appliquée le long du mouvement des images afin d'exploiter leur redondance temporelle. Les sous-bandes résultantes sont alors décomposées spatialement et encodées par un codeur entropique.

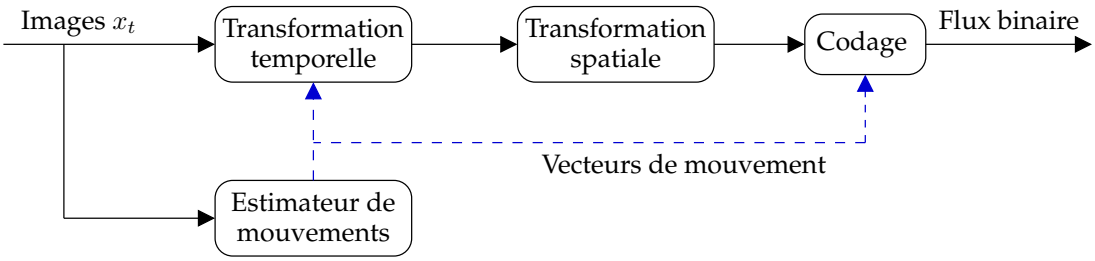


FIG. 0.1 – Schéma de principe d'un encodeur vidéo $t+2D$.

L'objectif de cette thèse consiste en l'étude et la construction de nouvelles transformées scalables mises en jeu dans le schéma de codage vidéo $t+2D$, afin d'améliorer le gain en compression. L'utilisation du formalisme lifting lors de la construction de ces transformées spatio-temporelles permet l'introduction d'opérateurs non-linéaires, particulièrement utiles pour représenter efficacement les singularités et discontinuités présentes dans une séquence vidéo. Nous nous intéressons particulièrement à :

- l'optimisation et la construction de nouvelles transformées temporelles compensées en mouvement, afin d'augmenter l'efficacité de codage objective et subjective ;
- l'élaboration et la mise en place de bancs de filtres M-bandes pour décomposer spatialement les sous-bande temporelles ;
- l'extension des propriétés de scalabilité du banc de synthèse M-bandes à des facteurs rationnels quelconques ;
- la construction de décompositions spatiales en ondelettes adaptatives, non-linéaires et inversibles, sans nécessiter la transmission d'une carte de décisions.

Cette thèse s'inscrit dans le développement de la norme de codage vidéo scalable SVC conduite par le consortium joint MPEG/ITU.

Organisation du manuscrit

Ce manuscrit s'articule en deux parties : la première est un état de l'art rappelant les bases de l'analyse multirésolution par ondelettes et décrivant les principes des schémas de codage vidéo scalables. La deuxième partie présente nos travaux de recherche, selon

les axes : optimisation de la transformée temporelle, décomposition spatiale par bancs de filtres M-bandes et construction de décompositions adaptatives non-linéaires.

Partie 1 – État de l'art

Chapitre 1 Ce chapitre rappelle les fondements de l'analyse multirésolution par **ondelettes** et dresse un panorama de **nouvelles représentations multirésolution**, permettant de fournir une description parcimonieuse et scalable des images.

Chapitre 2 Nous nous proposons de décrire les bases des principaux **schémas de codage vidéo scalable**. Nous décrivons dans un premier temps les extensions apportées au schéma de codage vidéo de type hybride pour permettre la scalabilité. Nous abordons ensuite les principes du schéma de codage vidéo par ondelettes de type $t + 2D$, utilisant une transformée en ondelettes temporelle appliquée dans le sens du mouvement des images.

Partie 2 – Mise en œuvre d'un codec vidéo scalable t+2D

Chapitre 3 L'utilisation du schéma lifting temporel permet la mise en œuvre simple d'une transformée temporelle, appliquée dans le sens du mouvement. Ce chapitre décrit la construction détaillée de la **transformée temporelle 5/3** compensée en mouvement et procède à l'étude systématique de ses propriétés.

Chapitre 4 Ce chapitre explore différentes stratégies d'**optimisation de la transformée temporelle** mise en jeu dans le schéma de codage vidéo $t + 2D$, visant l'amélioration objective et subjective de l'efficacité de codage. Des travaux sur la minimisation de la latence engendrée par la transformée temporelle sont aussi rapportés.

Chapitre 5 De part leur flexibilité et leur grande sélectivité fréquentielle, les **bancs de filtres M-bandes** sont des candidats idéaux à la **décomposition spatiale** des sous-bandes temporelles impliquées dans le schéma $t + 2D$. Nous présentons dans ce chapitre la construction d'une telle transformée et observons le gain en efficacité de codage obtenu. Nous décrivons enfin comment la modification d'un banc de synthèse M-bandes permet d'obtenir un schéma capable d'offrir des facteurs de scalabilité rationnels.

Chapitre 6 La transformation en ondelettes séparable n'est pas adaptée à la représentation économique d'images où les singularités et contours sont nombreux. Nous présentons dans ce chapitre des **transformées non-linéaires adaptatives** basées sur une structure lifting, où l'opérateur de mise à jour est modifié en fonction des caractéristiques locales du signal. Aucune information de décision n'est transmise dans le flux compressé et nous montrons les conditions nécessaires et suffisantes pour assurer la reconstruction parfaite.

Enfin, nous avons reproduit en Annexe B un article de revue légèrement en dehors de nos axes conducteurs, relatant nos travaux sur des filtres temporels M-bandes mis en œuvre au sein du codec H.264 afin de lui fournir des propriétés de scalabilité temporelle.

Première partie

Ondelettes et codage vidéo scalable : un état de l'art

Chapitre 1

Ondelettes dyadiques et nouvelles représentations multirésolution

Avec l'explosion de la diversité des modes de consommation de contenus multimédia, il est souvent souhaitable de disposer d'un même média dans des résolutions et des qualités différentes. Les exemples sont légion : sur un site web, les imagettes nous donnent un aperçu des images en grand format et permettent ainsi une présélection rapide du contenu. De même, on souhaitera disposer d'un morceau de musique numérisé de haute qualité lors d'un concert et on lui préférera une version prenant moins d'espace lors d'une écoute distraite sur baladeur.

En pratique, on utilise souvent plusieurs fichiers pour représenter différentes versions d'un même contenu multimédia. Cette solution coûteuse en bande passante n'est cependant pas satisfaisante. La scalabilité est la caractéristique d'un objet ou d'un algorithme à être représentable sur plusieurs niveaux de résolution ou de qualité. Nous justifions et motivons dans la section 1.1 comment cette propriété peut être mise à profit pour créer un *seul* flux compressé capable de représenter plusieurs versions d'un même contenu multimédia.

L'analyse multirésolution et la transformée en ondelettes sont des outils mathématiques capables de fournir une représentation scalable d'un signal. Tout d'abord, nous rappelons dans la section 1.2 les fondements mathématiques de ces outils. Nous décrivons ensuite comment la transformée en ondelettes peut être utilisée conjointement avec un schéma de codage emboîté afin de fournir un algorithme de compression d'image scalable et performant.

Il existe cependant d'autres transformations capables de fournir une représentation scalable d'un signal. La structure lifting, rappelée en section 1.3 permet d'étendre la théorie des ondelettes dans un cadre non-linéaire et autorise simplement la construction de transformées non-linéaires et inversibles. Nous étudions ensuite pourquoi les bases d'ondelettes séparables 2D ne sont pas bien adaptées à la représentation des images. Nous dressons alors un inventaire d'ondelettes géométriques fixes ou adaptatives appartenant à la famille des *Xlets*, permettent de contourner ce problème. Nous présentons enfin d'autres représentations multirésolution récentes, ayant pour but de fournir des bases mieux adaptées à la description scalable de signaux discontinus.

1.1 Introduction aux représentations multirésolution

Sur la scalabilité

Le terme scalabilité est un néologisme directement emprunté de l'anglais *scalability* qui peut approximativement être traduit par le terme échelonnabilité. La scalabilité est la

caractéristique d'un objet ou d'un signal à être représentable sur plusieurs niveaux de résolution ou de qualité. Une transformation sera ainsi dite scalable si elle est en mesure de représenter un signal sur plusieurs niveaux de résolution ou de qualité.

La notion de scalabilité est en fait très générale et il existe plusieurs types de scalabilité. Dans le cas d'un signal monodimensionnel, on parlera de scalabilité en résolution pour désigner le fait qu'un signal puisse être décrit par un nombre variable d'échantillons. Dans le cas d'une image, la scalabilité spatiale qualifie la propriété de pouvoir représenter une image sur plusieurs niveaux de résolution spatiale, comme illustré en Fig. 1.1.

$1/4$ $1/2$ $1/1$

FIG. 1.1 – Scalabilité spatiale. Exemples de facteurs de résolution dyadiques obtenus avec le codec scalable JPEG-2000.

Il est aussi possible de représenter un signal sur différents niveaux de qualité, où chaque échantillon ou coefficient peut être décrit avec une précision plus ou moins grande. On parlera dans ce cas de scalabilité en qualité. La Fig. 1.2 montre un exemple de scalabilité en qualité où chaque point de l'image est décrit avec plus ou moins de précision en fonction du débit qui lui est accordé.

$d = 0.04$ bpp $d = 0.1$ bpp $d = 0.2$ bpp

FIG. 1.2 – Scalabilité en qualité. Exemples de différentes qualités obtenues avec le codec JPEG-2000 lors du décodage à différents débits, exprimés en bits par pixel.

Il existe d'autres types de scalabilité : dans le cas d'une séquence vidéo, on parlera de scalabilité temporelle pour désigner la propriété de pouvoir la représenter à plusieurs fréquences temporelles, exprimées en nombre d'images par seconde. D'autres types de scalabilité peuvent être définis comme la scalabilité en complexité, en objets ou en délai mais nous ne les aborderons pas dans ce document.

Motivation et cas d'utilisation

Avec l'explosion des applications multimédia et le besoin croissant de diffusion de contenu à destination de récepteurs hétérogènes, la scalabilité est devenue indispensable dans la conception d'un schéma de compression d'image ou de codage vidéo. Cette propriété permet ainsi de pouvoir diffuser un *unique* flux vidéo compressé, capable d'être adapté par les nœuds d'un réseau ou d'être décodé par une grande variété de récepteurs.

Il existe de nombreux cas d'utilisation nécessitant une description scalable et parcimonieuse d'un contenu multimédia, relevant pour la plupart du domaine de l'adaptation de contenu. Par exemple, les images présentes sur Internet sont souvent disponibles sous deux voire trois résolutions (aperçu *thumbnail*, résolution moyenne et haute résolution) en fonction de la façon dont elles sont visualisées. De plus, il est souvent nécessaire de posséder un morceau de musique compressé sous plusieurs débits, en fonction de la qualité désirée et de la place disponible. Enfin, les opérateurs commerciaux de diffusion de contenus multimédia ont tout intérêt à utiliser un format scalable. Un opérateur de téléphonie mobile pourra ainsi diffuser un flux vidéo TV destiné à un parc hétérogène de récepteurs dont les écrans sont de tailles différentes.

De plus, la scalabilité est une propriété très utile lors de la diffusion de contenu multimédia dans un environnement enclin aux erreurs de transmissions, comme les réseaux IP sans fil. En effet, elle permet l'adaptation du débit du flux compressé en fonction de la capacité du canal, susceptible de varier selon les conditions de transmission, et augmente la robustesse d'un schéma de codage en cas de pertes, d'erreurs ou d'encombrements.

Ces nombreux cas d'utilisation poussent depuis quelques années les organismes internationaux de normalisation ITU, JPEG et MPEG à concevoir des algorithmes de compression d'images et de codage vidéo *scalables*. La norme de compression d'images fixes JPEG-2000, scalable en résolution et en qualité a ainsi été normalisée en 2000. Le futur algorithme de codage vidéo scalable SVC décrit dans la section 2.1.4, est quant à lui en cours de normalisation. On trouvera dans l'appel à proposition [6] qui a précédé sa création, les nombreuses motivations industrielles et cas d'utilisation auxquels il répond.

Techniques classiques de description scalable

Comment créer une représentation scalable d'un signal ? On peut tout d'abord penser naïvement à une solution de type Simulcast. C'est une stratégie brutale qui consiste simplement à proposer *simultanément* plusieurs versions du même contenu multimédia. C'est en effet une forme de description scalable mais qui est loin d'être parcimonieuse.

Une autre classe de solutions existe : les schémas de codage prédictif en couches. Dans ce type de stratégie, chaque version du contenu multimédia constitue une couche et un mécanisme existe pour permettre la prédiction d'une couche à partir d'une autre, réduisant ainsi la redondance comparée à une stratégie Simulcast. Un exemple simple de schéma de codage prédictif en couches peut être imaginé par un format de musique compressé, où chaque couche représente une qualité différente, obtenue par différence avec la couche de qualité inférieure. Cependant, l'efficacité de ce type de représentation repose principalement sur l'opérateur de prédiction utilisé entre couches et est susceptible de chuter si le nombre de couches est trop important. Enfin, la scalabilité offerte par les schémas en couches est grossière et statique : seules les couches disponibles peuvent fournir une version du contenu compressé et ces dernières doivent être connues au moment de l'encodage.

La transformée en ondelettes est par construction même, une transformation capable de donner une représentation scalable d'un signal. En effet, les coefficients issus de la transformée en ondelettes donnent une représentation du signal sur plusieurs niveaux de résolution, du plus grossier au plus fin. Nous rappelons dans la section suivante les bases de l'analyse par ondelettes.

1.2 Ondelettes dyadiques

L'analyse multirésolution par ondelettes d'un signal, d'une image ou plus généralement d'une fonction $f \in L^2(\mathbb{R}^N)$ consiste en sa projection sur des bases de fonctions, donnant des approximations de moins en moins fines de la fonction originale. Nous rappelons tout d'abord dans cette section le concept d'analyse multirésolution par ondelettes et voyons ensuite comment un algorithme de transformée rapide en ondelettes peut être mis en œuvre sous forme de banc de filtres. Après avoir établi un panorama d'ondelettes classiquement utilisées en compression d'image, nous décrivons alors plusieurs algorithmes de codage de coefficients d'ondelettes utilisés dans ce domaine.

1.2.1 Bases d'ondelettes

Analyse multirésolution

La construction d'espaces multirésolution aptes à représenter plus ou moins grossièrement une fonction $f \in L^2(\mathbb{R})$ a été proposée par Mallat et Meyer [79, 80, 89] pour fournir un cadre formel permettant l'analyse d'une fonction f sur plusieurs niveaux de résolution. On définit une *approximation multirésolution* comme une suite de sous-espaces vectoriels fermés $\{\boldsymbol{V}_j\}_{j\in\mathbb{Z}}$ de $L^2(\mathbb{R})$ emboîtés selon la relation :

$$\varnothing \subset \ldots \subset \boldsymbol{V}_2 \subset \boldsymbol{V}_1 \subset \boldsymbol{V}_0 \subset \boldsymbol{V}_{-1} \ldots \subset L^2(\mathbb{R}) \tag{1.1}$$

La projection d'une fonction $f \in L^2(\mathbb{R})$ sur un espace $\boldsymbol{V}_j$ représente alors une *approximation* de f au niveau de résolution j. Du fait de l'emboîtement des espaces $\{\boldsymbol{V}_j\}$, l'approximation de niveau j sera nécessairement plus précise que celle du niveau $j+1$ car l'espace $\boldsymbol{V}_j$ dispose de plus de fonctions que l'espace $\boldsymbol{V}_{j+1}$ pour représenter f.

On suppose alors l'existence d'une fonction $\phi \in L^2(\mathbb{R})$, appelée *fonction d'échelle* ou *ondelette père* telle que ses translatées $\{t \mapsto \phi(t-k)\}_{k\in\mathbb{Z}}$ forment une base orthonormale de $\boldsymbol{V}_0$. On affirme enfin que les fonctions de $\boldsymbol{V}_{j+1}$ sont obtenues par dilatation d'un facteur 2 des fonctions de $\boldsymbol{V}_j$ selon la relation :

$$\forall\, j \in \mathbb{Z},\ t \mapsto f(t) \in \boldsymbol{V}_j \ \Leftrightarrow\ t \mapsto f\Big(\frac{t}{2}\Big) \in \boldsymbol{V}_{j+1} \tag{1.2}$$

permettant ainsi de caractériser intuitivement les propriétés de l'analyse multirésolution et de supputer que l'approximation de f sur $\boldsymbol{V}_{j+1}$ est deux fois plus grossière que celle sur $\boldsymbol{V}_j$. On peut alors introduire la notion d'échelle et définir la projection de f sur $\boldsymbol{V}_j$ comme l'approximation de f à l'échelle 2^j, où j est le niveau de résolution.

L'utilisation de la relation de dilatation (1.2) nous permet alors d'affirmer que les fonctions $\{\phi_{j,k}\}_{k\in\mathbb{Z}}$ obtenues par dilatations et translations de ϕ et définies par :

$$\phi_{j,k} = t \mapsto \frac{1}{2^{j/2}}\phi\Big(\frac{t}{2^j} - k\Big),\ k \in \mathbb{Z} \tag{1.3}$$

forment une base orthonormale de $\boldsymbol{V}_j$.

Bases d'ondelettes orthogonales

La relation d'emboîtement implique que les projections de f sur $\boldsymbol{V}_j$ sont de plus en plus grossières, au fur et à mesure que j croît. La différence entre l'approximation sur $\boldsymbol{V}_j$ et celle sur $\boldsymbol{V}_{j+1}$ représente ainsi l'information de détail perdue par incrémentation du niveau de résolution j. Il est cependant possible de définir l'espace de détail $\boldsymbol{W}_{j+1}$ contenant les fonctions nécessaires à représenter cette information perdue, en utilisant l'opérateur de sommation directe $\oplus$ de sous-espaces vectoriels :

$$\boldsymbol{V}_j = \boldsymbol{V}_{j+1} \oplus \boldsymbol{W}_{j+1} \tag{1.4}$$

On peut alors montrer l'existence d'une fonction ψ appelée *ondelette mère* telle que ses translatées $\{t \mapsto \psi(t-k)\}_{k\in\mathbb{Z}}$ forment une base orthonormale de $\boldsymbol{W}_0$. On montre de même que les fonctions $\{\psi_{j,k}\}_{k\in\mathbb{Z}}$ définies par :

$$\psi_{j,k} = t \mapsto \frac{1}{2^{j/2}}\psi\Big(\frac{t}{2^j} - k\Big),\ k \in \mathbb{Z} \tag{1.5}$$

forment une base orthonormale de $\boldsymbol{W}_j$. Enfin, en exploitant les conditions limites de l'analyse multirésolution, on conclut que l'ensemble des fonctions $\{\psi_{j,k}\}_{(j,k)\in\mathbb{Z}^2}$ forme une base d'ondelettes orthogonales de $L^2(\mathbb{R})$.

Si f est une fonction discrète alors pour toute fonction ϕ, il existe un niveau de résolution j suffisamment petit tel que f appartienne à $\boldsymbol{V}_j$. On peut donc translater le niveau de résolution et fixer $j = 0$ pour que f appartienne à $\boldsymbol{V}_0$. La transformée en ondelettes d'une fonction $f \in \boldsymbol{V}_0$ sur n niveaux est alors définie comme la projection de cette fonction sur les espaces $\boldsymbol{V}_n$ et $\{\boldsymbol{W}_j\}_{1\leq j\leq n}$ car $\boldsymbol{V}_0 = \boldsymbol{V}_n \oplus \Big[\bigoplus_{j=1}^{n} \boldsymbol{W}_j\Big]$. Les coefficients de projection sur $\boldsymbol{V}_j$ sont notés $a_j[k]$ et nommés *coefficients d'approximation* tandis que ceux sur $\boldsymbol{W}_j$ sont notés $d_j[k]$ et nommés *coefficients d'ondelette* ou *coefficients de détail*. On a alors :

$$a_j[k] = \langle f, \phi_{j,k}\rangle \tag{1.6}$$

$$d_j[k] = \langle f, \psi_{j,k}\rangle \tag{1.7}$$

où $\langle\cdot,\cdot\rangle$ représente le produit scalaire dans $L^2(\mathbb{R})$.

Ces relations nous permettent de calculer explicitement les coefficients de la transformée en ondelettes de f sur n niveaux. Cependant, l'intégration sur $\mathbb{R}$ qu'elles nécessitent les rendent très lourdes à utiliser. Nous verrons dans la section 1.2.2 qu'il est possible de construire un algorithme de calcul rapide des coefficients $a_j[k]$ et $d_j[k]$.

Enfin, on remarquera que la transformée en ondelettes est une application linéaire, inversible et orthogonale. C'est donc une isométrie qui préserve la norme ℓ_2, c'est à dire l'énergie d'un signal. On a alors $\sum_k a_j[k]^2 = \sum_k a_{j+1}[k]^2 + d_{j+1}[k]^2$.

1.2.2 Ondelettes et bancs de filtres

Filtres miroirs conjugués

L'espace $\boldsymbol{V}_{j+1}$ étant un sous-espace vectoriel de $\boldsymbol{V}_j$, les fonctions de $\boldsymbol{V}_{j+1}$ peuvent être écrites comme une combinaison linéaire de fonctions de $\boldsymbol{V}_j$. On peut donc exprimer la fonction $t \mapsto \frac{1}{\sqrt{2}}\phi(\frac{t}{2})$ appartenant à $\boldsymbol{V}_0$ comme une combinaison linéaire des fonctions $\{t \mapsto \phi(t-k)\}_{k\in\mathbb{Z}}$ en introduisant la suite $h_0[k]$, $k \in \mathbb{Z}$:

$$\frac{1}{\sqrt{2}}\phi\Big(\frac{t}{2}\Big) = \sum_{k=-\infty}^{\infty} h_0[k]\phi(t-k) \tag{1.8}$$

De même, l'espace $\boldsymbol{W}_{j+1}$ étant un sous-espace vectoriel de $\boldsymbol{V}_j$, il est possible de définir la fonction $t \mapsto \frac{1}{\sqrt{2}}\psi(\frac{t}{2})$ comme une combinaison linéaire des fonctions $\{t \mapsto \phi(t-k)\}_{k\in\mathbb{Z}}$ en introduisant la suite $h_1[k]$, $k \in \mathbb{Z}$:

$$\frac{1}{\sqrt{2}}\psi\Big(\frac{t}{2}\Big) = \sum_{k=-\infty}^{\infty} h_1[k]\phi(t-k) \tag{1.9}$$

Les relations (1.8) et (1.9) sont aussi appelées *équations à deux échelles*. Enfin, la définition des suites h_0 et h_1 permet de montrer [79] qu'une condition suffisante assurant l'existence de ψ peut s'exprimer par la relation suivante, aussi appelée condition d'orthogonalité :

$$h_1[n] = (-1)^{1-n}h_0[1-n] \tag{1.10}$$

Transformée en ondelettes rapide

Mallat a montré [79] l'existence d'équations liant les coefficients d'approximation $a_j[k]$ et les coefficients d'ondelettes $d_j[k]$ obtenus entre deux niveaux de résolution consécutifs. En effet, par combinaison de (1.6), (1.7), (1.8) et (1.9), on vérifie aisément que :

$$a_{j+1}[k] = \sum_{n=-\infty}^{\infty} h_0[n-2k]a_j[n] = a_j \star \overline{h}_0[2k] \tag{1.11}$$

$$d_{j+1}[k] = \sum_{n=-\infty}^{\infty} h_1[n-2k]a_j[n] = a_j \star \overline{h}_1[2k] \tag{1.12}$$

où $\star$ est le produit de convolution et $\overline{h}$ dénote le retournement temporel du filtre h, où pour tout n, $\overline{h}[n] = h[-n]$. Les relations (1.11) et (1.12) permettent ainsi de calculer les coefficients de projection $a_{j+1}[k]$ et $d_{j+1}[k]$ à partir des seuls coefficients $a_j[k]$ et des suites h_0 et h_1 précédemment définies.

La présence du produit de convolution nous suggère l'utilisation d'un opérateur de filtrage, classiquement utilisé en traitement du signal. Les équations (1.11) et (1.12) font ainsi le lien entre la transformée en ondelettes définie précédemment comme la projection d'un signal dans les espaces $\boldsymbol{V}_j$ et $\{\boldsymbol{W}_j\}_{1\leq j\leq n}$ et son interprétation en termes de bancs de filtres. Les séquences $h_0[k]$ et $h_1[k]$ peuvent alors s'identifier respectivement aux réponses impulsionnelles d'un filtre passe-bas et d'un filtre passe-haut d'un banc d'analyse.

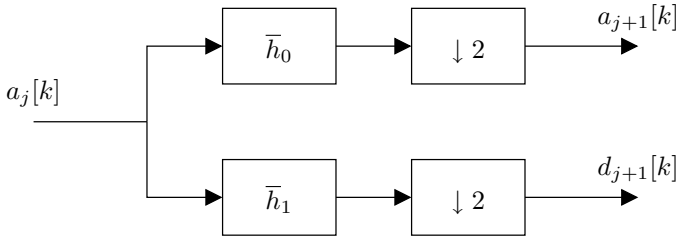


FIG. 1.3 – Banc de filtres d'analyse en quadrature miroir.

Il est alors envisageable de construire un algorithme de calcul rapide des coefficients $a_{j+1}[k]$ et $d_{j+1}[k]$ par filtrage des coefficients $a_j[k]$, d'une part par le filtre $\overline{h}_0$ et d'autre part par le filtre $\overline{h}_1$, suivi par la décimation d'un facteur 2. Cette dernière opération, aussi appelée sous-échantillonnage d'un facteur 2, est notée $[\downarrow 2]$ et consiste à se débarrasser d'un

coefficient sur deux. L'algorithme peut être représenté par un banc de filtres d'analyse en quadrature miroir [155] et est illustré par la Fig. 1.3.

Ce banc de filtres d'analyse permet ainsi l'implémentation effective de la transformée en ondelettes rapide. En effet, en supposant connus les filtres h_0 et h_1, la décomposition en ondelettes d'un signal x d'une longueur de n échantillons consiste à initialiser[1] $a_0[k] = x[k]$ et à utiliser le banc de filtres. On obtient alors les sous-bandes a_1 et d_1, comportant chacune $n/2$ échantillons. Ces signaux sont les coefficients de la projection de x sur les espaces $\boldsymbol{V}_1$ et $\boldsymbol{W}_1$. L'analyse multirésolution sur un nombre supérieur de niveaux s'obtient par la décomposition successive des signaux a_j et donc par une mise en cascade du banc de filtres jusqu'au niveau j_{max} désiré, comme illustré par la Fig. 1.4.

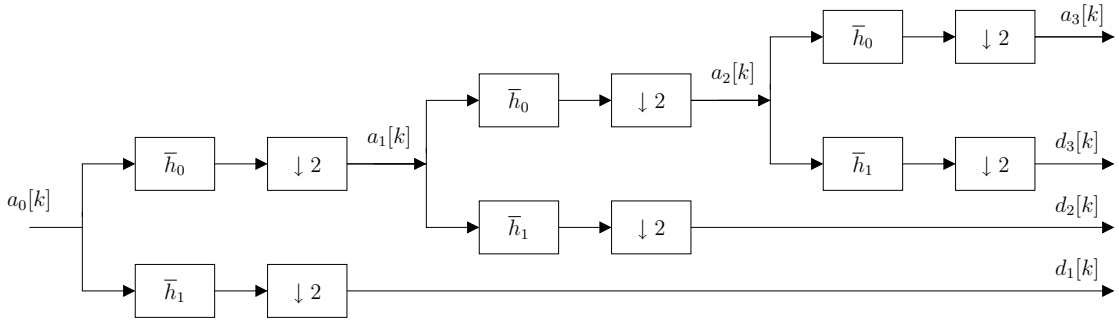


FIG. 1.4 – Banc de filtres d'analyse assurant une décomposition en ondelettes sur $j_{max} = 3$ niveaux de résolution. Elle correspond à une projection sur $\boldsymbol{V}_3$, $\boldsymbol{W}_3$, $\boldsymbol{W}_2$ et $\boldsymbol{W}_1$.

L'algorithme présenté permet ainsi de calculer rapidement la transformée en ondelettes d'un signal donné, sous la connaissance des filtres h_0 et h_1. Sa complexité est de $\mathcal{O}(n)$ opérations élémentaires où n est la taille du signal, rivalisant ainsi avec la transformée de Fourier rapide. Cependant, sa mise en œuvre sur un signal fini nécessite généralement l'utilisation d'une convolution périodique qui créé des coefficients d'ondelettes de large amplitude sur les bords du signal et nuit ainsi légèrement à l'efficacité de décorrélation. Les ondelettes de bords [79] et surtout la structure lifting présentée en section 1.3.1 permet de s'affranchir simplement de ce problème.

Reconstruction par transformée inverse

Comme pour la transformée directe décrite par les équations (1.11) et (1.12), il est possible de montrer la relation suivante, utile pour la reconstruction du signal original :

$$\begin{aligned} a_j[p] &= \sum_{n=-\infty}^{\infty} h_0[p-2n]a_{j+1}[n] + \sum_{n=-\infty}^{\infty} h_1[p-2n]d_{j+1}[n] \\ &= \widehat{a}_{j+1} \star h_0 + \widehat{d}_{j+1} \star h_1 \end{aligned} \tag{1.13}$$

où $\widehat{h}$ est le signal résultant du suréchantillonnage de h d'un facteur 2. Cette opération consiste en l'introduction de zéros entre les échantillons du signal d'origine : elle est définie pour tout n par $\widehat{h}[2n] = h[n]$ et $\widehat{h}[2n+1] = 0$, et se note $[\uparrow 2]$.

Cette relation nous permet alors de construire un algorithme rapide de reconstruction du signal a_j à partir de ses coefficients d'approximation $a_{j+1}[k]$ et de ses coefficients

[1] Nous ne détaillerons pas ici les conditions sur le signal continu sous lesquelles cette relation est vraie.

d'ondelette $d_{j+1}[k]$ du niveau supérieur. En prenant $a_0[k] = x[k]$, on montre ainsi qu'on peut reconstruire parfaitement le signal $x[k]$ à partir de ses coefficients $a_1[k]$ et $d_1[k]$. L'algorithme de reconstruction peut être représenté par un banc de filtres, nommé banc de filtres de synthèse et est illustré en Fig. 1.5.

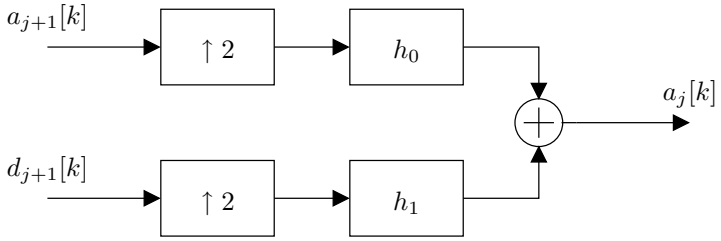


FIG. 1.5 – Banc de filtres de synthèse.

Nous avons donc mis en évidence l'existence d'algorithmes rapides de transformation en ondelettes et de reconstruction sous formes de banc de filtres. Leur mise en œuvre nécessitent la seule connaissance du filtre passe-bas h_0, le filtre passe-haut h_1 étant obtenu grâce à la condition d'orthogonalité (1.10).

Cas 2D et ondelettes séparables

Les résultats précédents ont été obtenus pour des fonctions monodimensionnelles appartenant à $L^2(\mathbb{R})$. L'extension aux dimensions supérieures peut se faire simplement par utilisation du produit tensoriel de bases d'ondelettes monodimensionnelles, conduisant à des bases d'ondelettes séparables. Dans le cas 2D, la transformée rapide en ondelettes consiste alors en une transformée en ondelettes 1D des colonnes puis des lignes de l'image (ou inversement). Un exemple d'une telle décomposition est illustré en Fig. 1.11.

Bases d'ondelettes adaptées à la compression d'un signal

Comment construire une base d'ondelettes adaptée à la compression d'un signal ? Il existe en effet de nombreuses fonctions d'échelles ϕ et donc d'ondelettes mère ψ, vérifiant les propriétés nécessaires à la construction de bases d'ondelettes. En fait, nous nous intéressons à la compression d'images et de séquences vidéos et souhaitons disposer ainsi d'une base permettant une représentation *parcimonieuse* d'un signal, c'est à dire donnant peu de coefficients d'ondelettes de grande amplitude. Il est donc souhaitable d'imposer des contraintes sur l'ondelette mère ψ afin de favoriser son aptitude à décorréler ce type de signaux.

Tout d'abord, il est fortement souhaitable que l'ondelette ψ soit à support fini. Ceci permet en effet la mise en œuvre simple de la transformée en ondelettes rapide. La symétrie de l'ondelette est aussi un critère important en compression d'images, permettant de donner un poids équivalent aux pixels lors de leur traitement et de préserver la linéarité de la phase.

Enfin, il est utile que l'ondelette possède un grand nombre de moments nuls. Ce paramètre important caractérise l'aptitude d'une ondelette à approximer les polynômes. On dit que ψ possède N moments nuls si et seulement si :

$$\forall\ 0 \leq n < N \quad \int_{\mathbb{R}} \psi(t) t^n dt = 0 \tag{1.14}$$

Ceci signifie que ψ est orthogonale à tout polynôme de degré inférieur ou égal à $N-1$. Ainsi, si f est un signal localement polynomial de degré inférieur ou égal à $N-1$, alors les coefficients de détail $d_j[k]$ résultant de la transformation en ondelettes seront localement nuls. Comme une image est bien modélisée par des fonctions polynomiales par morceaux, sa transformée en ondelettes avec un nombre suffisant de moments nuls est susceptible de contenir de nombreux coefficients d'ondelettes proches de zéro, correspondants aux régions où l'image présente un comportement polynomial.

Banc de filtres à reconstruction parfaite et ondelettes biorthogonales

Nous avons abordé dans les sections précédentes le cas des ondelettes orthogonales liées par la condition (1.10). La seule connaissance du filtre h_0 nous a ainsi permis de mettre en œuvre un algorithme de transformée rapide en ondelettes. Est-il cependant possible de construire une transformée par bancs de filtres plus générale en omettant cette condition ?

Considérons la structure décrite en Fig. 1.6, combinant un banc de filtres d'analyse h_0 et h_1 et un banc de synthèse dont les réponses impulsionnelles sont $\widetilde{h}_0$ et $\widetilde{h}_1$. Ces quatre filtres sont volontairement supposés indépendants. On s'intéresse aux conditions nécessaires et suffisantes, nommées conditions de reconstruction parfaite, liant ces filtres et assurant que le signal reconstruit $\widetilde{x}$ soit strictement égal au signal d'entrée x.

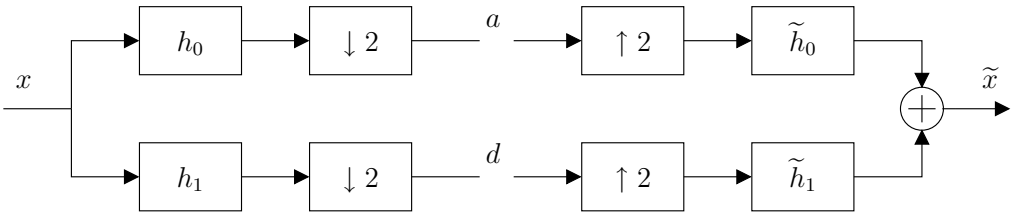


FIG. 1.6 – Banc de filtres d'analyse-synthèse.

L'utilisation de la transformée de Fourier permet de formuler simplement le problème. En notant $\widehat{h}(f)$ la transformée de Fourier en fréquence normalisée f du filtre h, on montre que les conditions de reconstruction parfaite s'écrivent dans le domaine fréquentiel :

$$\widehat{h}_0(f)\widehat{\widetilde{h}}_0(f) + \widehat{h}_1(f)\widehat{\widetilde{h}}_1(f) = 2 \tag{1.15}$$

$$\widehat{h}_0(f+1/2)\widehat{\widetilde{h}}_0(f) + \widehat{h}_1(f+1/2)\widehat{\widetilde{h}}_1(f) = 0 \tag{1.16}$$

Ces équations imposent donc des conditions sur les réponses fréquentielles des filtres h_0, h_1, $\widetilde{h}_0$ et $\widetilde{h}_1$, et autorisent leur construction uniquement dans le domaine de Fourier. Les conditions de reconstruction parfaite peuvent cependant s'écrire aussi dans le domaine temporel sous la forme concise suivante, en utilisant l'opérateur de Kronecker δ :

$$\forall\, 0 \leq i,j \leq 1,\ \forall\, n \in \mathbb{Z}, \quad \sum_k h_i[k]\widetilde{h}_j[k-2n] = \delta_n \delta_{i-j} \tag{1.17}$$

En supposant la connaissance des filtres passe-bas d'analyse h_0 et de synthèse $\widetilde{h}_0$, les conditions de reconstruction parfaite imposent les coefficients des filtres passe-haut :

$$h_1[n] = (-1)^{1-n}\widetilde{h}_0[1-n] \tag{1.18}$$

$$\widetilde{h}_1[n] = (-1)^{1-n}h_0[1-n] \tag{1.19}$$

et permettent ainsi de caractériser entièrement la transformée par ses filtres passe-bas d'analyse h_0 et de synthèse $\widetilde{h}_0$. De plus, ces conditions impliquent que les familles $\{h_0[k-2n],\ h_1[k-2n]\}_{n\in\mathbb{Z}}$ et $\{\widetilde{h}_0[k-2n],\ \widetilde{h}_1[k-2n]\}_{n\in\mathbb{Z}}$ soient biorthogonales entre elles, ce qui permet de donner une interprétation en termes d'ondelettes au banc de filtres à reconstruction parfaite. Elle revient à lever les contraintes d'orthogonalité imposées aux bases $\boldsymbol{V}_j$ et $\boldsymbol{W}_j$ à les remplacer par des contraintes de biorthogonalité. Ces dernières conduisent alors à l'introduction de bases duales $\widetilde{\boldsymbol{V}}_j$ et $\widetilde{\boldsymbol{W}}_j$ et de leur fonctions duales $\widetilde{\phi}$ et $\widetilde{\psi}$ associées, utilisées lors de la reconstruction. On parlera alors d'ondelettes biorthogonales et de transformée en ondelettes biorthogonales.

1.2.3 Panorama d'ondelettes dyadiques utilisées en codage d'image

Il existe de nombreuses ondelettes dyadiques décrites dans la littérature [79, 109] (Spline, Shannon-Nyquist, Daubechies, etc...) utilisées en codage, débruitage ou analyse de signaux. Nous présentons ici quelques ondelettes couramment utilisées en codage d'image.

Ondelette de Haar

L'ondelette de Haar est une ondelette orthogonale symétrique possédant un seul moment nul $N = 1$ et un support de $p = 2$ échantillons. Les coefficients de la réponse impulsionnelle de son filtre passe-bas h_0 sont présentés dans la partie gauche du Tab. 1.1. Daubechies a montré que c'est la seule ondelette orthogonale et symétrique correspondant à un banc de filtres à réponse impulsionnelle finie. De part sa simplicité, cette ondelette illustrée en Fig. 1.7 est assez utilisée en codage d'image.

n	$h_0[n]$
0	0.70710678118655
1	0.70710678118655

n	$h_0[n]$
0	0.48296291314483
1	0.83651630373771
2	0.22414386804192
3	-0.12940952255095

TAB. 1.1 – Coefficients de la réponse impulsionnelle du filtre passe-bas d'analyse $h_0[n]$ associé à l'ondelette de Haar (gauche) et à l'ondelette de Daubechies-4 (droite).

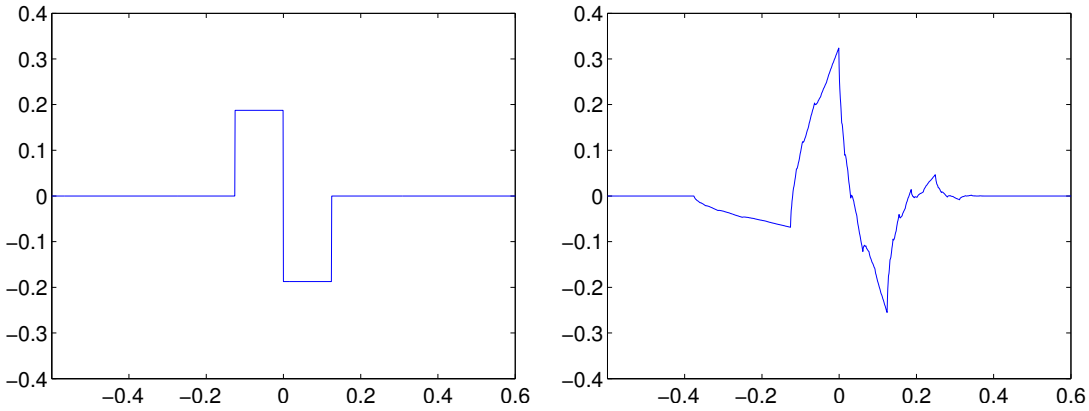


FIG. 1.7 – Ondelette de Haar (gauche) et ondelette Daubechies-4 (droite).

Ondelette Daubechies-4

L'ondelette Daubechies-4 fait partie de la famille des ondelettes orthogonales de Daubechies, possédant un support de $2N$ échantillons pour N moments nuls. L'ondelette Daubechies-4 possède donc $N = 2$ moments nuls et un support de $p = 4$ échantillons. Les coefficients de la réponse impulsionnelle de son filtre passe-bas h_0 sont dressés dans la partie droite du Tab. 1.1. On peut montrer que les ondelettes de cette famille ont une largeur de support minimale et un déphasage minimal pour un nombre de moments nuls donné. Elles sont pourtant peu utilisées en codage car elles sont fortement asymétriques et très irrégulières. Cependant, l'ondelette Daubechies-4 illustrée en Fig. 1.7 est courte, orthogonale et suffisamment régulière pour susciter un intérêt en codage.

Ondelettes biorthogonales 5/3

Les ondelettes biorthogonales 5/3 font partie de la famille des ondelettes biorthogonales symétriques de Cohen-Daubechies-Feauveau (CDF) [79]. Elles sont dénommées ainsi car la largeur du support de leurs filtres passe-bas, détaillés dans le Tab. 1.2, est de $p = 5$ échantillons à l'analyse et $\widetilde{p} = 3$ à la synthèse. De plus, elles possèdent $N = \widetilde{N} = 2$ moments nuls. De part leur relative simplicité et la symétrie qu'elles offrent, les ondelettes 5/3 présentées en Fig. 1.8 sont assez utilisées en codage d'image.

Les ondelettes de cette famille sont aussi dénommées CDF $(N, \widetilde{N})$, où N désigne le nombre de moments nuls de l'ondelette d'analyse ψ et $\widetilde{N}$ son équivalent à la synthèse. Comme pour les ondelettes de Daubechies, il est possible de montrer que les ondelettes CDF ont un support minimal pour un nombre de moments nuls $(N, \widetilde{N})$ donnés.

n	$h_0[n]$	$\widetilde{h}_0[n]$
0	1.06066017177982	0.70710678118655
1	0.35355339059327	0.35355339059327
2	-0.17677669529664	

TAB. 1.2 – Coefficients des réponses impulsionnelles symétriques des filtres passe-bas d'analyse $h_0[n]$ et de synthèse $\widetilde{h}_0[n]$ associés aux ondelettes CDF 5/3.

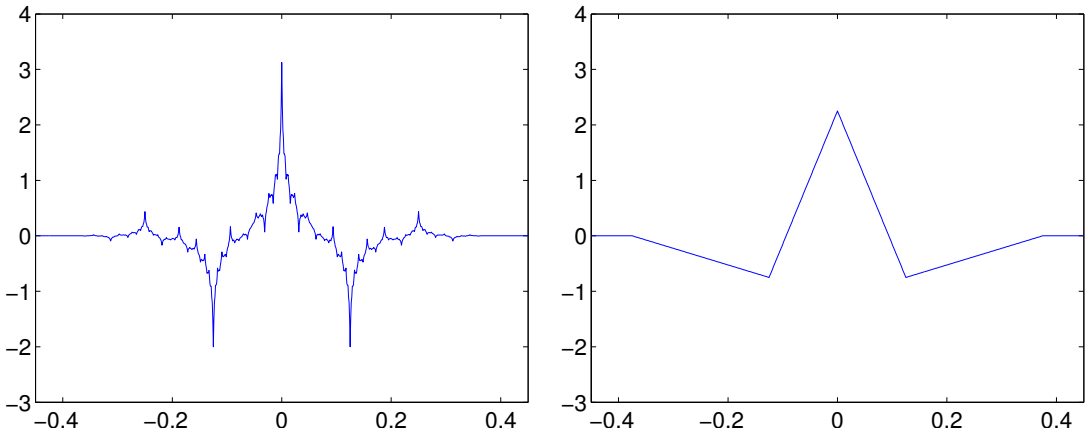


FIG. 1.8 – Ondelette CDF 5/3 d'analyse ψ et sa duale $\widetilde{\psi}$.

Ondelettes biorthogonales 9/7

Tout comme les ondelettes 5/3, les ondelettes biorthogonales 9/7 font partie de la famille des ondelettes biorthogonales symétriques CDF. Les filtres passe-bas associés aux ondelettes 9/7 possèdent ainsi $p = 9$ coefficients à l'analyse, $p = 7$ coefficients à la synthèse et sont décrits dans le Tab. 1.3. Les ondelettes biorthogonales 9/7 sont illustrées en Fig. 1.9 et possèdent $N = 4$ moments nuls à l'analyse et $\widetilde{N} = 4$ à la synthèse.

Les ondelettes 9/7 possèdent un grand nombre de moments nuls pour un support relativement court. Elles sont de plus symétriques et très proches de l'orthogonalité. C'est une caractéristique importante en codage qui lui permet d'assurer que l'erreur de reconstruction soit très proche de l'erreur de quantification, en terme d'erreur quadratique moyenne. Antonini et Barlaud furent les premiers [18] à montrer la supériorité de la transformée en ondelettes biorthogonale 9/7 pour la décorrélation d'images naturelles. Elle est depuis très utilisée en codage d'image [122, 159] et est utilisée par le codec JPEG-2000 [5, 139]. Une étude assez complète des propriétés théoriques des ondelettes biorthogonales 5/3 et 9/7 est présentée dans [153].

n	$h_0[n]$	$\widetilde{h}_0[n]$
0	0.85269867900940	0.78848561640566
1	0.37740285561265	0.41809227322221
2	-0.11062440441842	-0.04068941760956
3	-0.02384946501938	-0.06453888262894
4	0.03782845550699	

TAB. 1.3 – Coefficients des réponses impulsionnelles symétriques des filtres passe-bas d'analyse $h_0[n]$ et de synthèse $\widetilde{h}_0[n]$ associés à l'ondelette CDF 9/7.

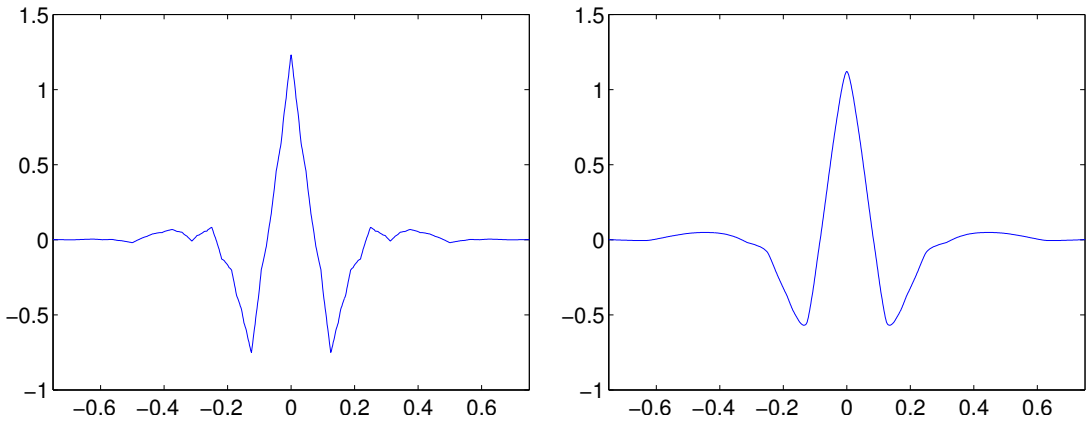


FIG. 1.9 – Ondelette CDF 9/7 d'analyse ψ et sa duale $\widetilde{\psi}$.

1.2.4 Compression d'image par transformée en ondelettes

Décomposition en ondelettes séparables

La transformation de signaux multidimensionnels peut être réalisé de façon séparable par transformations successives de l'image sur ses dimensions. Ainsi, la décomposition

en ondelettes des lignes puis des colonnes d'une image permet d'obtenir sa décomposition 2D sur un puis sur plusieurs niveaux, comme illustré en Fig. 1.10.

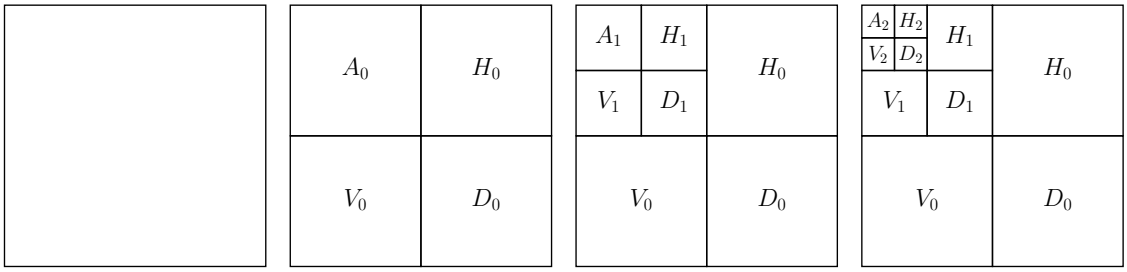


FIG. 1.10 – Décomposition successive d'une image en ondelettes sur trois niveaux. On remarquera la disposition pyramidale des sous-bandes d'approximation A_k de niveau k et des sous-bandes de détail horizontal H_k, vertical V_k et diagonal D_k.

L'image originale est tout d'abord décomposée en une sous-bande d'approximation A_0 et en sous-bandes de détails H_0, V_0 et D_0, correspondant respectivement aux détails horizontaux, verticaux et diagonaux. La décomposition successive des sous-bandes d'approximation A_k permet d'obtenir alors l'analyse multirésolution de l'image, qui se présente sous une forme pyramidale. La Fig. 1.11 illustre les sous-bandes obtenues par la décomposition en ondelettes 9/7 séparable de l'image *Lena* sur 3 niveaux (le contraste de l'image a été augmenté afin que les coefficients de détail soient visibles).

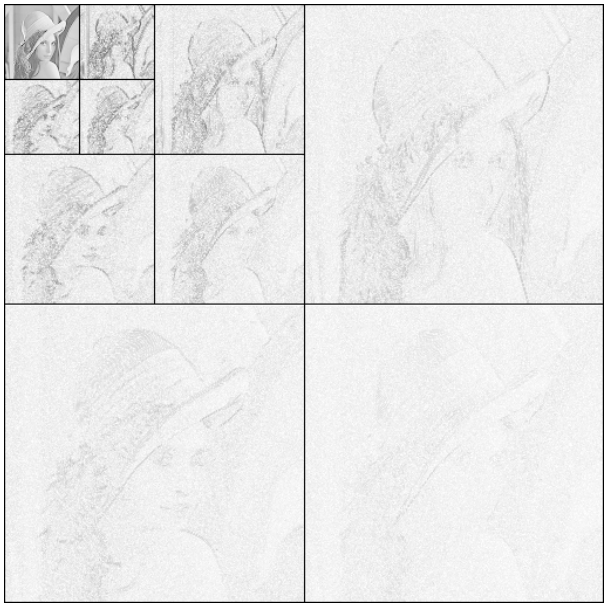


FIG. 1.11 – Décomposition en ondelettes séparables biorthogonales 9/7 de *Lena* sur 3 niveaux de résolution.

Nous détaillons dans les sous-sections suivantes les principaux algorithmes utilisés pour encoder efficacement et de façon scalable les coefficients issus de la décomposition en ondelettes d'une image.

EZW

L'algorithme de codage emboîté par ondelettes à arbres de zéros EZW (*Embedded Zerotree Wavelet*) a été proposé par Shapiro [130] en 1993. Il permet un codage efficace des coefficients d'ondelettes tout en assurant une scalabilité spatiale dyadique et une scalabilité fine en qualité. L'algorithme consiste en un codage progressif par plans de bits (*bitplanes*) des coefficients de la pyramide de décomposition spatiale, en tenant compte de leur dépendance hiérarchique entre niveaux de résolutions, comme illustré par la Fig. 1.12.

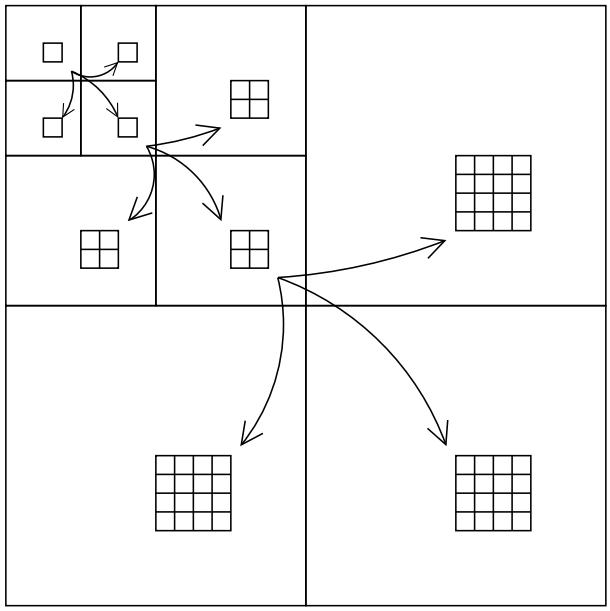


FIG. 1.12 – Prise en compte par le codec emboîté EZW de la dépendance hiérarchique spatiale des coefficients d'ondelettes entre niveaux de résolutions.

La pyramide 2D est encodée de façon progressive par plans de bits, par l'algorithme suivant. On considère à l'initialisation un seuil $T = 2^n$ où n est une valeur suffisamment grande pour que $T + T/2$ soit supérieur à l'amplitude maximale de tous les coefficients de la pyramide. Chaque itération i est composée de deux passes : une passe de description des coefficients significatifs et une passe de raffinement des valeurs des coefficients significatifs. La pyramide est parcourue sous-bande par sous-bande, en commençant par la sous-bande d'approximation et en progressant vers les hauts niveaux de résolution. Durant la première passe, chaque coefficient x de la pyramide est codé ou pas par un symbole et est rajouté au flux binaire. Tout d'abord, si $|x| > T$ alors le coefficient est déclaré comme significatif et est codé par le symbole P si il est positif et par le symbole N sinon. Si au contraire $|x| < T$, le coefficient n'est pas déclaré comme significatif et deux cas se présentent. Si x possède des descendants dans la pyramide dont la valeur absolue est supérieure à T alors il est codé par le symbole I. Sinon, il est codé par le symbole Z, comme *Zerotree root*, nœud racine d'un arbre de zéros, et aucun de ses descendants ne sera codé durant cette passe. Durant la passe de raffinement, les $(n-i)$-ième bits de poids forts de tous les coefficients significatifs sont encodés. Le seuil T est alors divisé par 2 et on relance une itération, en ne considérant durant la première passe que les coefficients déclarés comme non-significatifs. Le flux binaire ainsi formé constitue un fichier aisément décompressable, scalable en résolution spatiale et en qualité. La scalabilité en qualité est

obtenue par troncation des plans de bits de poids faible tandis que la scalabilité spatiale est assurée par la suppression des zones codant les sous-bandes de détail.

Le schéma de codage EZW est simple et très efficace. Il offre des courbes de débit-distorsion meilleures que le codec JPEG, produit des images de qualité visiblement supérieure et offre une scalabilité spatiale et en débit.

SPIHT

L'algorithme SPIHT (*Set Partitioning In Hierarchical Tree*), proposé par Said et Pearlman [122] est une amélioration du schéma de codage EZW. Il repose sur les mêmes concepts : codage progressif par plans de bits et utilisation des dépendances hiérarchiques qu'entretiennent les coefficients d'une pyramide de décomposition 2D. L'algorithme est cependant plus sophistiqué : contrairement à l'algorithme EZW qui n'utilise qu'un seul ensemble décrivant la signifiance des coefficients, SPIHT met en jeu une liste des ensembles insignifiants (LIS), une liste des coefficients insignifiants (LIP) et une liste des coefficients significatifs (LSP). Tout comme EZW, SPIHT utilise une passe de description des coefficients significatifs et une passe de raffinement. Enfin, du fait de sa meilleure modélisation de la signifiance des coefficients, l'algorithme SPIHT offre une meilleure efficacité de codage que EZW.

EZBC

Le codec EZBC [60] de Hsiang et Woods (*Embedded ZeroBlocks with Context modeling*) est basé sur le codec SPIHT en utilisant un codeur arithmétique contextuel, prenant en compte le voisinage du coefficient courant. L'algorithme EZBC est ainsi capable de prendre en compte les relations de dépendance statistique existant entre les coefficients d'une même sous-bande. Il offre une efficacité de codage supérieure au codec SPIHT et est à la base du schéma de codage vidéo MC-EZBC décrit dans le chapitre suivant.

EBCOT et la norme JPEG-2000

Le codec EBCOT (*Embedded Block Coding with Optimized Truncation*) est issu des travaux de Taubman [138] mais n'appartient pas à la famille des schémas de codage à arbre de zéros. Dans EBCOT, chaque sous-bande est tout d'abord découpée en petits blocs indépendants : les *codeblocks*. Ces derniers sont alors codés de façon progressive, par plans de bits et au moyen d'un codeur arithmétique contextuel. Enfin, une procédure d'optimisation débit-distorsion est utilisée pour déterminer le choix optimal des *codeblocks* à conserver pour assurer une qualité maximale pour un ensemble de débits donnés et connus à l'encodage : les points de troncatures. Les débits intermédiaires restent toutefois accessibles mais n'offriront peut-être pas une qualité de reconstruction optimale. Le codec EBCOT est très performant et offre une efficacité de codage comparable à EZBC. L'algorithme utilisé dans la norme JPEG-2000 [139] est largement basé sur ce codec.

Codeurs morphologiques - EMDC

Après avoir observé la tendance qu'ont les coefficients d'ondelettes à s'agglomérer près des contours d'une image, Servetto et Ramchandran [129] ont utilisé un opérateur morphologique de dilatation au sein d'un codeur d'images fixes, afin de modéliser ces agglutinations. Les auteurs obtiennent des résultats expérimentaux comparables avec le

codeur emboîté SPIHT et observent une amélioration visuelle notable des images décodées. Dans la continuation de ces travaux, Lazzaroni, Leonardi et Signoroni ont récemment proposé le schéma de codage d'image EMDC (*Embedded Morphological Dilation Coding*) [71], combinant les avantages de l'approche morphologique et des codeurs à arbre de zéros. Cet algorithme possède une efficacité de codage supérieure aux codeurs emboîtés et ouvre des perspectives intéressantes sur l'utilisation de modèles morphologiques pour mieux capturer l'information géométrique contenue dans les images.

1.3 Nouvelles représentations multirésolution

L'analyse multirésolution par ondelettes exposée dans la section précédente permet d'obtenir une représentation scalable d'un signal. Cependant, cette représentation est nécessairement linéaire et ceci constitue un inconvénient majeur dans la description de signaux réels comme les images naturelles où les discontinuités, contours et autres singularités sont nombreux. Afin de pallier à ce problème, de nouvelles décompositions multirésolution mieux adaptées à la représentation de tels signaux ont été introduites.

Nous présentons tout d'abord le schéma lifting de Sweldens. C'est une structure de décomposition multirésolution toujours inversible, capable de représenter n'importe quelle transformée en ondelettes dyadique basée sur des bancs de filtres à réponse impulsionnelle finie. De plus, elle autorise la construction de transformées multirésolution non-linéaires, de façon très naturelle. La structure lifting est à la base de nombreuses décompositions multirésolution non-linéaires dont on donnera quelques exemples.

De plus, bien que les ondelettes soient des outils adaptés à la description des discontinuités de signaux monodimensionnels, cette propriété n'est plus vraie pour des dimensions supérieures. Les ondelettes séparables sont en effet isotropes et ne peuvent pas capturer par exemple la régularité présente le long d'un contour d'une image. De nombreuses constructions adaptées aux images ont été proposées pour tenir compte de ce problème, nommées ondelettes géométriques, que nous décrivons dans la suite du document. Nous aborderons enfin d'autres représentations multirésolution récentes, palliant à certaines faiblesses inhérentes aux bases d'ondelettes.

1.3.1 Structure lifting

La formulation en banc de filtres et l'algorithme de transformée en ondelettes rapide, décrits dans la section 1.2.2, permettent une réalisation effective de la transformée en ondelettes discrète. Cependant, il n'est pas aisé de construire et de déterminer les filtres passe-bas h_0 et passe-haut h_1 mis en jeu dans la transformée à partir des équations de reconstruction parfaite (1.15) et (1.16).

C'est en cherchant à améliorer les propriétés d'un banc de filtres à reconstruction parfaite par l'ajout de nouveaux étages que Sweldens [136] a mis en évidence la structure dite en lifting. Cette structure possède le triple avantage de pouvoir représenter les transformées en ondelettes dyadiques, de toujours assurer une reconstruction parfaite et d'être suffisamment flexible pour construire de nouvelles transformées.

Définition

On appelle une structure en *lifting* dyadique, un banc dans lequel un signal d'entrée $x[k]$ est tout d'abord séparé en deux composantes : généralement, sa composante paire

$x[2k]$ et sa composante impaire $x[2k+1]$. Grâce à un nombre fini d'étages, on ajoute alors successivement sur une des composantes le résultat d'un opérateur appliqué sur l'autre composante. Ces opérateurs se nomment généralement prédicteur ou opérateur de *prédiction* et opérateur de *mise à jour*, selon le type d'opération qu'ils effectuent sur le signal. Les composantes finales sont alors multipliées par des constantes arbitraires.

Dans le cas particulier d'une structure lifting à deux étages, utilisant une étape de séparation en composantes polyphases ("lazy wavelets") du signal d'entrée, un opérateur de prédiction suivi d'un opérateur de mise à jour, on obtient le schéma d'analyse présenté en Fig. 1.13, en utilisant le signal $a_j[k] = x[k]$. L'ensemble des opérations effectuées peut alors s'écrire sous la forme suivante :

$$
\begin{aligned}
d^0_{j+1}[k] &= a_j[2k+1] + P\big(\{a_j[2k]\}_{k\in\mathbb{Z}}\big) \\
a^0_{j+1}[k] &= a_j[2k] + U\big(\{d^0_{j+1}[k]\}_{k\in\mathbb{Z}}\big) \\
d_{j+1}[k] &= \alpha\, d^0_{j+1}[k] \\
a_{j+1}[k] &= \beta\, a^0_{j+1}[k]
\end{aligned}
$$

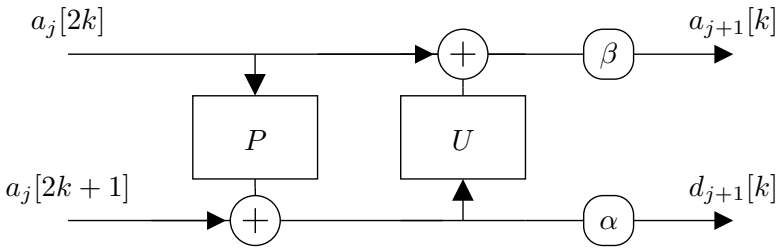


FIG. 1.13 – Structure d'analyse en lifting à deux étages.

Dans cette structure, le signal d'entrée a_j est tout d'abord séparé en deux composantes, nommées aussi sous-bandes : la composante paire $a_j[2k]$ et la composante impaire $a_j[2k+1]$. On ajoute alors à la sous-bande impaire le résultat de l'opérateur de prédiction P appliqué à la composante paire, donnant ainsi le signal $d^0_{j+1}[k]$. L'opérateur P est généralement choisi de façon à minimiser l'amplitude de la sous-bande de détail et se borne ainsi à prédire la sous-bande impaire à partir de la sous-bande paire.

Le signal $d^0_{j+1}[k]$, aussi nommé résidu de prédiction, est alors mis à jour au moyen de l'opérateur U et est ajouté à la sous-bande $a_j[2k]$ pour obtenir le signal $a^0_{j+1}[2k]$. Enfin, les signaux $a^0_{j+1}[k]$ et $d^0_{j+1}[k]$ sont multipliés par les réels β et α. On obtient alors respectivement la sous-bande d'approximation $a_{j+1}[k]$ et la sous-bande de détail $d_{j+1}[k]$.

Propriétés

Le lien entre la structure lifting et la transformée en ondelettes est simple. Daubechies a montré [43] que toute transformée en ondelettes dont les filtres sont à réponse impulsionnelle finie, peut être factorisée sous forme lifting avec un nombre fini d'étages de prédiction et de mise à jour. Ceci justifie ainsi l'utilisation des notations a_j et d_j introduites dans la section 1.2.1 pour désigner les coefficients de la transformée en ondelettes d'un signal. La transformée en ondelettes d'un signal peut donc être réalisée par la structure d'analyse en lifting de la Fig. 1.13. On remarquera ainsi son analogie avec le banc de filtres d'analyse de la Fig. 1.3, présenté dans la section 1.2.2.

Le théorème de factorisation de Daubechies est donc un outil puissant permettant de lier la formulation en banc de filtres à une formulation de type lifting. Ce théorème est de plus constructif et propose un algorithme permettant d'obtenir explicitement les opérateurs de prédiction P et de mise à jour U à partir des filtres h_0 et h_1 mis en jeu dans un banc de filtres d'analyse. Cet algorithme est basé sur la factorisation des transformées en Z de h_0 et h_1 par division euclidienne dans $\mathbb{R}[z, z^{-1}]$. À titre d'exemple, nous donnons en fin de section les structures lifting des transformées en ondelettes classiquement utilisées en codage d'images énoncées précédemment dans la section 1.3.1.

Cependant, l'intérêt principal de la structure en lifting réside dans la propriété suivante : quels que soient les opérateurs de prédiction et de mise à jour utilisés, la transformation par schéma lifting est inversible et on peut retrouver le signal original a_j à partir de ses composantes a_{j+1} et d_{j+1}. En effet, l'inversion du schéma d'analyse se réalise par un simple renversement des étapes et une inversion des signes, comme illustré par la Fig. 1.14 et les équations suivantes :

$$\begin{aligned} a^0_{j+1}[k] &= a_{j+1}[k]/\beta \\ d^0_{j+1}[k] &= d_{j+1}[k]/\alpha \\ a_j[2k] &= a^0_{j+1}[k] - U\big(\{d^0_{j+1}[k]\}_{k\in\mathbb{Z}}\big) \\ a_j[2k+1] &= d^0_{j+1}[k] - P\big(\{a_j[2k]\}_{k\in\mathbb{Z}}\big) \end{aligned}$$

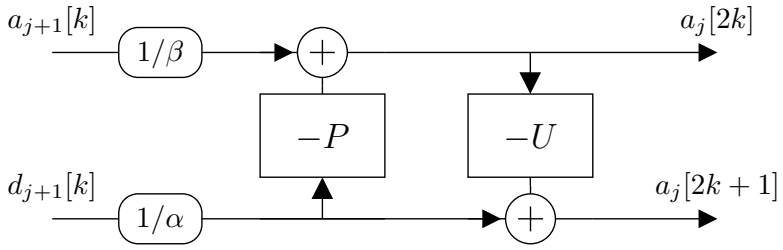


FIG. 1.14 – Structure de synthèse en lifting.

Il est important de remarquer que les opérateurs de prédiction P et de mise à jour U n'ont pas besoin d'être linéaires ni même inversibles pour que le schéma soit inversible. C'est une propriété importante qui permet la construction simple de transformations multirésolution inversibles, non nécessairement linéaires et qui ne sont donc pas représentables par des bases d'ondelettes classiques ou des structures en banc de filtres. Par exemple, le schéma de codage vidéo $t+2D$ abordé dans cette thèse utilise des images à la place d'échantillons et met en œuvre des opérateurs non-linéaires faisant appel à la compensation de mouvement et à d'autres traitements spatio-temporels.

La structure en lifting possède d'autres avantages par rapport à une structure en banc de filtres. La gestion des effets de bords lors de la transformation est par exemple facilitée. En effet, il suffit simplement de replier symétriquement les opérateurs de prédiction et de mise à jour lors du traitement des échantillons situés aux bords du signal. De plus, le schéma lifting permet de minimiser la complexité [43] de la mise en œuvre d'une transformation en ondelettes. On peut ainsi montrer que la structure lifting nécessite toujours moins de multiplications et d'additions qu'une structure en banc de filtres.

La construction explicite des opérateurs de prédiction P et de mise à jour U est plus problématique. Il est ainsi clair que dans un schéma à deux étages, le prédicteur P sera

choisi de façon à minimiser l'amplitude des coefficients de détail. Il est plus délicat de justifier le choix d'un opérateur de mise à jour car de nombreux critères rentrent en jeu. Un exemple de construction explicite d'une structure lifting adaptée à la compression d'images avec perte est donné dans [56] : en fonction des propriétés statistiques de l'image, l'opérateur de prédiction est choisi de façon à minimiser la variance des coefficients de détail et l'opérateur de mise à jour est conçu pour minimiser l'erreur de reconstruction. On remarquera cependant qu'il est difficile dans le cas général de donner une interprétation aux opérateurs P et U dans une structure lifting possédant un nombre d'étages supérieur à deux.

Généralisations

La structure en lifting peut se généraliser de nombreuses façons. Il est tout d'abord possible d'utiliser un nombre M de composantes polyphases supérieur à deux. Ceci permet la construction de représentations multirésolutions fournissant des facteurs de scalabilité d'un rapport M. De plus, le processus de séparation (*split*) d'un signal en composantes polyphases n'est pas nécessairement basé sur la parité. Ce processus peut être complètement arbitraire, pourvu qu'il soit inversible. Il est ainsi possible d'utiliser une séparation en quinconce 2-bandes non-séparable pour la décomposition d'images, minimisant ainsi le nombre de sous-bandes à traiter.

De plus, il est possible d'utiliser une autre fonction que l'addition $\oplus$ lors du mélange d'une sous-bande avec le résultat de l'opérateur appliqué sur l'autre sous-bande, pourvu que la fonction soit inversible. Les travaux de Piella [111, 115] et de Solé [131] envisagent ainsi à la place de l'addition des fonctions non-linéaires, mettant en œuvre un seuillage et des tests conditionnels. De même, les opérateurs finaux de multiplications peuvent être changés tant qu'ils restent inversibles. La structure reste dite en lifting si elle peut être inversée par renversement des étapes qui la composent.

Exemple de construction *ad-hoc*

Sweldens donne [136] un exemple de la construction *ad-hoc* d'une transformation en ondelettes sous forme lifting. Supposons un échantillon impair x_{2t+1}, il est raisonnable de prédire sa valeur par la moyenne de ses deux voisins pairs $(x_{2t} + x_{2t+2})/2$; on peut donc prendre comme opérateur de prédiction $P = -(x_{2t} + x_{2t+2})/2$. Les coefficients de détail h_t seront donc donnés par :

$$h_t = x_{2t+1} - (x_{2t} + x_{2t+2})/2 \tag{1.20}$$

L'opérateur de mise à jour est choisi de manière similaire et symétrique ; on souhaite ainsi qu'il soit de la forme $U = \xi(h_{t-1} + h_t)/2$. Si on impose de plus que la moyenne courante du signal d'entrée $\sum_t x_t$ soit égale à la moyenne du signal d'approximation $\sum_t l_t$, on obtient $\xi = 1/4$. Les coefficients d'approximation l_t seront donc donnés par :

$$l_t = x_{2t} + (h_{t-1} + h_t)/4 \tag{1.21}$$

La structure ainsi construite et décrite par les équations (1.20) et (1.21) est, aux coefficients multiplicatifs $\alpha = 1/\sqrt{2}$ et $\beta = \sqrt{2}$ près, la transformation biorthogonale en ondelettes 5/3 décrite dans la section 1.2.3.

Exemple de structure en lifting d'ondelettes dyadiques

Cette section énumère les structures en lifting des transformées en ondelettes classiquement utilisées en codage d'image et présentées dans le panorama de la section 1.2.3. La notation est simplifiée et on omet l'indice de résolution j en ne décrivant qu'un seul niveau de décomposition, les autres niveaux étant obtenus par des décompositions subséquentes. Le signal d'entrée a_j est alors noté x et les sous-bandes résultantes d'approximation a_{j+1} et de détail d_{j+1} sont notées respectivement l et h.

La structure lifting de la transformée de Haar est très simple : elle revient à calculer la moyenne et la différence des deux échantillons x_{2t} et x_{2t+1}. Elle s'exprime par :

$$h_t^0 = x_{2t+1} - x_{2t} \quad \text{(P)}$$
$$l_t^0 = x_{2t} + \frac{1}{2} h_t^0 \quad \text{(U)}$$
$$h_t = \frac{1}{\sqrt{2}} h_t^0 \quad \text{(S1)}$$
$$l_t = \sqrt{2} l_t^0 \quad \text{(S2)}$$

La structure en lifting de la décomposition en ondelettes Daubechies-4 est plus complexe et nécessite trois étages. Elle s'écrit :

$$h_t^0 = x_{2t+1} - \sqrt{3} x_{2t} \quad \text{(P1)}$$
$$l_t^0 = x_{2t} + \frac{\sqrt{3}}{4} h_t^0 + \frac{\sqrt{3}-2}{4} h_{t+1}^0 \quad \text{(U)}$$
$$h_t^1 = h_t^0 + l_{t-1}^0 \quad \text{(P2)}$$
$$h_t = \frac{\sqrt{3}-1}{\sqrt{2}} h_t^1 \quad \text{(S1)}$$
$$l_t = \frac{\sqrt{3}+1}{\sqrt{2}} l_t^0 \quad \text{(S2)}$$

La structure en lifting de la transformation en ondelette 5/3 correspond intuitivement à la prédiction d'un échantillon pair par la moyenne de ces voisins et au rehaussement des échantillons impairs afin de préserver la moyenne courante du signal. Plus précisément, elle s'exprime par les relations suivantes :

$$h_t^0 = x_{2t+1} - \frac{1}{2}(x_{2t} + x_{2t+2}) \quad \text{(P)}$$
$$l_t^0 = x_{2t} + \frac{1}{4}(h_{t-1}^0 + h_t^0) \quad \text{(U)}$$
$$h_t = \frac{1}{\sqrt{2}} h_t^0 \quad \text{(S1)}$$
$$l_t = \sqrt{2} l_t^0 \quad \text{(S2)}$$

Enfin, la structure lifting de la transformée biorthogonale 9/7 est composée de quatre étages : deux opérateurs de prédictions et deux opérateurs de mise à jour. On trouvera sa forme explicite dans [43].

Ondelettes entières

L'introduction d'un opérateur non-linéaire d'arrondi dans la structure lifting d'une transformée en ondelettes permet d'obtenir aisément une nouvelle transformée entière, c'est à dire à valeurs dans $\mathbb{Z}$. Cette décomposition entière est scalable, inversible et dispose de propriétés proches de la transformée originale. Ce type de construction est particulièrement utile en codage d'image sans perte car les coefficients issus de la transformation sont entiers et ne nécessitent pas de quantification. Les travaux de Calderbanks [28] ont consisté en l'étude de transformées entières donnant la meilleure efficacité de codage en compression d'image sans perte. Ses résultats ont conclu sur la supériorité de la transformée 5/3 entière, reprise ensuite par la norme JPEG-2000 et décrite simplement par les relations :

$$h_t = x_{2t+1} - \lfloor (x_{2t} + x_{2t+2})/2 + 1/2 \rfloor \quad \text{(P)}$$

$$l_t = x_{2t} + \lfloor (h_{t-1} + h_t)/4 + 1/2 \rfloor \quad \text{(U)}$$

où $\lfloor \cdot \rfloor$ désigne l'opérateur d'arrondi à l'entier inférieur.

1.3.2 Ondelettes géométriques non-adaptatives

Les bases d'ondelettes séparables utilisées pour décomposer les images possèdent un support carré indéformable et sont isotropes. Pour cette raison, elles ne peuvent pas représenter de manière optimale des régions comportant des contours ou des singularités locales. De nombreuses transformées en ondelettes anisotropes, aptes à la représentation optimale d'images comportant des contours, ont été proposées afin de pallier à cet inconvénient. Nous nous proposons dans cette section de décrire quelques ondelettes géométriques non-adaptatives, qui ont la particularité de posséder une base fixe et indépendante de l'image qu'elles représentent.

Ridgelets

Les ondelettes sont efficaces pour représenter des fonctions continues par morceaux et pour capturer des singularités *ponctuelles* dans le cas monodimensionnel. Ceci n'est pas vrai dans le cas 2D, où l'utilisation de bases d'ondelettes séparables ne permet pas de saisir la régularité présente le *long* des contours. La représentation par Ridgelets a été proposée par Candès et Donoho [31] pour apporter une solution à ce problème et permet de capturer efficacement la régularité présente le long de contours rectilignes. Elle repose sur l'utilisation de la transformée de Radon qui permet de représenter une image $f \in L^2(\mathbb{R}^2)$ de façon bijective dans le domaine polaire :

$$R_f(\theta, t) = \int_{\mathbb{R}^2} f(x, y) \delta\big(x\cos(\theta) + y\sin(\theta) - t\big) dxdy$$

Les ridges $R_f(\theta, t)$ représentent le résultat de la projection radiale de f sur la droite d'équation $x\cos(\theta) + y\sin(\theta) = t$. La transformée de Radon est donc capable de transformer les singularités rectilignes présentes dans une image en singularités ponctuelles. La transformée en Ridgelets s'obtient alors en appliquant une transformée en ondelettes 1D sur le long des ridges $R_f(\theta, \cdot)$ en utilisant la variable d'intégration t. La décomposition d'une image $f \in L^2(\mathbb{R}^2)$ en Ridgelets s'écrit alors :

$$RT_f(a, b, \theta) = \frac{1}{\sqrt{a}} \int_{\mathbb{R}} \psi\Big(\frac{t-b}{a}\Big) R_f(\theta, t) dt$$

où a est un facteur d'échelle, b un paramètre de translation, θ l'angle de projection et ψ une ondelette. Elle peut s'écrire aussi :

$$RT_f(a,b,\theta) = \int_{\mathbb{R}^2} \psi_{a,b,\theta}(x,y) f(x,y) dxdy$$
$$\text{avec } \psi_{a,b,\theta}(x,y) = \frac{1}{\sqrt{a}} \psi\Big[\frac{x\cos(\theta) + y\sin(\theta) - b}{a}\Big]$$

et correspond ainsi à la projection de l'image f sur la base fixe $\{\psi_{a,b,\theta}\}_{a,b,\theta}$.

La transformée en Ridgelets est cependant définie dans le domaine continu et n'est pas directement utilisable en compression d'images. Cependant, on peut facilement la discrétiser et en déduire une trame : cela conduit à une description redondante, permettant d'assurer la reconstruction parfaite. On notera enfin les travaux récents de Do et Vetterli [46] proposant une transformée en Ridgelets discrète, inversible et non-redondante. Bien qu'intéressante, la transformée en Ridgelets n'est cependant adaptée qu'aux images présentant des discontinuités le long de contours rectilignes : elle n'est donc pas optimale pour la représentation d'images naturelles, comportant de nombreuses singularités ponctuelles, rectilignes et curvilignes.

Curvelets

La transformée en Ridgelets n'est adaptée qu'aux images présentant des discontinuités le long de contours rectilignes et son intérêt est limité dans le cas d'images naturelles. Cependant, il est clair qu'une image possède *localement* des contours rectilignes : c'est l'idée de la transformation en Curvelets introduite par Candès et Donoho [30, 48]. Elle se décrit en deux étapes : le support de l'image est tout d'abord partitionné en carrés de taille variable avec recouvrement, pour éviter les effets de bords, et ces carrés sont alors décomposés par une analyse en Ridgelets discrète.

Durant cette transformée, les contours non capturés par l'analyse en ondelettes séparables se retrouvent dans les sous-bandes de détail. Un partitionnement suffisamment fin des sous-bandes permet alors d'obtenir des blocs où ces contours forment des lignes droites et sont donc adaptés à l'analyse en Ridgelets. La transformée en Curvelets est inversible mais redondante car l'analyse en Ridgelets discrète sous-jacente est réalisée au moyen d'une FFT du plan polaire, nécessitant plus de points que ceux disponibles dans la grille rectangulaire. Starck [134] montre que son utilisation donne de bons résultats en débruitage d'images.

Contourlets

Les Contourlets sont des ondelettes géométriques non-adaptatives issues des résultats des travaux de Do et Vetterli [47]. Les bases de Contourlets possèdent un grand nombre d'orientations différentes et sont aptes à saisir la régularité présente le long des contours d'une image. De plus, contrairement aux Curvelets, elles sont définies directement dans le domaine discret et leur mise en œuvre en est grandement facilitée. La transformée en Contourlets s'effectue au moyen d'une pyramide Laplacienne redondante et d'un banc de filtres directionnels, construit au travers d'un arbre de décomposition binaire de l étages. Ce dernier utilise des filtres en éventail pour séparer l'information horizontale et verticale et des opérateurs de cisaillement (*shearing*) pour varier les orientations obtenues. La combinaison de ces outils permet de construire une base de Contourlets offrant 2^l

orientations possibles. La transformation en Contourlets est inversible mais la pyramide Laplacienne sous-jacente crée cependant une petite redondance (allant jusqu'à 33 %) qui réduit l'intérêt de son utilisation en compression d'image fixe.

Des travaux plus récents sur les CRISP-contourlets [76] permettent de s'affranchir de ce problème en proposant une transformée en Contourlets inversible et non-redondante mais aucun résultat expérimental de compression n'est présenté. Une méthode hybride combinant une décomposition en Contourlets à l'échelle fine suivie d'une décomposition en ondelettes pour les résolutions grossières a été proposée par Chapellier [33] et offre une efficacité de codage supérieure à la transformée 9/7 séparable dans les bas débits.

1.3.3 Ondelettes géométriques adaptatives

Plutôt que d'utiliser une base fixe, de nombreuses constructions font appel à une base dont les fonctions sont choisies pour s'adapter au mieux à une image donnée. On parle alors d'ondelettes géométriques adaptatives. Le point commun de ces décompositions réside dans une étape d'estimation préalable de la géométrie de l'image (par triangulation, détection de contours, estimation de régularité...) avant de procéder à la décomposition.

Bandelettes

Le flux géométrique d'une image est défini comme un champ indiquant les directions où les variations d'intensités sont régulières. Les bases de Bandelettes, dues à Le Pennec et Mallat [72] sont obtenues par déformation d'une base d'ondelettes selon le flux géométrique local. Les Bandelettes sont cependant construites dans le domaine continu et leur mise en œuvre dans le cas discret est complexe. De plus, le modèle de flux géométrique retenu n'est pas scalable : bien qu'elle offre des résultats expérimentaux en compression d'image convaincants, la transformée en Bandelettes ne semble pas optimale.

Une nouvelle génération de Bandelettes, dues à Peyré [110], sont directement construites dans le domaine discret. La transformation en Bandelettes discrète est définie par l'algorithme suivant. On réalise d'abord une décomposition 2D en ondelettes séparables. Les sous-bandes résultantes sont alors partitionnées en sous-blocs, de façon à isoler les contours orientés. Les sous-blocs sont à leur tour décomposés par une transformée de Haar orientée dans la direction d'angle $k\pi$, où k est un nombre rationnel choisi de façon à minimiser l'amplitude des coefficients résultants. Les paramètres k de chaque sous-bloc décrivent en fait la géométrie de l'image et sont codés à part. Les résultats expérimentaux observés en compression d'image lors de l'utilisation de la transformée en Bandelettes discrète sont satisfaisants.

Représentation adaptée aux contours

Un autre schéma de décomposition non-linéaire et adapté aux contours a été proposé par Cohen et Matei [42, 87]. Il consiste en la construction d'un opérateur de prédiction non-linéaire, capable d'agrandir une image d'un facteur deux. L'utilisation de cet opérateur au sein d'une pyramide Laplacienne permet alors d'obtenir une décomposition multirésolution d'une image. La prédiction est effectuée au moyen d'un ensemble limité de *stencils*, illustrés en Fig. 1.15, qui sont des supports de prédiction de formes variées. Ces *stencils* sont ainsi en mesure de prédire les quatre petits pixels à partir du support de prédiction. Le choix des *stencils* de prédiction est fait à chaque pixel lors d'une étape

préalable de détection de contours et sont codés dans une carte de décision. Cette approche donne de bons résultats sur les images synthétiques et sur les images naturelles à contours nets. Cependant, l'efficacité de cette représentation reste très tributaire de la pertinence et du coût de la carte de décisions.

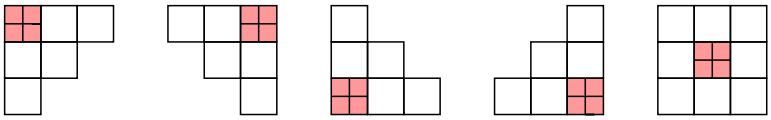


FIG. 1.15 – Stencils : supports de prédiction utilisés dans la représentation de Cohen et Matei. Les pixels de support sont en clair tandis que ceux qui seront prédits sont grisés.

Ondelettes orientées

Afin de représenter au mieux les contours, Chappelier [32] a proposé un schéma de décomposition adaptatif original mettant en œuvre des ondelettes géométriques orientées. Après une décomposition polyphase de l'image en quinconce, les pixels sont prédits soit par rapport à leurs voisins horizontaux, soit par rapport à leurs voisins verticaux, comme illustré sur la Fig. 1.16. La mise à jour est alors faite en fonction de l'état de connexité du pixel. Une carte de décision externe est utilisée pour mémoriser les décisions binaires prises à chaque pixel lors de la prédiction et une procédure d'optimisation débit-distorsion est utilisée pour minimiser son coût. Les résultats expérimentaux obtenus sont bons sur des images présentant de forts contours horizontaux et verticaux comme *Barbara* et sont satisfaisants sur d'autres images naturelles.

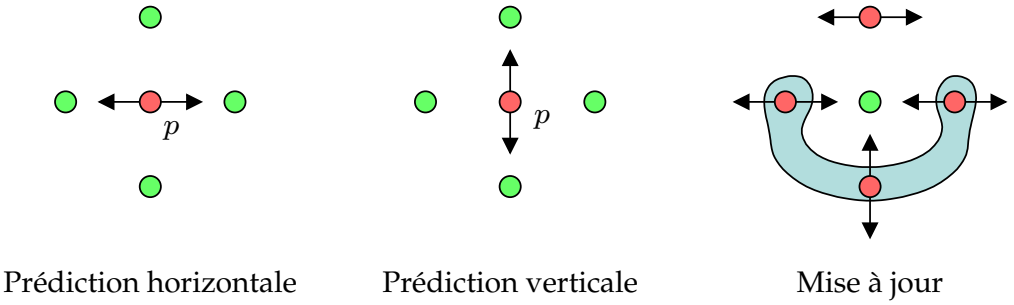


FIG. 1.16 – Étapes de prédiction adaptative et de mise à jour utilisées dans la décomposition en ondelettes orientées de Chappelier.

1.3.4 Autres représentations

L'analyse multirésolution par ondelettes possède certaines faiblesses inhérentes aux bases d'ondelettes même : la transformée en ondelettes n'est ainsi pas invariante par translation et a tendance à créer des coefficients de détail interdépendants entre eux sur plusieurs niveaux de résolution. Nous décrivons ci-après quelques représentations permettant de pallier à ces inconvénients.

Représentation en arbre dual

La transformation en ondelettes classique n'est pas invariante par translation. Cette caractéristique est gênante dans plusieurs applications, par exemple lors de la détection de mouvements dans le domaine transformé, comme décrit en section 2.2.5. La représentation en arbre dual, décrite par Kingsbury [67] permet de s'affranchir de ce problème en proposant une transformée redondante approximativement invariante par translation.

Elle consiste d'une part en la décomposition classique d'un signal en ondelettes et d'autre part, en la décomposition de ce même signal avec les mêmes filtres mais déphasés d'un échantillon. Dans le cas monodimensionnel, cette représentation est redondante d'un facteur 2 et peut être assimilée à une transformation en ondelettes *complexes* où la partie réelle est fournie par la décomposition en ondelettes et la partie imaginaire est donnée par les coefficients issus de la décomposition déphasée.

Footprints

Les discontinuités représentent souvent une partie importante de l'information véhiculée par un signal, en particulier dans le cas des images. Lors d'une décomposition en ondelettes, elles engendrent cependant la création de coefficients de large amplitude présents à tous les niveaux des différentes sous-bandes de détail. Comme illustré sur la Fig. 1.11, ces coefficients entretiennent une interdépendance mutuelle et cette caractéristique n'est pas souhaitable en compression de signal.

Les Footprints, introduites par Dragotti et Vetterli [49], sont des bases redondantes de fonctions capables d'engendrer n'importe quel signal fini polynomial par morceaux, de taille N et de degré maximal D. Elles sont obtenues par la décomposition en ondelettes d'un ensemble de fonctions polynomiales élémentaires de degré $d = 0, 1, \ldots D$ et présentant une discontinuité à l'indice $k = 0, 1, \ldots N-1$, formant ainsi une base redondante de $N \times (D+1)$ fonctions. La décomposition en Footprints est alors réalisée par un algorithme spécifique rapide de poursuite adaptative dans cette base. Les résultats obtenus sur des applications de débruitage et de compression de signaux sont très satisfaisants.

1.4 Conclusion

La transformation en ondelettes est un outil capable de donner une représentation multirésolution et parcimonieuse d'un signal monodimensionnel. Dans le cas de signaux multidimensionnels comme les images, il est possible de construire des bases d'ondelettes séparables par produit tensoriel. Cependant, ces dernières possèdent un support carré indéformable et sont isotropes : pour ces raisons, elles ne peuvent pas représenter de manière optimale les régions d'une image comportant des contours ou des singularités locales. Afin de pallier à cet inconvénient, de nombreuses bases d'ondelettes anisotropes (Curvelets, Contourlets, Bandelettes, Ondelettes orientées...) ont alors été proposées pour permettre une représentation plus économique des images.

En parallèle de ces travaux, la découverte de la structure lifting a permis de construire simplement des transformées multirésolution, toujours inversibles et autorisant la mise en œuvre d'opérateurs non-linéaires capables de saisir les singularités d'un signal. De plus, la structure lifting est aisément extensible au cas multidimensionnel et constitue un cadre idéal pour concevoir des transformées scalables capables de fournir une représentation économique d'une séquence vidéo.

Chapitre 2

Codage vidéo scalable : un état de l'art

Les travaux menés tout au long de cette thèse ont pour but de construire un schéma de décomposition permettant la description *scalable* et *parcimonieuse* d'une séquence vidéo. Avant toutes choses, il est cependant nécessaire de dresser un inventaire des schémas de codage vidéo scalable existants.

La majeure partie des codecs vidéos actuels, dont les célèbres MPEG-2 et DivX, sont des schémas de codage dits de type hybride. Capable d'offrir une scalabilité grossière en couches, ce type de schéma constitue le socle de nombreux autres codecs et nous détaillons son principe dans la section 2.1. Nous dressons ensuite un inventaire rapide des principaux codecs normalisés par les organismes MPEG et ITU et décrivons alors en détails les extensions MPEG-4 FGS et SVC, construites sur la base de codecs hybrides et permettant d'étendre leurs propriétés de scalabilité.

Les travaux sur les schémas de codage vidéo par ondelettes sont plus récents. Ces derniers sont intrinsèquement scalables et nous décrivons dans la section 2.2 la structure de codage la plus prometteuse : le schéma de codage $t + 2D$, basé sur l'utilisation d'un filtrage temporel compensé en mouvement. Nous détaillons alors les avancées majeures réalisées sur ce schéma, dont l'introduction du lifting temporel, et décrivons les nombreuses améliorations et variantes récemment publiées sur cette structure. Nous nous attarderons plus particulièrement sur la description détaillée du codec MC-EZBC qui est à la base du prototype utilisé pour valider nos travaux de recherche.

2.1 Codage vidéo hybride scalable

Cette section rappelle les principes de base des schémas de codage vidéo hybride, dont sont issus les codecs de la famille MPEG. Ils sont dit hybrides car ils mettent généralement en jeu une prédiction temporelle des blocs d'une image par rapport à une autre image suivie d'une transformation spatiale de type DCT des résidus de prédiction. Cette structure de codage n'est cependant pas scalable et nous décrivons dans la suite les principales extensions apportées au schéma pour y remédier.

2.1.1 Schéma de principe d'un codeur vidéo hybride

Le schéma de principe d'un encodeur vidéo hybride est donné en Fig. 2.1. C'est une structure d'encodage en boucle fermée : un décodeur est intégré à l'encodeur et fournit les images reconstruites qui serviront à prédire l'image courante, constituant ainsi une boucle de rétroaction. Les images d'entrées x_t provenant d'une séquence vidéo sont lues et sont transformées par les étapes suivantes.

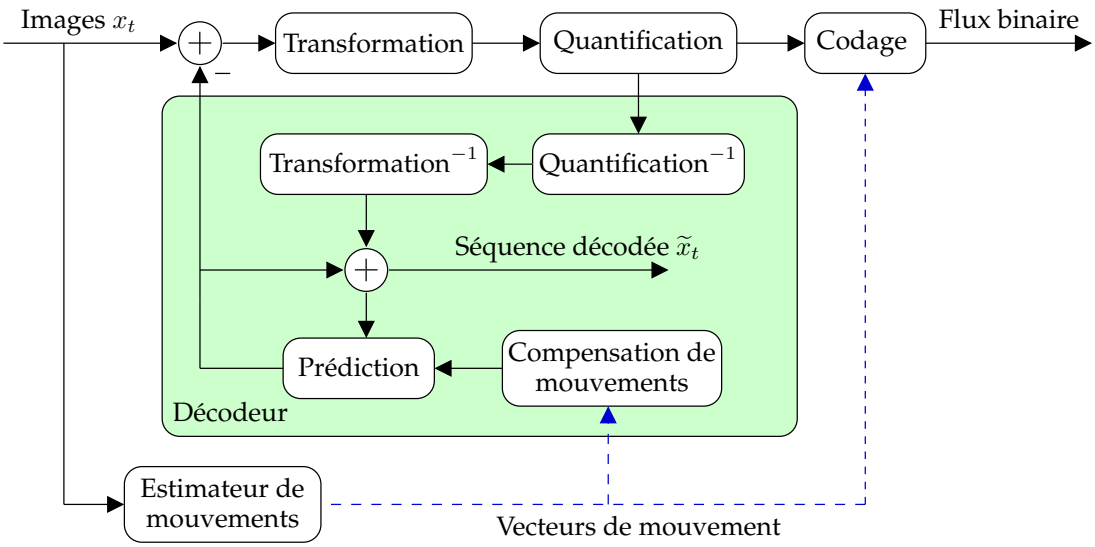


FIG. 2.1 – Schéma de principe d'un encodeur vidéo hybride avec boucle de rétroaction.

Estimation de mouvement Avant transformation des images d'entrée, on procède à une estimation de mouvement. Ce dernier est généralement représenté par des champs de blocs de taille fixe ou variable, dont la précision peut être subpixellique. La connaissance du mouvement permet alors une réduction efficace de la redondance temporelle présente entre les images d'une séquence vidéo.

Prédiction et soustraction de l'image prédite Le principe essentiel du schéma de codage hybride réside dans la propriété suivante : les images courantes sont prédites par rapport à des images reconstruites précédemment. Cette stratégie permet de simuler le comportement du décodeur afin d'éviter une quelconque dérive lors de la reconstruction de la séquence mais implique la présence d'un décodeur intégré dans l'encodeur. L'image prédite est alors soustraite à l'image courante et conduit à une image résultante nommée résidu de prédiction ou DFD (*Displaced Frame Difference*). Il existe trois modes classiques de prédiction des images. Les images dites *Intra* (I) ne sont pas prédites : elles sont assez volumineuses mais sont indépendantes des autres images. Les images dites *Inter* de type (P) sont prédites par rapport à une image précédente et sont plus simples. Enfin, les images dites *Inter* de type (B) sont prédites bidirectionnellement par rapport à une image passée et une image future, et sont encore plus concises. Les images d'une séquence vidéo sont généralement encodées par un motif de prédiction cyclique fixe, illustré en Fig. 2.2.

Transformation spatiale et quantification Les images résiduelles de prédiction sont transformées spatialement pour exploiter leur redondance spatiale. La transformée utilisée est généralement une transformée en blocs de type DCT 8×8, utilisée dans la norme JPEG et dont les propriétés sont rappelées dans la section 5.1.2. Les coefficients résultants sont alors quantifiés par des tables, sous le contrôle d'un paramètre de qualité Q.

Codage entropique Après quantification, les coefficients des images sont encodés par un parcours en zig-zag, un codeur de type RLE (*Run-Length Encoding*) et un codeur entropique de Huffman. Les champs de mouvement sont quant à eux encodés sans perte

au moyen de codes de longueur variable (VLC) (*Variable Length Coding*).

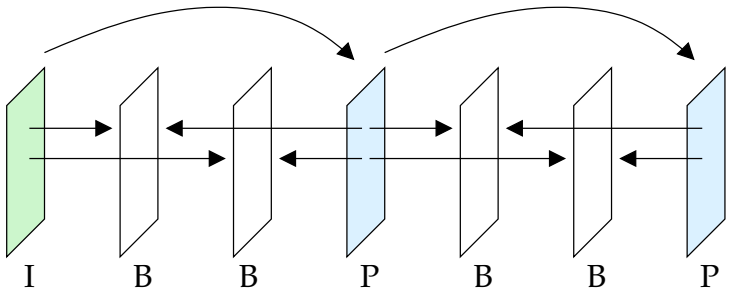


FIG. 2.2 – Agencement des modes de prédiction IBBPBBP d'un groupe d'images.

Les schémas de codage vidéo hybride permettent de compresser efficacement une séquence vidéo mais ne sont pas en mesure de fournir directement une représentation scalable. Les codecs MPEG-2 et MPEG-4 Part. 2 disposent cependant d'une structure prédictive en couches, capable d'offrir une forme de scalabilité grossière, où chaque couche représente une version de la séquence vidéo à une résolution spatio-temporelle et un débit donné. En l'absence de cette structure en couches, il n'est pas possible de modifier le débit, la résolution spatiale ou la fréquence temporelle d'une séquence vidéo compressée sans procéder à un transcodage. Cette opération nécessite un décodage et un réencodage complet de la séquence vidéo et est généralement très coûteuse en temps de calcul. De nombreuses stratégies ont cependant été mises au point [10, 96] pour diminuer la complexité de l'étape de transcodage.

2.1.2 Panorama des codecs MPEG et H.26X

MPEG-1 et MPEG-2

Le codec MPEG-1 [1] utilise le schéma de principe décrit en Fig. 2.1. L'estimation de mouvement est réalisée sur des macroblocs de taille fixe de 16×16 pixels, avec une précision pouvant aller jusqu'au demi-pixel. Le codec MPEG-2 [2] est une extension de MPEG-1 et permet la gestion des images entrelacées, couramment utilisées en télévision numérique. Il possède une efficacité de codage honorable et constitue la base des normes de diffusion de télévision numérique DVB et ATSC.

MPEG-4 Part. 2

La norme MPEG-4 [3] définit deux algorithmes de codage vidéo. Le premier est nommé MPEG-4 Partie 2 et est basé sur le codec MPEG-2. Il met en jeu un modèle de mouvement plus sophistiqué que ce dernier, lui permettant d'utiliser 4 vecteurs de mouvement par macrobloc, de gérer la compensation globale de mouvement et autorisant une précision pouvant aller jusqu'au quart de pixel. De plus, le codec MPEG-4 utilise un mécanisme de prédiction spatiale des macroblocs de type *Intra* par rapport à leur voisins, pour diminuer leur coût de codage. De façon similaire, il met aussi en œuvre une stratégie de prédiction médiane des champs de mouvement. Enfin, on remarquera que le célèbre codec DivX n'est autre qu'une variante du codec MPEG-4 Partie 2.

H.264/AVC MPEG-4 Part. 10

Le schéma de codage H.264 [140] a été développé par l'ITU dans la continuation des travaux sur le codec H.263. Afin d'éviter la multiplication de normes non-interopérables entres elles et dans le but de fournir une norme de codage vidéo efficace et unifiée, l'organisme de normalisation MPEG a décidé de reprendre les spécifications du codec H.264 et de les intégrer dans une nouvelle partie de la norme MPEG-4 : la norme MPEG-4 Partie 10, rebaptisée AVC (*Advanced Video Coding*).

Le codec H.264 est basé sur le schéma de principe d'un codeur vidéo hybride mais se différencie de ses prédécesseurs sur plusieurs points. Tout d'abord, la prédiction temporelle s'effectue sur les subdivisions des macroblocs, qui peuvent prendre les tailles suivantes : 16×16, 8×16, 8×8, 8×4, ... , 4×4. De plus, chaque bloc est prédit selon un *mode* de prédiction qui peut être de type *Intra*, monodirectionnel ou bidirectionnel. Il existe deux autres modes de prédiction bidirectionnelle *directs*, ne nécessitant pas le codage de vecteurs mouvement. Le choix de la subdivision d'un bloc et de son mode de prédiction est réalisé par la minimisation Lagrangienne d'un critère de coût $D + \lambda R$, où D représente la distorsion créée par la prédiction et R le coût de la description du mode, de la subdivision et du vecteur mouvement. Lors de l'étape de prédiction, le codec H.264 maintient un tampon de plusieurs images, pouvant être réutilisées lors de la prédiction des blocs. De plus, le codec utilise une transformée spatiale de type DCT entière, décrite dans la section 5.1.2, permettant d'éviter les dérives observées à la reconstruction lors de l'utilisation de la DCT classique. Un algorithme de correction des artefacts de type bloc (*deblocking*) est mis en jeu dans la boucle de décodage et apporte un gain substantiel de l'efficacité de codage comparé au codec MPEG-4 Part. 2. Enfin, le codage entropique peut être réalisé par le codeur contextuel à codes de longueur variable CAVLC (*Context Adaptive Variable Length Coding*) ou par le codeur arithmétique contextuel CABAC (*Context Adaptive Binary Arithmetic Coding*), permettant une meilleure modélisation des coefficients quantifiés et un codage plus efficace. Malgré sa complexité accrue, le codec H.264 est un schéma de codage vidéo performant qui surpasse tous les schémas étudiés précédemment.

2.1.3 Scalabilité et l'extension MPEG-4 FGS

Le schéma de codage MPEG-4 Partie 2 n'offre pas une scalabilité en qualité *fine* : il est ainsi nécessaire d'utiliser une structure en couches pour représenter une séquence vidéo sur plusieurs débits. Une extension de la norme MPEG a cependant été proposée pour lui adjoindre la propriété de scalabilité en qualité fine : l'extension FGS (*Fine Grain Scalability*) [4]. Cette extension consiste en l'utilisation de deux couches : une couche de base (*Base layer*), représentant la séquence vidéo dans une qualité grossière et une couche de raffinement, qui contient le résidu de la différence de la séquence vidéo avec la couche de base. La couche de base est compatible au format MPEG-4 Part. 2 et peut être décodée indépendamment de la couche de raffinement, dans laquelle les coefficients de texture sont codés par plans de bits pour permettre une scalabilité fine en qualité.

L'extension FGS est compatible avec la structure prédictive en couches classique du codec MPEG-4 Part. 2 et le codec MPEG-4 FGS dispose donc d'une scalabilité spatio-temporelle grossière et d'une scalabilité en débit fine. Cependant, son efficacité de codage n'est pas satisfaisante [34, 161] : on observe des pertes de qualité pouvant aller jusqu'à 3 dB par rapport au codec MPEG-4 Part. 2. En dépit de sa scalabilité, cette chute de performance apparaît trop élevée et la norme MPEG-4 FGS n'a jamais été vraiment déployée.

2.1.4 SVC ou l'extension scalable de H.264

Conscient de la nécessité d'un schéma de codage vidéo scalable et efficace pour faciliter l'adaptation de contenu et le codage robuste, l'organisme de normalisation MPEG s'est joint à l'ITU pour lancer un appel à propositions [6] en Décembre 2003 sur la création d'une nouvelle norme de codage vidéo scalable : la norme SVC (*Scalable Video Coding*). Les travaux de Schwarz et Wiegand [7, 123] portant sur un schéma prédictif en couches basé sur le codec H.264 ont alors montré la meilleure efficacité de codage subjective. Sur la base de ce schéma, le consortium MPEG a démarré la normalisation du futur schéma SVC [141], dont la finalisation est prévue pour début 2007.

Tout comme l'extension MPEG-4 FGS, le codec SVC est un schéma prédictif en couches et est illustré par la Fig. 2.3. Il met en jeu une couche de base (*Base Layer*) compatible avec le codec H.264, représentant la séquence vidéo dans sa résolution spatio-temporelle la plus faible. Les couches supplémentaires, dites de raffinement, représentent la séquence vidéo sur des résolutions spatio-temporelles plus élevées.

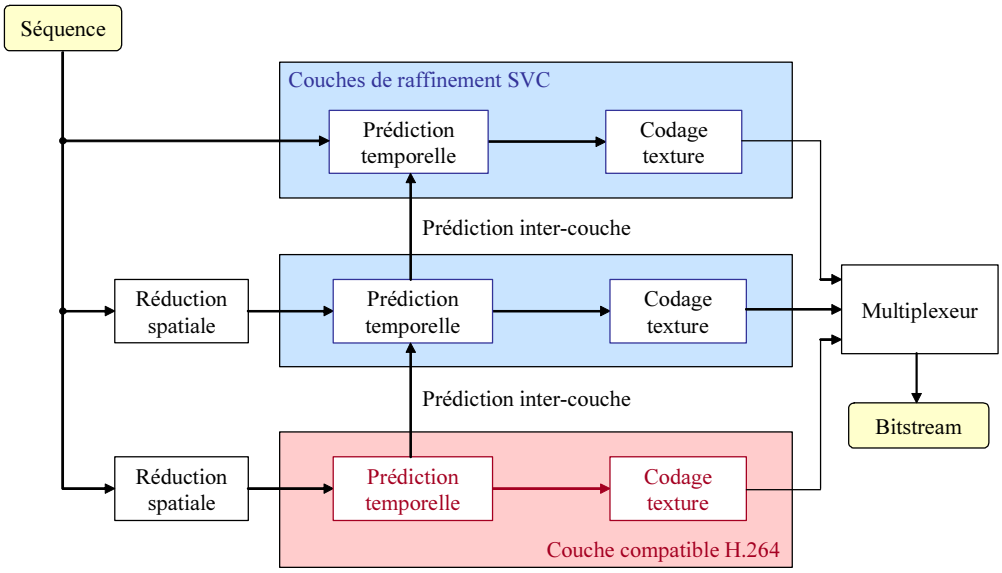


FIG. 2.3 – Structure de l'encodeur du schéma de codage SVC.

Les couches de raffinement sont obtenues par un schéma de codage similaire à H.264. Cependant, contrairement à ce dernier, les coefficients des textures sont encodés d'abord avec un pas de quantification grossier, puis l'erreur de quantification résultante est re-quantifiée avec un pas plus fin, et ainsi de suite de façon progressive, autorisant ainsi une scalabilité "moyenne" en qualité. De plus, il existe un mécanisme de prédiction entre les différentes couches, permettant la réutilisation des textures et des champs de mouvement provenant des couches précédentes. Ce mécanisme consiste en l'ajout d'un nouveau mode de prédiction, permettant d'utiliser un bloc provenant de la couche précédente. La prédiction des blocs est faite de façon bidirectionnelle, en utilisant les blocs des images voisines de l'image courante. Enfin, il n'y pas de restriction dyadique sur l'opérateur de réduction spatiale, permettant au schéma SVC d'offrir une scalabilité spatiale quelconque, directement liée aux résolutions spatiales des différentes couches de raffinement.

Le schéma de codage vidéo SVC offre une très bonne efficacité de codage, presque équivalente au codec H.264, et possède une couche de base compatible avec ce dernier. Il offre une scalabilité en qualité "moyenne", une scalabilité spatiale quelconque et une scalabilité temporelle dyadique. Sa mise en œuvre n'est cependant pas évidente car il nécessite plusieurs paramètres qui sont difficiles à estimer. Les contraintes λ_j (une pour chaque niveau temporel), utilisées lors de la prédiction temporelle pour déterminer le mode optimal de prédiction d'un bloc par minimisation Lagrangienne en sont un exemple : elles sont dépendantes de la séquence et influent beaucoup sur l'efficacité globale de codage.

2.2 Codage vidéo scalable par ondelettes

Après quelques travaux pionniers sur l'extension séparable de la transformée en ondelettes 2D au cas 3D, tout n'a vraiment commencé qu'avec le schéma de codage $t+2D$ permettant la prise en compte du mouvement dans la décorrélation des images d'une séquence vidéo. De nombreuses extensions ont alors été proposées et le schéma de codage MC-EZBC, sur lequel est basé notre prototype, a été rendu public.

2.2.1 Premières approches

Karlsson et Vetterli [64] ont utilisé en 1988 une transformée en ondelettes étendue au cas séparable 3D afin de compresser une séquence vidéo. En appliquant la transformée de Haar dans les trois directions T, X, Y, les auteurs ont obtenu un schéma de codage vidéo entièrement scalable et d'efficacité respectable. Cependant, la présence d'un mouvement trop important dans une séquence rend inefficace la décorrélation opérée par la transformée dans la direction temporelle et conduit à l'apparition de zones floues dans les images décodées. L'utilisation de la transformée biorthogonale $9/7$ et de codeurs emboîtés 3D [63, 66] basés sur les codecs SPIHT et EBCOT améliore légèrement l'efficacité de ce type de schéma de codage, sans toutefois atteindre celle des schémas de codage hybride pour des séquences présentant un mouvement. Il faut attendre l'introduction du schéma $t+2D$ et la prise en compte du mouvement au sein du filtre temporel, réalisée par Ohm en 1994, pour obtenir des performances honorables.

2.2.2 Schéma de codage vidéo t+2D

En 1994, Taubman et Zakhor [137] ont proposé un schéma de codage vidéo par ondelettes où une étape préalable d'alignement des images permettait de prendre en compte un éventuel mouvement global de translation. Ce type de schéma ne peut cependant pas modéliser finement les caractéristiques locales du mouvement et il revient à Ohm [92] de décrire le premier schéma de codage vidéo où un filtre temporel est appliqué dans le sens du mouvement des images, avant que ces dernières ne soient décomposées spatialement : c'est le schéma de codage vidéo $t+2D$. Ce schéma fait intervenir un filtre temporel compensé en mouvement (*Motion Compensated Temporal Filtering*) (MCTF) et est à l'origine de nombreux travaux sur le codage vidéo par ondelettes. Nous décrivons dans la suite le principe général de ce schéma de codage et présentons les premiers filtres temporels utilisés. Nous donnons alors un exemple détaillé de schéma de codage : le codec MC-EZBC, qui servira de prototype d'expérimentation aux travaux décrits dans les chapitres suivants. Nous dressons enfin un panorama des principaux développements et travaux de recherche menés sur ce schéma.

Principe général

Le principe du schéma de codage vidéo $t+2D$, illustré par la Fig. 2.4, repose sur l'utilisation d'un filtre temporel compensé en mouvement (MCTF), où l'on applique une transformée en ondelettes dans le sens du mouvement des images, pour tirer bénéfice de la redondance temporelle des trames. Les sous-bandes temporelles résultantes sont alors décomposées spatialement pour exploiter leur redondance spatiale. Elles sont ensuite quantifiées et codées de façon scalable par un codeur emboîté.

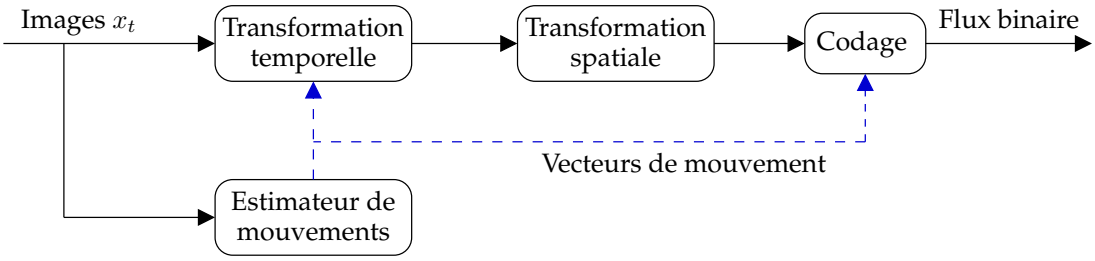


FIG. 2.4 – Schéma de principe d'un encodeur vidéo $t+2D$.

En parallèle du traitement des images, un estimateur de mouvement est placé en amont du schéma et fournit des champs de mouvement utilisés lors de la transformée temporelle. Ces champs sont alors encodés par un codeur sans perte puis intégrés au flux compressé. On remarquera que l'encodage est fait entièrement en boucle ouverte, contrairement aux schémas généraux des codeurs hybrides présentés dans la section 2.1. Il n'y a ainsi pas de rétroaction d'un décodeur assujetti à un débit donné et inclus dans l'encodeur, permettant d'obtenir aisément un schéma de codage scalable en qualité.

Extraction et scalabilité

La scalabilité du schéma $t+2D$ est assurée par un composant annexe, l'extracteur, qui permet de dégrader quasi-instantanément un flux compressé en un autre flux selon une qualité, résolution spatiale et temporelle spécifiées par l'utilisateur. Il permet par exemple d'obtenir une vidéo compressée à 128 kbits/s à partir d'une vidéo à 512 kbits/s ou de réduire la résolution d'une séquence vidéo compressée. Ce composant donne ainsi au schéma général les propriétés de scalabilité en qualité, temporelle et spatiale.

La structure même du flux compressé, dont un exemple est donné dans la section 2.2.4, permet à l'extracteur de supprimer rapidement les informations non nécessaires à la construction d'un nouveau flux de qualité inférieure. Ce mécanisme est rendu possible par les propriétés de scalabilité dyadiques inhérentes aux transformées temporelle et spatiale utilisées. La scalabilité temporelle permet ainsi d'obtenir des séquences vidéos de fréquence temporelle réduite d'un facteur dyadique, par suppression des sous-bandes temporelles de détail. La scalabilité spatiale permet d'obtenir des séquences vidéos de résolution spatiale réduite d'un facteur dyadique et est obtenue par suppression des sous-bandes spatiales de détail des sous-bandes temporelles.

La scalabilité en qualité repose, quant à elle, sur la stratégie utilisée par le codeur emboîté pour empaqueter les coefficients spatio-temporels. Ceux-ci étant organisés par plans de bits (*bitplanes*) ordonnés, il suffit de supprimer les plans de poids faible pour obtenir le débit souhaité. La scalabilité en qualité résultante est d'une granularité fine : il est possible de générer un flux compressé à un débit précis au kilobit par seconde près.

2.2.3 Premiers filtres temporels

Dénotons par x_t les images de la séquence vidéo où t représente l'indice temporel de la séquence et $\boldsymbol{v}_{2t+1}$ le champ de vecteurs prédisant l'image x_{2t+1} à partir de l'image x_{2t}. Chaque matrice x_t possède un indice spatial $\boldsymbol{n}$ et se note aussi $x_t(\boldsymbol{n})$, où $\boldsymbol{n}$ est un vecteur entier désignant un pixel de l'image. De plus, on note respectivement $h_{t,j}$ et $l_{t,j}$ les sous-bandes de détail et d'approximation issues de la décomposition temporelle au niveau j et l'indice j est omis lorsqu'un seul niveau de décomposition est considéré.

Schéma de codage fondateur de Ohm

Le schéma de codage vidéo fondateur de Ohm [92] utilise un filtre temporel basé sur la décomposition temporelle de Haar compensée en mouvement. Son principe, illustré par la Fig. 2.5, est le suivant : en considérant deux images consécutives x_{2t} et x_{2t+1}, il est possible d'estimer le champ de mouvement v_{2t+1} capable de prédire l'image x_{2t+1} à partir de l'image de référence x_{2t} et d'appliquer la transformée de Haar le long de ce mouvement. Avant de procéder, il est cependant nécessaire de distinguer les différents types de pixels mis en jeu dans ce filtrage. Tous les pixels de x_{2t+1} sont connectés à un pixel de l'image de référence x_{2t}. La réciproque n'est pas vraie : certains pixels de x_{2t} ne sont pas connectés comme p_r et sont qualifiés de "recouverts". Tous les pixels formant un couple de connexion unique comme p_2 et n_2 sont dits "connectés". En cas de connexion multiple d'un pixel de l'image x_{2t} avec plusieurs de x_{2t+1}, seul un couple de pixels est considéré comme "connecté" (ici, p_0 et n_0). Les autres pixels de x_{2t+1} seront qualifiés de "découverts" (ici, n_d), le choix étant fait selon le premier pixel rencontré lors du balayage de l'écran. Cette association entre pixels est unique et est déterminée entièrement par le champ de mouvement v_{2t+1} : elle pourra donc être reconstruite lors de la synthèse.

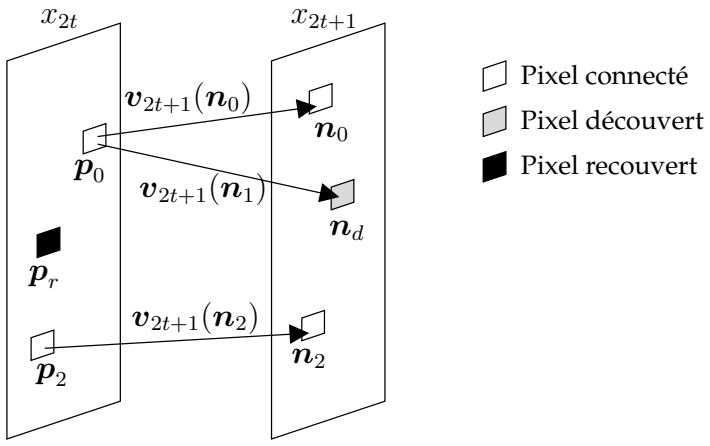


FIG. 2.5 – Filtrage de Haar compensé en mouvement selon la technique de Ohm.

Le filtrage de Haar compensé en mouvement permet de transformer le couple d'images x_{2t} et x_{2t+1} en sous-bandes h_t et l_t, respectivement synchrones. Il est effectué selon l'état

de connexion des pixels, grâce aux équations suivantes :

$$\begin{aligned}
&\text{Si le pixel } \boldsymbol{n} \text{ est connecté à } \boldsymbol{p}, \begin{cases} l_t(\boldsymbol{n}) &= \big(x_{2t+1}(\boldsymbol{n}) + x_{2t}(\boldsymbol{p})\big)/2 \\ h_t(\boldsymbol{p}) &= \big(x_{2t}(\boldsymbol{p}) - x_{2t+1}(\boldsymbol{n})\big)/2 \end{cases} \\
&\text{Si le pixel } \boldsymbol{n} \text{ est découvert, } l_t(\boldsymbol{n}) = x_{2t+1}(\boldsymbol{n}) \\
&\text{Si le pixel } \boldsymbol{p} \text{ est recouvert, } h_t(\boldsymbol{p}) = \big(x_{2t-1}(\widetilde{\boldsymbol{n}}) - x_{2t}(\boldsymbol{p})\big)/2
\end{aligned}$$

En cas de recouvrement, le pixel $h_t(\boldsymbol{p})$ est obtenu en utilisant le pixel $\widetilde{\boldsymbol{n}}$ de l'image précédente x_{2t-1}, où $\widetilde{\boldsymbol{n}}$ est associé au pixel $\widetilde{\boldsymbol{p}} = \boldsymbol{n}$. C'est donc une approximation basée sur le champ de vecteur x_{2t-1} de l'image précédente : en cas de mouvement complexe ou rapide, les pixels recouverts sont mal filtrés et créent des coefficients de grande amplitude dans les sous-bandes de détail h_t. Enfin, la reconstruction parfaite est possible si les vecteurs de mouvement sont entiers mais Ohm constate qu'elle reste cependant de bonne qualité quand la précision du mouvement est subpixellique.

Le schéma de Ohm est le premier schéma de codage vidéo par ondelettes scalable offrant des performances raisonnables, comparées aux schémas de codage vidéo hybrides. Cependant, son incapacité à gérer les mouvements subpixelliques et l'approximation utilisée lors du filtrage passe-haut des pixels recouverts nuisent beaucoup à son efficacité de codage. Il faudra attendre le schéma de Choi et Woods [38] en 1999 pour corriger cet inconvénient.

Filtrage de Haar compensé en mouvement selon Choi et Woods

Le filtrage temporel effectué dans le schéma de Choi et Woods [38] est très similaire à celui de Ohm. La différence essentielle réside dans le fait que les images de détail h_t sont ici synchrones avec les images impaires x_{2t+1} et les images d'approximation sont synchrones avec les images paires x_{2t}. Cette différence de traitement permet de s'assurer que tous les pixels soient filtrés passe-haut de façon homogène. En utilisant le même schéma illustré en Fig. 2.5, les équations de filtrage temporel du schéma de Choi et Woods s'écrivent :

$$\begin{aligned}
h_t(\boldsymbol{n}) &= \frac{\sqrt{2}}{2}\big(x_{2t+1}(\boldsymbol{n}) - x_{2t}(\boldsymbol{n} - \boldsymbol{v})\big) \\
\text{Si } \boldsymbol{n} \text{ est connecté, } l_t(\boldsymbol{n} - \bar{\boldsymbol{v}}) &= \frac{\sqrt{2}}{2}\big(x_{2t+1}(\boldsymbol{n} - \bar{\boldsymbol{v}} + \boldsymbol{v}) + x_{2t}(\boldsymbol{n} - \bar{\boldsymbol{v}})\big) \\
\text{Sinon, } l_t(\boldsymbol{n}) &= \sqrt{2}x_{2t}(\boldsymbol{n})
\end{aligned} \tag{2.1}$$

où $\boldsymbol{v} = \boldsymbol{v}_{2t+1}(\boldsymbol{n})$, $\bar{\boldsymbol{v}}$ est l'arrondi entier de $\boldsymbol{v}$ et où $\boldsymbol{n}$ est dit connecté si il existe un pixel $\boldsymbol{p}$ dans l'image x_{2t} tel que $\boldsymbol{n} - \bar{\boldsymbol{v}} = \boldsymbol{p}$. En effectuant la décomposition temporelle successive des images d'approximation l_t sur plusieurs niveaux, on est en mesure de réaliser l'analyse temporelle de Haar compensée en mouvement d'un groupe d'images, illustrée en Fig. 2.6.

La simple modification de Choi et Woods permet d'obtenir des images de détail uniformément filtrées ; elle conduit à une amélioration significative de l'efficacité du schéma de codage par rapport à celui de Ohm et surpasse même le codeur hybride MPEG-1, pourtant non-scalable. Cependant, ce schéma n'est pas en mesure de gérer les mouvements subpixelliques car l'arrondi utilisé dans le filtrage passe-bas interdit la reconstruction parfaite. Il est nécessaire d'utiliser une modification spécifique [62] pour pouvoir l'assurer en cas d'utilisation de mouvements au demi-pixel. De plus, l'extension de ce type de schéma

à des filtres bidirectionnels n'est pas une tâche aisée. Le lifting temporel, proposé en 2001 par Pesquet-Popescu [108], permet en fait de résoudre simplement *tous* les problèmes liés à la reconstruction parfaite lors de la construction d'un filtre temporel compensé en mouvement et est décrit dans la sous-section suivante.

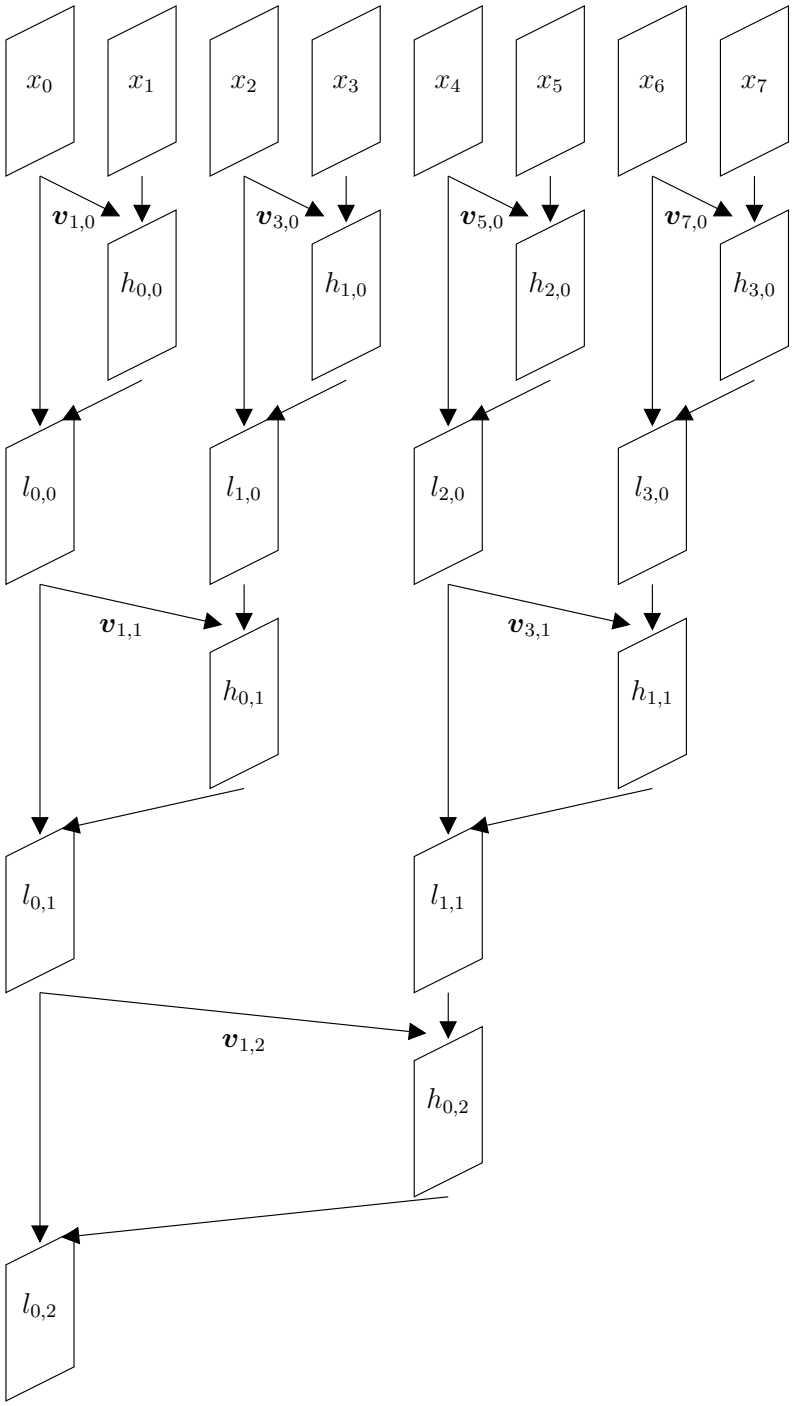


FIG. 2.6 – Décomposition temporelle de Haar d'un groupe d'images sur 3 niveaux.

Lifting temporel

Le schéma de lifting temporel consiste à utiliser la formulation lifting, rappelée en section 1.3.1, pour exprimer une transformée temporelle compensée en mouvement. Introduit par Pesquet-Popescu [108], il remplace les équations de filtrage du schéma de Choi et Woods (2.1) par les équations suivantes, en conservant les mêmes notations :

$$
\begin{aligned}
h_t(\boldsymbol{n}) &= \frac{\sqrt{2}}{2}\big(x_{2t+1}(\boldsymbol{n}) - x_{2t}(\boldsymbol{n}-\boldsymbol{v})\big) \\
l_t(\boldsymbol{p}) &= \sqrt{2}\, x_{2t}(\boldsymbol{p}) + \alpha\, h_{t-1}(\boldsymbol{p}+\boldsymbol{v}) \\
&\text{avec } \begin{cases} \alpha = 1 & \text{si } \boldsymbol{p} \text{ est connecté } (\exists\, \boldsymbol{n} \text{ tel que } \boldsymbol{n} - \bar{\boldsymbol{v}} = \boldsymbol{p}) \\ \alpha = 0 & \text{si } \boldsymbol{p} \text{ n'est pas connecté} \end{cases}
\end{aligned}
$$

Dans le cas du filtre temporel de Haar compensé en mouvement, cette modification du filtre passe-bas permet de construire un transformée toujours inversible par simple renversement des étapes de lifting et ceci même dans le cas subpixellique.

De plus, cette subtile modification a une portée bien plus large que le simple cas du filtre temporel de Haar compensé en mouvement. Comme souligné par les auteurs, le schéma lifting autorise l'utilisation d'opérateurs temporels de prédiction P et de mise à jour U non-linéaires, tout en conservant l'inversibilité du schéma. Par exemple, il est possible d'inclure une étape de compensation de mouvement ou d'interpolation subpixellique au sein de ces opérateurs. On peut alors écrire dans le cas général :

$$
\begin{aligned}
h_t &= x_{2t+1} - P\big(\{x_{2t}\}_{t\in\mathbb{N}}, \{\boldsymbol{v}_t\}_{t\in\mathbb{N}}\big) \\
l_t &= x_{2t} + U\big(\{h_t\}_{t\in\mathbb{N}}, \{\boldsymbol{v}_t\}_{t\in\mathbb{N}}\big)
\end{aligned}
$$

où P et U sont des opérateurs quelconques. Le schéma lifting permet ainsi la construction de transformées temporelles plus longues, inversibles et dotées d'opérateurs bidirectionnels comme la transformée 5/3 compensée en mouvement, décrite en détails dans le chapitre 3. On notera enfin les travaux ultérieurs de Secker et Taubman [125] qui ont confirmé l'intérêt du schéma lifting lors de la construction du filtre temporel mis en jeu dans le schéma de codage vidéo $t+2D$.

2.2.4 Exemple de schéma de codage t+2D : le codec MC-EZBC

Le codec MC-EZBC est un schéma de codage vidéo $t+2D$ issu des travaux de Hsiang et Woods [61] qui étend les travaux de Choi en utilisant le codeur emboîté EZBC, rappelé en section 1.2.4. Ce codec est à la base de notre schéma de codage vidéo, sur lequel une grande partie de nos expérimentations ont été menées. Nous nous proposons dans cette section de décrire ses caractéristiques détaillées.

Filtrage temporel

Afin d'éliminer la redondance temporelle des images d'une séquence vidéo, le codec MC-EZBC utilise une transformée temporelle de Haar basée sur celle de Choi et Woods, décrite dans la section précédente. Elle met en jeu une estimation de mouvement subpixellique, pouvant aller jusqu'au $1/8^{ème}$ de pixel et utilise une méthode de compensation de mouvement avec chevauchement des blocs, permettant d'amoindrir les effets de blocs visibles à bas débits. Bien que relativement efficace, cette transformée est cependant

seulement mono-directionnelle et n'utilise qu'une seule image pour assurer la prédiction temporelle. La transformée temporelle 5/3, décrite dans le chapitre 3 résout ce problème et permet un gain substantiel de l'efficacité globale de codage.

Estimation, élagage et codage des vecteurs de mouvement

L'estimation des champs de mouvement au sein du codec MC-EZBC est faite par l'algorithme *Hierarchical Variable Size Block Matching* (HVSBM), décrit par Choi [38]. Comme son nom l'indique, c'est un algorithme d'appariement hiérarchique de blocs de taille variable où le mouvement est d'abord estimé sur des gros blocs puis raffiné sur les subdivisions de ces blocs. Pour des raisons de simplicité, l'estimation du mouvement est faite seulement sur la composante de luminance Y des images. Cependant, on notera que des gains significatifs en PSNR moyen peuvent être obtenus [17] en exploitant les composantes de chrominances durant l'estimation de mouvement.

Le principe de l'algorithme HVSBM est le suivant. L'algorithme démarre dans l'image courante avec un bloc de grande taille, typiquement 64×64 pixels et recherche un bloc similaire dans l'image de référence, en minimisant un critère d'erreur quadratique moyenne. Le vecteur mouvement du bloc est mémorisé et le bloc est alors subdivisé en 4 sous-blocs. On relance la procédure de recherche de blocs pour les sous-blocs, en mémorisant leurs vecteurs mouvement. On procède récursivement, jusqu'à obtenir des blocs de taille 4×4. On peut alors construire un arbre quaternaire (*quad-tree*) contenant tous les vecteurs mouvement des blocs et ceux de leurs sous-blocs. La recherche du mouvement est faite par un algorithme du type *full-search* où l'on parcourt exhaustivement une fenêtre de recherche. Dans notre implémentation, cette fenêtre est initialisée à 4×4 pixels lors du premier niveau temporel et est doublée à chaque niveau suivant. Elle est aussi doublée si l'erreur quadratique entre le bloc courant et le bloc candidat est supérieure à un seuil donné, pour éviter les erreurs d'appariement de blocs en cas de mouvement rapide.

L'arbre ainsi créé décrit un champ de mouvement quasiment dense, de résolution 4 fois inférieure à celle de l'image. Ce champ est bien sûr trop gros et donc trop coûteux pour être encodé tel quel. On utilise alors une procédure d'élagage de l'arbre, basée sur le calcul du Lagrangien $D + \lambda R$ de chaque nœud, où D est l'erreur quadratique moyenne créée par le vecteur associé au nœud et R son débit estimé par entropie. En utilisant une contrainte λ fixée, on retire alors tous les nœuds de l'arbre dont la réduction de distorsion n'est pas rentable au vu de leur coût, selon le critère Lagrangien. La procédure d'élagage est appliquée récursivement sur chaque nœud, en parcourant l'arbre de bas en haut.

Au final, on obtient une description hiérarchique du champ de mouvement, avec de larges zones en cas de mouvement uniforme et de petites sections, correspondantes aux objets de petite taille. Le codage des composantes du champ de mouvement est alors fait sans perte, au moyen d'une prédiction différentielle et d'un codeur arithmétique sans mémoire. Le codage des champs de mouvement dans le codec MC-EZBC n'est donc pas scalable et ceci nuit à l'efficacité globale de codage lors de l'utilisation de la scalabilité spatiale. Cependant, plusieurs travaux sur la scalabilité des vecteurs mouvements [9, 23] sont rapportés dans la section 2.2.5.

Organisation du bitstream dans MC-EZBC

Le flux vidéo compressé, nommé aussi *bitstream*, est organisé de manière à fournir une représentation de la séquence vidéo scalable aisément extractible. Par simple suppres-

sion de sous-bandes spatio-temporelles ou de plans de bits constituant ces dernières, l'extracteur peut fournir une scalabilité temporelle, spatiale et en débit. La description de l'organisation hiérarchique du bitstream est illustrée par la Fig. 2.7 et correspond aux sous-bandes temporelles issues de la décomposition du GOP de la Fig. 2.6.

Le bitstream MC-EZBC est constitué par la concaténation de GOPs (*Group Of Pictures*) élémentaires et indépendants. Un GOP est la représentation compressée d'une suite de 2^L images où L est la profondeur de l'analyse temporelle. Il contient les champs de mouvement compressés, les sous-bandes temporelles codées et diverses informations groupées dans un entête. Chaque GOP peut être décodé indépendamment de tous les autres. Cependant, cela ne signifie pas nécessairement qu'un GOP puisse être reconstruit indépendamment car certaines transformées temporelles, notamment la transformée 5/3, nécessitent un contexte de GOP pour pouvoir reconstruire le GOP courant.

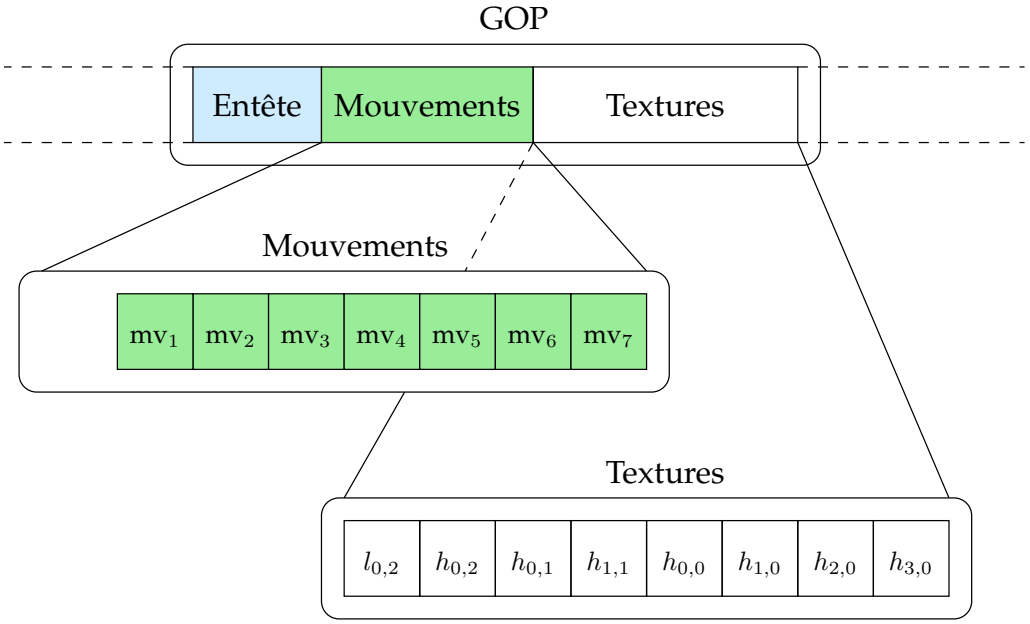


FIG. 2.7 – Organisation hiérarchique du flux vidéo *bitstream* compressé MC-EZBC.

Le bloc d'entête contient le nombre d'images contenues dans le GOP et la taille du GOP en octets. De plus, sa structure flexible permet de lui adjoindre d'autres informations comme des points de coupure de scènes, des informations de modes de blocs, etc... Néanmoins, ces dernières fonctionnalités ne sont pas utilisées et seules les informations de mouvement et de texture sont utilisées par notre prototype durant le codage.

Le bloc de mouvement contient les champs de mouvement compressés. Ces derniers sont organisés de manière synchrone avec les sous-bandes de détail contenues dans le bloc de textures, comme illustré par la pyramide de décomposition temporelle de la Fig. 2.6. Le champ mv_1 représente ainsi l'information de mouvement nécessaire pour prédire la sous-bande temporelle $h_{0,2}$. On rappelle que les champs de mouvement sont codés ici sans perte et de façon non-scalable, par un codeur arithmétique sans mémoire.

Les sous-bandes temporelles sont empaquetées dans le bloc de textures. Elles sont ordonnées dans le sens classique utilisé lors d'une décomposition en ondelettes monodimensionnelle. Comme illustré en Fig. 2.6, les sous-bandes d'approximation $l_{0,2}$ et de dé-

tail $h_{0,2}$ du dernier niveau temporel sont les premières à être encodées. Viennent ensuite les sous-bandes de détail $\{h_{0,1}, h_{1,1}\}$ de l'avant-dernier temporel puis celles $\{h_{t,0}\}_{t \in \mathbb{N}}$ du premier niveau temporel. Les sous-bandes temporelles sont des images transformées spatialement et encodées par l'algorithme EZBC, rappelé en section 1.2.4. Elles sont donc organisées en sous-bandes spatiales, qui sont elles-mêmes agencées par plans de bits, en commençant par ceux de poids fort. Le bitstream proposé est donc emboîté sous la forme {Temporel, Spatial, Plans de bits}.

La structure hiérarchique ainsi présentée permet d'obtenir aisément une scalabilité de type temporelle, spatiale et en qualité. La scalabilité temporelle est obtenue par suppression des champs de mouvement et sous-bandes temporelles des niveaux temporels les plus élevés. L'extracteur permet ainsi d'obtenir une séquence vidéo de fréquence temporelle réduite d'un facteur dyadique. De même, la scalabilité spatiale est obtenue par suppression des sous-bandes spatiales des niveaux spatiaux les plus élevés. Enfin, la scalabilité en débit est simplement réalisée par suppression des plans de bits de poids faible.

Il est à noter que d'autres types d'organisation de bitstreams existent mais tous offrent une structure facile à élaguer pour assurer la scalabilité spatiale, temporelle et en qualité. Bottreau et al. décrivent ainsi une structure hiérarchique [24] similaire à celle du codec MC-EZBC, mais emboîtée dans l'ordre {Plans de bits, Temporel, Spatial}. D'autres structures hiérarchiques mettant en œuvre des stratégies de protection et des codes correcteurs d'erreurs sont utilisées dans des schémas de codage vidéo robuste, adaptés en cas de possibilité de perte d'information. Cependant, de telles techniques de codage robuste sont en dehors de la portée de ce document et nous nous restreignons à un cadre où aucune erreur de transmission ne peut affecter le bitstream.

Filtrage spatial et codage des textures

La transformation spatiale des sous-bandes temporelles est réalisée par une décomposition pyramidale dyadique séparable 2D au moyen des ondelettes biorthogonales 9/7, décrites dans la section 1.2.3. La décomposition est faite sur un nombre suffisant de niveaux afin de s'assurer que la dernière sous-bande spatiale d'approximation soit d'une taille supérieure à 8×8 pixels. Le choix de la transformée biorthogonale 9/7 est inspiré des travaux d'Antonini [18] qui montrèrent que cette transformée offrait la meilleure efficacité de codage lors de la compression d'images *naturelles*. Ces travaux ont été confirmés expérimentalement et ultérieurement par Villasenor [159].

On remarquera sur la Fig. 3.5 (page 71) et sur la Fig. 5.3 (page 128), que les sous-bandes temporelles d'approximation résultant du filtrage temporel passe-bas ont un aspect très similaire à des images naturelles. Ceci justifie ainsi l'utilisation des ondelettes biorthogonales 9/7. Cependant, les sous-bandes temporelles de détail, résultant du filtrage temporel passe-haut, ont quant à elles un aspect visuel totalement différent. La transformée 9/7 n'est alors peut-être pas le filtre le plus adapté à leur transformation. L'optimisation et la recherche d'une décomposition idéale adaptée à la décorrélation des sous-bandes temporelles de détail sont largement abordées dans les chapitres 5 et 6.

Le codage effectif des coefficients d'ondelettes résultant de la décomposition spatiale des sous-bandes temporelles est effectué par le codec emboîté EZBC, décrit dans la section 1.2.4. Chaque sous-bande temporelle est encodée indépendamment, selon la méthode d'allocation débit-distorsion du codec EZBC.

2.2.5 Améliorations apportées au schéma t+2D

Suite au succès du schéma de codage vidéo $t+2D$ et conforté par les perspectives du lifting temporel, de nombreuses améliorations et optimisations ont été apportées au schéma original afin d'améliorer son efficacité de codage. Nos propres travaux s'inscrivent dans ces développements et sont détaillés dans les chapitres suivants. Nous nous proposons cependant de dresser dans cette section un inventaire des principales avancées menées sur le schéma de codage $t + 2D$.

Modèles de mouvement inversibles et grille triangulaire déformable

L'utilisation de champs de mouvement estimés par blocs permet de traduire correctement les mouvements translationnels présents dans une séquence vidéo. Cependant, les champs résultants ne sont pas inversibles et entraînent la création de zones déconnectées et découvertes lors de l'étape de mise à jour du filtrage temporel. Dans le cas particulier du filtre temporel de Haar, Konrad [69] a pourtant montré l'existence d'une transformée temporelle de Haar *transverse* où le mouvement ne nécessite pas d'inversion durant l'étape de mise à jour. Ce mécanisme n'est cependant pas généralisable à des transformées bidirectionnelles.

Les travaux de Secker et Taubman [126, 127] préconisent quant à eux l'utilisation d'un modèle de mouvement basé sur une grille triangulaire déformable de type *mesh*, presque toujours inversible (sauf en cas de retournement de maille). Ce type de modèle reste cependant coûteux à encoder, difficile à estimer et parfois même mal adapté pour décrire un mouvement translationnel rapide.

Diverses stratégies de mise à jour lors du filtrage temporel

Le problème de la création de zones déconnectées et découvertes lors du filtrage passe-bas temporel a donné lieu à la publication de nombreux travaux sur diverses stratégies de mise à jour. Hanke [57] propose par exemple l'utilisation d'un filtre passe-bas lors de la mise à jour, afin d'amoindrir les discontinuités entre zones connectées et non-connectées. La puissance du filtre utilisée est paramétrée par un critère basé sur la divergence locale du champ de mouvement et permet donc la reconstruction parfaite. L'auteur observe un gain en efficacité de codage mitigé mais constate une diminution de la fluctuation temporelle de PSNR au cours du temps.

Lors de la présence d'un pixel connecté de façon multiple, la transformée temporelle de Choi préconise le filtrage passe-bas avec le premier pixel rencontré dans le sens du balayage de l'écran. Cette stratégie déterministe est aisément reconstructible au décodage mais n'est pas vraiment justifiée. Pesquet-Popescu [108] utilise plutôt des critères basés sur l'énergie du mouvement local ou sur la distorsion pour choisir un pixel mieux adapté au filtrage passe-bas. Une autre approche décrite par Tillier [145] consiste à prendre la *moyenne* des pixels auxquels le pixel courant est connecté : l'auteur démontre que cette stratégie minimise l'erreur de reconstruction et conduit à un gain significatif de l'efficacité de codage.

D'autres travaux [78, 133] préconisent une mise à jour adaptative par seuillage et pondération, basée sur des critères psychovisuels et des mesures locales d'activité. On notera enfin les structures d'André [13] et de van der Schaar [157], mettant en œuvre des transformées à longue prédiction temporelle sans étape de mise à jour, s'affranchissant ainsi du problème d'inversibilité des champs de mouvement.

Codage t+2D en boucle fermée

Le schéma de codage traditionnel $t+2D$ illustré en Fig. 2.4 est en boucle ouverte : les images sont prédites temporellement par rapport aux images originales, contrairement au schéma hybride où les images sont prédites grâce aux images reconstruites. L'inconvénient de la prédiction en boucle ouverte réside dans le fait que le champ de mouvement estimé est optimal du point de vue de l'encodeur mais pas du point de vue du décodeur, qui n'a accès qu'aux images reconstruites.

Rusert a proposé un schéma de codage vidéo $t+2D$ [119] en boucle fermée où les images sont prédites par rapport à des images reconstruites pour un certain débit de contrôle, connu à l'encodage. Il obtient alors une efficacité de codage légèrement supérieure à celle d'un schéma en boucle ouverte pour des débits proches du débit de contrôle, mais inférieure dans les autres cas. Le schéma proposé par Xiong [163] utilise une approche similaire et obtient des résultats comparables.

Utilisation de différents modes de prédiction temporelle

Contrairement au filtre temporel de Haar compensé en mouvement où tous les blocs d'une image sont prédits par rapport à l'image précédente, le schéma de codage vidéo SVC utilise un algorithme de prédiction temporelle adaptatif. Chaque bloc est en effet prédit différemment en fonction du *mode* de prédiction qui minimise son coût codage parmi une dizaine de modes de prédiction (intra, monodirectionnel passé, monodirectionnel futur, bidirectionnel, direct...).

Rusert [121] a proposé l'utilisation d'une prédiction temporelle basée sur différents modes au sein du schéma de codage MC-EZBC et a observé des gains significatifs de l'efficacité de codage par rapport à un filtre temporel statique. On remarquera enfin que le codec Vidwav [9] et le schéma de codage vidéo mis en œuvre par Luo [77] utilisent aussi plusieurs modes de prédiction temporelle.

Autres filtres temporels

L'utilisation du schéma lifting temporel permet d'étendre simplement le filtre temporel de Haar au filtre temporel $5/3$ compensé en mouvement. Ce filtre bidirectionnel correspond à l'utilisation de la transformée en ondelettes $5/3$, rappelée en section 1.2.3, dans le sens du mouvement des images. Les premiers travaux sur le filtre temporel $5/3$ compensé en mouvement sont mentionnés dans [94, 125] et offrent une efficacité de codage comparable aux schémas de codage hybride. Cependant, nos travaux [106, 146] sur l'étude systématique du filtre temporel $5/3$ et sa mise en œuvre au sein du schéma de codage MC-EZBC ont montré un gain en efficacité de codage significatif comparé au schéma de codage hybride. Ces recherches constituent le point d'achoppement entre l'état de l'art et le début de mes travaux, et sont rapportés dans le chapitre suivant.

D'autres structures de prédiction temporelle, basées sur des supports plus longs comme le filtrage UMCTF [157] (*Unconstrained MCTF*) ou les filtres temporels $(N,0)$ [13] (sans étape de mise à jour) ont aussi été proposés. Ces techniques conduisent à une augmentation de l'efficacité de codage en présence d'un mouvement non-uniforme et améliorent la qualité visuelle des sous-bandes temporelles d'approximation.

Les filtres temporels précédemment décrits sont généralement basés sur des transformées en ondelettes dyadiques et n'offrent ainsi que des facteurs de scalabilité temporelle d'ordre 2. Afin d'élargir cette gamme de facteurs, Tillier [143] a proposé des bancs

de filtres monodirectionnels 3-bandes, offrant une meilleure efficacité de codage que les filtres de Haar et capables de fournir des facteurs de scalabilité d'ordre 3. Une extension au cas bidirectionnel a été proposée ultérieurement [144], améliorant encore l'efficacité de codage.

Codage 2D+t Inband et schéma de codage 2D+t+2D

Dans le schéma de codage vidéo $t + 2D$, les images sont d'abord transformées temporellement puis spatialement. Les techniques d'estimation et de compensation de mouvement par blocs sont efficaces et rapides mais ont l'inconvénient d'introduire des discontinuités de type bloc lors de la compensation. Des solutions utilisant le chevauchement des blocs sont possibles mais ne se révèlent pas entièrement satisfaisantes.

Suite aux travaux de Park [95] sur l'estimation de mouvement dans le domaine ondelettes, Andreopoulos a introduit en 2002 le schéma de codage vidéo $2D + t$ [14, 15], où les images sont tout d'abord décomposées spatialement par ondelettes puis transformées temporellement. Ce schéma permet ainsi d'estimer le mouvement et de compenser les images dans le domaine transformé, afin de réduire les effets de blocs : on parle aussi de schéma de codage *Inband*. La transformée en ondelettes n'étant pas invariante par translation, il est toutefois nécessaire d'utiliser une base d'ondelettes redondantes pour décomposer les images et appliquer des algorithmes d'estimation et de compensation de mouvement spécifiques [95]. Une telle structure permet l'amélioration de la qualité visuelle des séquences vidéo décodées à bas débits, moyennant une baisse légère du PSNR.

Une extension du schéma Inband a été décrite par Mehrseresht et Taubman [88] et suggère l'utilisation d'une structure de type $2D + t + 2D$. Elle consiste en la prédécomposition spatiale des images, suivie d'une transformation temporelle et de la continuation de la décomposition spatiale en ondelettes. Cette structure possède l'avantage de préserver l'efficacité de codage offerte par la structure $t+2D$ en terme de PSNR, tout en minimisant l'apparition d'artefacts de type blocs lors du décodage de séquences à bas débits.

Scalabilité et codage des champs de mouvement

Dans le schéma de codage vidéo $t + 2D$ de Ohm, les champs de mouvement sont encodés sans perte et de manière non-scalable. Ceci nuit aux propriétés de scalabilité du schéma général car ces champs sont incompressibles et peuvent prendre une place trop importante pour les résolutions faibles : il n'est ainsi pas nécessaire de posséder des champs de mouvement de taille 4CIF lors du décodage d'une séquence au format QCIF.

Afin de résoudre ce problème, de nombreux travaux proposent l'utilisation d'un codage scalable des champs de mouvement : par précision et par plans de bits [23, 24], par différentes couches spatiales [9] ou en utilisant une transformée spatiale en ondelettes [127]. Toutes ces stratégies conduisent à l'amélioration de l'efficacité de codage lors de l'utilisation de la scalabilité spatiale ou temporelle. On notera aussi les travaux de Tsai [150] sur l'encodage de champs de mouvement par balayage des valeurs selon la courbe fractale de Hilbert, conduisant à une légère réduction de leur coût de codage.

Schéma de codage Vidwav

Le schéma de codage Vidwav [9] est issu des travaux de Song, Wu, Xiong et Xu [133, 164, 165] sur l'algorithme 3D-ESCOT et a été rendu public en 2004 lors de l'appel à propositions MPEG. C'est un schéma de codage $t + 2D$ efficace, capable de gérer les modes de

prédiction temporelle et permettant l'utilisation de la structure $2D+t+2D$. Il possède une structure en couches et peut offrir une scalabilité en qualité fine par un encodage progressif des coefficients de texture. Le codec Vidwav offre une efficacité de codage comparable au schéma MC-EZBC en résolution nominale mais donne de meilleurs résultats en scalabilité spatiale. Tout comme le codec MC-EZBC, ce codec nous servira à expérimenter certains de nos travaux de recherche décrits dans les chapitres suivants.

Afin d'améliorer les performances du codec Vidwav, Leonardi et al. ont proposé l'architecture STool (*STool*) [74], fondée sur un schéma $2D + t + 2D$ où la prédécomposition spatiale en ondelettes est faite sur plusieurs couches. La présence de ces dernières permet une prédiction de la sous-bande spatio-temporelle d'approximation par rapport à celle de la couche de résolution spatiale inférieure, afin de réduire la redondance spatiale. Cette prédiction est réalisée en boucle fermée, après décodage de la sous-bande temporelle. Enfin, il est possible d'utiliser une variante 3D [73] du codeur emboîté morphologique EMDC, décrit en section 1.2.4, afin d'améliorer encore l'efficacité de codage du schéma.

2.3 Conclusion

Les efforts de recherche menés depuis plus de vingt ans sur le codage vidéo scalable ont conduit à la normalisation de plusieurs algorithmes dont les célèbres codecs de la famille MPEG et H.26X, appartenant à la classe des schémas de codage hybride. Dotés d'une grande efficacité de codage, ces schémas ne sont cependant pas en mesure de fournir directement une représentation scalable d'une séquence vidéo et il faut recourir à une structure en couches pour obtenir une scalabilité spatio-temporelle et en qualité grossière. Les extensions MPEG-4 FGS et SVC ont alors été proposées pour permettre une scalabilité fine en qualité tout en combinant une scalabilité spatio-temporelle mais ne se sont pas révélées entièrement satisfaisantes.

En parallèle de ces travaux, les recherches sur les schémas de codage vidéo par ondelettes $t + 2D$, mettant en œuvre une transformée temporelle appliquée selon le mouvement des images continuèrent. L'avènement du lifting temporel a alors révolutionné la donne en permettant l'élargissement du support de prédiction temporelle et en autorisant l'introduction d'opérateurs non-linéaires quelconques au sein de filtres spatio-temporels. Cette technique a permis la construction de schémas de codage par ondelettes offrant une efficacité de codage comparable avec les schémas hybrides, tout en possédant des propriétés de scalabilité spatiale, temporelle et en qualité fine. De nombreuses améliorations et optimisations ont ensuite été apportées pour améliorer l'efficacité de codage de ces structures : nos travaux s'inscrivent dans ces développements et sont détaillés dans les chapitres suivants.

Deuxième partie

Mise en œuvre d'un codec vidéo scalable t+2D

Chapitre 3

Filtrage temporel 5/3

Le schéma de codage $t + 2D$, décrit dans la section précédente, est une architecture de codage en boucle ouverte permettant la description scalable et parcimonieuse d'une séquence vidéo. Il repose sur l'utilisation d'une transformée temporelle appliquée le long du mouvement des images afin d'exploiter leur redondance temporelle. La plupart des filtres temporels utilisés sont basés sur une transformée de Haar compensée en mouvement ; cette dernière possède une bonne efficacité de décorrélation temporelle et reste simple à mettre en œuvre. Cependant, le filtre temporel de Haar met en jeu une prédiction temporelle monodirectionnelle et n'utilise qu'une seule image de référence pour prédire une image courante. Que peut-on espérer d'une transformée plus longue ?

La transformée en ondelettes $5/3$ est bidirectionnelle, possède un support plus large et constitue une candidate idéale pour assurer la transformée temporelle mise en jeu dans un schéma de codage $t + 2D$. Nous nous proposons de décrire dans la la section 3.1 comment le schéma de lifting temporel permet de construire un filtre temporel $5/3$ compensé en mouvement, doté d'une très bonne efficacité de décorrélation temporelle.

Nous présentons alors dans la section 3.2 les résultats expérimentaux obtenus lors de la mise en œuvre du filtre temporel $5/3$ au sein de notre schéma de codage vidéo. Des mesures de performance objectives sont présentées et nous comparons l'efficacité de notre schéma avec des codecs vidéo actuels, couramment utilisés. Ces résultats serviront de référence aux optimisations menées dans les chapitres 4 et 5.

Ces travaux font suite à ceux de Tillier [146] sur le filtrage temporel $5/3$ et ont conduit à la publication d'un article général de revue [106] sur la compensation de mouvement et l'utilisation du schéma lifting en codage vidéo scalable.

3.1 Filtrage temporel 5/3 compensé en mouvement

3.1.1 Notations

Introduisons tout d'abord quelques notations qui seront utilisées tout au long de cette section. Les images de la séquence vidéo sont notées x_t où t est l'indice temporel de l'image. Chaque matrice x_t possède un indice spatial $\boldsymbol{n}$ et se note aussi $x_t(\boldsymbol{n})$ où $\boldsymbol{n}$ est un vecteur entier désignant un pixel de l'image. On ne décrit que le cas noir et blanc où les valeurs des matrices représentent la luminance du pixel considéré.

Les sous-bandes d'approximation issues de la décomposition temporelle au niveau j et résultant du filtrage temporel passe-bas sont notées $l_{t,j}$. Les sous-bandes de détail, résultant du filtrage temporel passe-haut sont notées quant à elles $h_{t,j}$. Par décompositions successives des sous-bandes d'approximation, il est aisé d'obtenir une analyse multirésolution et l'indice j est omis lorsqu'un seul niveau de la décomposition est considéré. Nous utiliserons alors les notations l_t et h_t.

3.1.2 Lifting temporel

Comme vu dans la section 2.2.3, la formulation lifting permet de mettre en œuvre simplement une transformée en ondelettes quelconque dans le sens du mouvement d'une séquence vidéo. Considérons une transformée appliquée sur les images x_t dont la structure lifting possède une étape de prédiction et une étape de mise à jour. Les sous-bandes d'approximation l_t et de détail h_t résultantes sont alors obtenues par :

$$h_t^0 = x_{2t+1} - P\big(\{x_{2t}\}_{t\in\mathbb{N}}\big) \tag{3.1}$$
$$l_t^0 = x_{2t} + U\big(\{h_t\}_{t\in\mathbb{N}}\big) \tag{3.2}$$
$$h_t = \zeta_h\, h_t^0$$
$$l_t = \zeta_l\, l_t^0$$

où P est l'opérateur de prédiction, U l'opérateur de mise à jour, ζ_h et ζ_l les constantes de normalisation et où $\{x_{2t}\}_{t\in\mathbb{N}}$ représente l'ensemble des images d'indice pair de la séquence vidéo et $\{h_t\}_{t\in\mathbb{N}}$ l'ensemble des images de détail.

Le formalisme de la structure lifting garantit l'inversibilité du schéma, quels que soient les opérateurs P et U. En particulier, ils n'ont pas besoin d'être linéaires ni même inversibles. Les images originales peuvent être ainsi reconstruites par un simple retournement des étapes de lifting et une négation des signes :

$$l_t = l_t^0/\zeta_l$$
$$h_t = h_t^0/\zeta_h$$
$$x_{2t} = l_t^0 - U\big(\{h_t\}_{t\in\mathbb{N}}\big) \tag{3.3}$$
$$x_{2t+1} = h_t^0 + P\big(\{x_{2t}\}_{t\in\mathbb{N}}\big) \tag{3.4}$$

Les opérateurs de prédiction P et de mise à jour U sont dits spatio-temporels car ils disposent de la totalité des pixels d'un ensemble temporel d'images pour effectuer leur filtrage. Ainsi, tous les pixels des images d'indice pair $\{x_{2t}\}_{t\in\mathbb{N}}$ peuvent ainsi être utilisés par l'opérateur P pour prédire chaque pixel de l'image courante x_{2t+1}. De même, l'opérateur de mise à jour U dispose de tous les pixels de l'ensemble des images de détail $\{h_t\}_{t\in\mathbb{N}}$ pour effectuer son filtrage passe-bas.

Nous avons rappelé dans la section 2.2.2 qu'il est nettement plus efficace de décomposer temporellement les images *dans le sens du mouvement* en utilisant les mécanismes d'estimation et de compensation de mouvement classiquement utilisés en codage vidéo. Les travaux de Pesquet-Popescu [108] ont de plus mis en évidence que ces mécanismes non-linéaires pouvaient être très naturellement introduits dans la structure lifting précédente, conduisant ainsi à une structure lifting compensée en mouvement.

Les opérateurs de prédiction P et de mise à jour U doivent être donc modifiés pour tenir compte du mouvement. En utilisant les champs préalablement fournis par un module d'estimation de mouvements, ils peuvent ainsi mettre en correspondance les zones mouvantes présentes dans les images avant de les filtrer. On peut voir ce module d'estimation comme une pré-décision, influençant les opérateurs de prédiction et de mise à jour. Afin de pouvoir reconstruire les images, les champs de mouvement sont transmis à part et encodés sans perte. La Fig. 3.1 illustre la structure en lifting d'un filtre temporel compensé en mouvement.

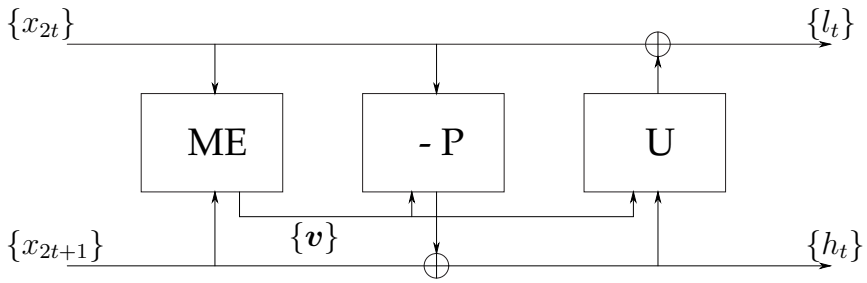


FIG. 3.1 – Structure lifting d'un filtre temporel compensé en mouvement.

Bien que les champs de mouvement utilisés dans l'opérateur de prédiction P et de mise à jour U ne soient pas exactement les mêmes, nous les considérons en pratique comme tels. En effet, pour des raisons de complexité et pour économiser le débit d'information, les champs utilisés lors de la mise à jour sont calculés par inversion des champs estimés lors de la prédiction. Ceci revient à faire l'hypothèse d'être en présence d'un mouvement homogène. Cependant, si cette hypothèse n'est pas vérifiée, l'inversion du champ n'est pas directement possible et une étude plus poussée est nécessaire. Plusieurs travaux [38, 92] préconisent ainsi une gestion particulière des pixels non-connectés ou connectés de façon multiple dans le cas du filtre temporel de Haar. Dans le cas du filtre temporel 5/3, nous présentons une étude complète dans la section 3.1.3. On notera cependant que l'utilisation d'un modèle de mouvement basé sur des grilles déformables de type *mesh* [126] et non sur des blocs permet d'obtenir un mouvement continu où l'inversion est toujours possible. Enfin, d'autres travaux [152] exploitent la similarité des champs de mouvement pour pouvoir réduire la quantité de mouvement à transmettre.

L'opérateur de prédiction P est donc une fonction de plusieurs images $\{x_{2t}\}_{t\in\mathbb{N}}$ choisie pour approximer au mieux l'image x_{2t+1}. Ceci permet ainsi d'aboutir à des images de détail h_t de faible dynamique, à priori plus simple à coder. La prise en compte du mouvement permet de construire alors un filtre temporel s'appliquant selon le sens du mouvement, par appariement préalable des pixels. Tout comme dans le cas des codecs hybrides, il est possible d'introduire des techniques de compensation de mouvement subpixellique de manière à améliorer la prédiction temporelle. Ceci conduit naturellement à la construction d'un opérateur P spatio-temporel où l'estimation de mouvement est précédée par une interpolation des images de références. Enfin, des méthodes de compensation de mouvement avec recouvrement (*overlap*) sont aussi simples à mettre en œuvre.

3.1.3 Construction d'une transformée 5/3 compensée en mouvement

Considérons une transformée 5/3 biorthogonale appliquée sur l'axe temporel où les opérateurs de prédiction et de mise à jour sont des filtres possédant un support de deux échantillons. Nous pouvons alors décrire la transformée au moyen des paramètres α, β, γ, δ, ζ_h et ζ_l, en omettant l'indice spatial :

$$h_t^0 = x_{2t+1} - (\alpha\, x_{2t} + \beta\, x_{2t+2}) \tag{3.5}$$

$$l_t^0 = x_{2t} + \gamma\, h_{t-1}^0 + \delta\, h_t^0 \tag{3.6}$$

$$h_t = \zeta_h\, h_t^0$$

$$l_t = \zeta_l\, l_t^0$$

Le filtrage temporel décrit ci-dessus correspond ainsi à une transformée sans mouvement, similaire à la transformée présentée dans la section 2.2.1. Comme vu précédemment, il est cependant plus efficace d'appliquer la transformée biorthogonale dans le sens du mouvement de la séquence vidéo. De plus, la formulation lifting d'une transformée en ondelettes permet d'introduire de façon très naturelle des opérateurs non-linéaires. On peut alors introduire simplement le mouvement en réalisant le filtrage temporel non pas sur les images, mais sur les images compensées en mouvement.

La prise en compte du mouvement nécessite l'introduction de champs de vecteurs. Le filtrage est fait par prédiction de l'image x_{2t+1} par rapport aux images x_{2t} et x_{2t+2}. Il est alors nécessaire d'utiliser deux champs de mouvement : un champ avant $\boldsymbol{v}^+_{2t+1}$, prédisant x_{2t+1} par rapport à x_{2t} et un champ arrière $\boldsymbol{v}^-_{2t+1}$, prédisant x_{2t+1} par rapport à x_{2t+2}. Comme dans le cas du filtre temporel de Haar selon Choi et Woods, décrit en section 2.2.3, les images d'approximation l_t sont synchrones avec les images paires x_{2t} et les images de détail h_t le sont avec les images impaires x_{2t+1}. Ces notations nous permettent alors de réécrire les équations (3.5) et (3.6) pour aboutir à la transformée temporelle 5/3 compensée en mouvement, illustrée par les Fig. 3.2 et 3.3 et décrite par :

$$h_t^0(\boldsymbol{n}) = x_{2t+1}(\boldsymbol{n}) - \Big[\alpha\, x_{2t}\big(\boldsymbol{n} - \boldsymbol{v}^+_{2t+1}(\boldsymbol{n})\big) + \beta\, x_{2t+2}\big(\boldsymbol{n} - \boldsymbol{v}^-_{2t+1}(\boldsymbol{n})\big)\Big] \tag{3.7}$$

$$l_t^0(\boldsymbol{m}) = x_{2t}(\boldsymbol{m}) + \gamma\, h^0_{t-1}\big(\boldsymbol{m} + \boldsymbol{v}^-_{2t-1}(\boldsymbol{p})\big) + \delta\, h^0_t\big(\boldsymbol{m} + \boldsymbol{v}^+_{2t+1}(\boldsymbol{q})\big) \tag{3.8}$$

$$h_t = \zeta_h\, h_t^0$$

$$l_t = \zeta_l\, l_t^0$$

L'utilisation de deux champs de mouvement pour prédire x_{2t+1} à partir de x_{2t} et x_{2t+2} rend immédiate la mise en œuvre de l'étape de prédiction (3.7). Par contre, l'équation de mise à jour (3.8) utilise les pixels $\boldsymbol{p}$ et $\boldsymbol{q}$ qui doivent être déterminés par retournement des champs de mouvement. Du fait de la non-inversibilité de ces derniers, les pixels $\boldsymbol{p}$ et $\boldsymbol{q}$ peuvent ne pas exister du tout ou être en nombre supérieur à un : leur détermination constitue un point non-trivial et est abordé en détail dans la section ci-dessous.

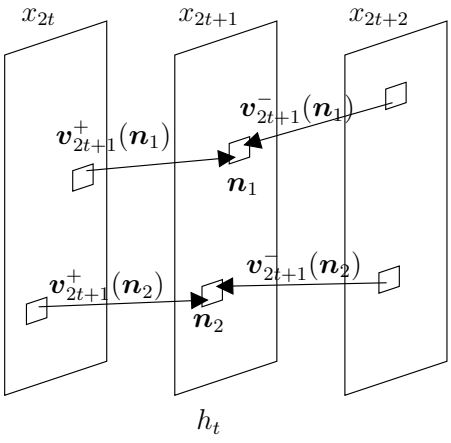


FIG. 3.2 – Opérateur de prédiction mis en jeu dans la transformée 5/3. Tous les pixels $\boldsymbol{n}_k$ sont connectés des deux côtés.

La formulation lifting permet d'introduire d'autres opérateurs que la compensation de mouvement comme l'estimation de mouvement subpixellique ou l'utilisation de chevau-

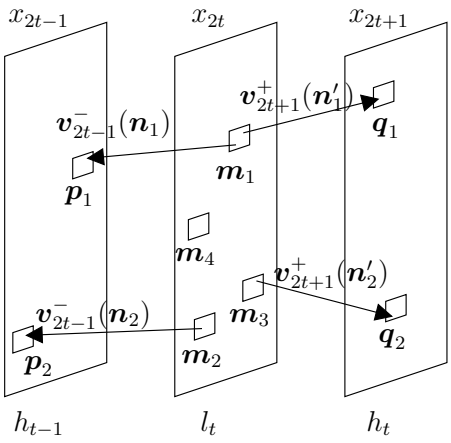


FIG. 3.3 – Opérateur de mise à jour utilisé dans la transformée 5/3. Le pixel $\boldsymbol{m}_1$ est connecté des deux côtés, les pixels $\boldsymbol{m}_2$ et $\boldsymbol{m}_3$ sont simplement connectés tandis que le pixel $\boldsymbol{m}_4$ n'est pas connecté.

chement (*overlap*) lors de la compensation de mouvement. Après avoir clarifié le comportement de l'opérateur de mise à jour et le choix des paramètres dans les sous-sections suivantes, nous verrons en fin de section une formulation définitive de la transformée temporelle 5/3 compensée en mouvement permettant d'introduire de tels opérateurs.

Notons que l'équation (3.7) n'a de sens que si le pixel $\boldsymbol{n}$ de l'image x_{2t+1} peut être prédit bidirectionnellement. En effet, en cas de coupure de scène ou d'occultation par un objet mouvant, une des prédictions n'a pas de sens et ne devra pas être utilisée. Elle risque de fausser la prédiction et de réduire ainsi l'efficacité globale de codage. La prise en compte de cette fonctionnalité reviendrait simplement à fixer α ou β à zéro. Cependant, sa mise en œuvre nécessiterait une détection préalable de coupure de scènes et probablement une segmentation homogène des images selon leur type de prédiction (bidirectionnelle, passée ou future). Nous ne détaillerons pas cette forme de prédiction adaptative, qui se rapproche fortement de la notion de *modes* abordée dans la section 2.2.5. De plus, il est probable que l'information concernant le type de prédiction nécessiterait d'être renseignée et transmise dans le flux vidéo compressé, pour que le décodeur puisse reconstruire la prédiction.

Opérateur de mise à jour

Du fait de la non-inversibilité des champs de mouvement, l'opérateur de mise à jour n'est pas complètement défini par l'équation (3.8). Il reste à déterminer les pixels $\boldsymbol{p}$ et $\boldsymbol{q}$ associés avec $\boldsymbol{m}$. Ceci est équivalent à résoudre les équations implicites $\boldsymbol{m} = \boldsymbol{p} - \boldsymbol{v}^{-}_{2t-1}(\boldsymbol{p}) = \boldsymbol{q} - \boldsymbol{v}^{+}_{2t+1}(\boldsymbol{q})$. Ces équations peuvent avoir aucune ou plusieurs solutions pour un $\boldsymbol{m}$ donné. Cette interprétation est fortement liée à l'existence de pixels non-connectés, connectés simplement et connectés de façon multiple comme décrit dans la section 2.2.2. Le cas présent est légèrement différent car il existe des pixels non-connectés et connectés provenant à la fois des directions avant et arrière.

En nous référant à la Fig. 3.3, nous notons $\mathrm{M}^{-} = \{\boldsymbol{p} \mid \boldsymbol{p} - \boldsymbol{v}^{-}_{2t-1}(\boldsymbol{p}) = \boldsymbol{m}\}$ l'ensemble des pixels connectés au pixel $\boldsymbol{m}$ dans l'image précédente et on dénote de la même façon l'ensemble des pixels connectés au pixel $\boldsymbol{m}$ dans l'image future par $\mathrm{M}^{+} = \{\boldsymbol{q} \mid \boldsymbol{q} -$

$\boldsymbol{v}^+_{2t+1}(\boldsymbol{q}) = \boldsymbol{m}\}$. Un pixel $\boldsymbol{m}$ est dit non-connecté dans le passé si Card $\mathrm{M}^- = 0$. Il est dit connecté simplement si Card $\mathrm{M}^- = 1$ et connecté de façon multiple si Card $\mathrm{M}^- > 1$. Ces mêmes notations s'utilisent pour décrire un état de connexion dans le futur avec M^+.

On distingue quatre cas représentant l'état de connexion d'un pixel $\boldsymbol{m}$:

1. $\mathrm{M}^- \neq \emptyset$ et $\mathrm{M}^+ \neq \emptyset$. Le pixel $\boldsymbol{m}$ est connecté bidirectionnellement sur l'image précédente et sur l'image future. L'existence des pixels $\boldsymbol{p}$ et $\boldsymbol{q}$ est donc garantie. C'est le cas le plus favorable dans la mesure où le pixel $\boldsymbol{m}$ sera filtré temporellement passe-bas par l'équation (3.8).
2. $\mathrm{M}^- = \emptyset$ et $\mathrm{M}^+ \neq \emptyset$. Le pixel $\boldsymbol{m}$ n'est ici connecté que sur l'image future et ne sera donc filtré que dans ce sens. L'expression $\boldsymbol{m} + \boldsymbol{v}^-_{2t-1}(\boldsymbol{p})$ n'a donc pas de sens dans l'équation (3.8) et impose $\gamma = 0$. Ce cas correspond par exemple à l'apparition d'un objet à l'instant $2t$ qui n'existait pas dans l'image x_{2t-1}. Ce cas peut aussi survenir lors d'un mauvais appariement de pixels, dû à un mouvement complexe, trop rapide ou à une variation brusque de la luminosité.
3. $\mathrm{M}^- \neq \emptyset$ et $\mathrm{M}^+ = \emptyset$. Le pixel $\boldsymbol{m}$ n'est ici connecté que sur l'image passé. Ce cas est analogue au cas précédent, en interchangeant $\boldsymbol{p}$ et $\boldsymbol{q}$. Ce cas impose $\delta = 0$ et apparaît par exemple lors de l'occlusion d'un objet par un autre ou en présence d'un mouvement complexe.
4. $\mathrm{M}^- = \emptyset$ et $\mathrm{M}^+ = \emptyset$. Le pixel $\boldsymbol{m}$ n'est connecté à aucun autre pixel. Il n'existe ainsi pas de pixel $\boldsymbol{p}$ et $\boldsymbol{q}$ vérifiant $\boldsymbol{p} - \boldsymbol{v}^-_{2t-1}(\boldsymbol{p}) = \boldsymbol{m}$ ou $\boldsymbol{q} - \boldsymbol{v}^+_{2t+1}(\boldsymbol{q}) = \boldsymbol{m}$, imposant donc $\gamma = \delta = 0$. Le pixel $\boldsymbol{m}$ ne subira donc pas de filtrage temporel passe-bas. Ce cas peut survenir si un objet apparaît furtivement sur une seule image de la séquence vidéo, comme par exemple des flashes lumineux, des objets parasites ou des feuilles tournoyantes comme dans la séquence *Tempête*. Comme dans les exemples précédents, ce cas est aussi susceptible d'apparaître en présence d'un mouvement complexe ou trop rapide.

Ces cas décrivent le comportement de l'opérateur de mise à jour dans les cas non-connectés et connectés. Prenons un pixel $\boldsymbol{m}$ vérifiant le cas 3, où $\mathrm{M}^- \neq \emptyset$ et $\mathrm{M}^+ = \emptyset$. L'opérateur de mise à jour compensé en mouvement s'écrit donc pour ce pixel :

$$l^0_t(\boldsymbol{m}) = x_{2t}(\boldsymbol{m}) + \gamma\, h^0_{t-1}\big(\boldsymbol{m} + \boldsymbol{v}^-_{2t-1}(\boldsymbol{p})\big)$$
$$\text{où } \boldsymbol{p} \in \mathrm{M}^-, \text{ càd } \boldsymbol{p} \text{ vérifie } \boldsymbol{p} - \boldsymbol{v}^-_{2t-1}(\boldsymbol{p}) = \boldsymbol{m}$$

Ce cas est similaire à celui d'un filtrage temporel de Haar. Cependant, une ambiguïté subsiste en cas de connexion multiple. En effet, si Card $\mathrm{M}^- > 1$ alors il existe *plusieurs* pixels $\boldsymbol{p}$ vérifiant $\boldsymbol{p} - \boldsymbol{v}^-_{2t-1}(\boldsymbol{p}) = \boldsymbol{m}$. Quelle solution adopter en cas de connexion multiple ? Dans le filtre temporel de Haar utilisé à l'origine dans le codec MC-EZBC, Choi et Woods [38] utilisent le premier pixel $\boldsymbol{p}$ rencontré dans le sens du balayage de l'écran. Divers critères basés sur la minimisation de la distorsion locale et sur l'uniformité locale du mouvement ont été proposés par Pesquet-Popescu [108] pour choisir le pixel $\boldsymbol{p}$ candidat. Cependant, Tillier a montré [145] qu'en cas de connexion multiple durant l'étape de mise à jour du filtre temporel de Haar, une stratégie optimale consiste à prendre la *moyenne* des pixels connectés au pixel $\boldsymbol{m}$, $\sum_{\boldsymbol{p} \in \mathrm{M}^-} \boldsymbol{p}/$ Card M^-. En utilisant un autre formalisme, Girod [52] a prouvé indépendamment un résultat analogue. Les auteurs montrent que cette solution conduit théoriquement et expérimentalement à une minimisation de l'erreur quadratique moyenne de reconstruction et donc à une augmentation du PSNR.

Le Tab. 3.1 présente quelques statistiques d'état de connexité des pixels lors de l'étape de mise à jour, où $\mathcal{M}^- =$ Card M^- et $\mathcal{M}^+ =$ Card M^+. Ces résultats ont été obtenus sur

une décomposition sur quatre niveaux de la séquence vidéo *Foreman* CIF sur les images d'indice compris entre 16 et 32. On remarque que le cas le plus fréquent est celui des pixels simplement connectés sur les images précédentes et futures, particulièrement dans premiers niveaux temporels. Cela témoigne de l'uniformité du mouvement à ces niveaux, due à la faible distance temporelle séparant ces images. On remarque de plus que les taux de pixels non-connectés et connectés de façon multiple augmentent dans les niveaux temporels supérieurs. La cause de cette augmentation est due à une probable inhomogénéité du mouvement à ces niveaux.

$\mathcal{M}^-\backslash\mathcal{M}^+$	0	1	>1
0	0.25	2.55	0.18
1	2.36	89.36	2.35
>1	0.17	2.54	0.18

Premier niveau temporel

$\mathcal{M}^-\backslash\mathcal{M}^+$	0	1	>1
0	0.97	5.76	0.88
1	4.43	76.63	4.45
>1	0.82	5.47	0.55

Second niveau temporel

$\mathcal{M}^-\backslash\mathcal{M}^+$	0	1	>1
0	4.16	14.30	3.14
1	5.28	50.71	5.70
>1	3.03	11.39	2.25

Troisième niveau temporel

$\mathcal{M}^-\backslash\mathcal{M}^+$	0	1	>1
0	9.83	15.80	5.75
1	8.06	34.14	6.88
>1	3.89	11.15	4.46

Quatrième niveau temporel

TAB. 3.1 – Pourcentages de pixels non-connectés ($\mathcal{M}=0$), simplement connectés ($\mathcal{M}=1$) et connectés de façon multiple ($\mathcal{M}>1$) dans la direction avant ($\mathcal{M}^+$) et arrière ($\mathcal{M}^-$) à plusieurs niveaux temporels lors de l'étape de mise à jour.

La présence et la juxtaposition de ces zones non-connectées et connectées est nuisible du point de vue de l'efficacité de codage. En effet, elle conduit à la création dans l'image d'approximation d'une mosaïque de régions hétérogènes qui seront filtrées ou pas, en fonction de leur état de connexité. Les changements abrupts à la frontière de ces régions induisent après transformation spatiale, une augmentation de l'amplitude des coefficients d'ondelettes. Ils contribuent ainsi à la réduction de l'efficacité du codage spatial des images d'approximation. De plus, ces discontinuités entre régions se propagent entre niveaux temporels et réduisent l'efficacité de la prédiction temporelle des niveaux suivants. Plusieurs approches ont été proposées afin de réduire ce phénomène. Hanke propose ainsi [57] un filtrage spatial passe-bas des frontières entre régions connectées et non-connectées et observe une baisse de la fluctuation du PNSR des images décodées. D'autres travaux [78, 133] préconisent une mise à jour adaptative par seuillage et pondération, basée sur des critères psychovisuels et des mesures locales d'activité. Nous proposons cependant dans la section 4.2 une transformée temporelle permettant de s'affranchir de ce problème et offrant une efficacité de codage vidéo supérieure à la transformée temporelle 5/3.

Choix des paramètres

Muni des équations (3.7) et (3.8) et des différents cas pouvant apparaître durant la mise à jour, on peut alors déterminer les valeurs des paramètres α, β, γ, δ, ζ_h et ζ_l utilisés lors de la transformation. Nous sommes donc en présence d'un filtre de prédiction et de mise à jour qui varient tous deux dans les directions spatiales et temporelles. Une analyse rigoureuse est ainsi délicate à mener. Pour simplifier les choses, considérons par exemple

une région immobile de la séquence vidéo. Bien que le mouvement local soit nul, on peut observer des variations d'intensité lumineuse dues au bruit, à la variation de l'illumination, etc... De telles régions peuvent constituer une partie importante des images d'une séquence vidéo comme le fond ou les objets statiques, justifiant ainsi notre hypothèse simplificatrice. Dans ces régions sans mouvements, les filtres sont temporellement invariants et ceci rend alors possible l'utilisation d'outils classiques en traitement du signal comme la transformée en Z.

Le choix des coefficients α et β est fait de façon à satisfaire le comportement passe-haut du filtre de prédiction. On impose un nombre de moments nuls égal à un, exigeant ainsi qu'en présence d'un signal constant les coefficients de détail résultants soient nuls. Ceci impose de l'équation (3.5) que $\alpha + \beta = 1$. De plus, dans notre hypothèse d'invariance temporelle des filtres, il n'y aucune raison de privilégier une prédiction basée sur l'échantillon passé x_{2t} ou sur le futur x_{2t+2}. Ceci impose alors $\alpha = \beta = 1/2$.

De même, la détermination des coefficients γ et δ doit permettre à l'opérateur de mise à jour d'assurer un comportement passe-bas. La transformée en Z associée aux coefficients d'approximation l_t vaut $L(z) = -\alpha\gamma z^{-2} + \gamma z^{-1} + (1 - \beta\gamma - \alpha\delta) + \delta z - \beta\delta z^2$. Le caractère passe-bas étant assuré par $L(-1) = 0$, on obtient alors $\gamma + \delta = 1/2$. Comme précédemment, la non-prédilection pour une direction passée ou future impose au final : $\gamma = \delta = 1/4$.

Les constantes multiplicatives ζ_h et ζ_l ont été choisies par similarité avec la transformée biorthogonale 5/3, décrite dans la section 1.2.3. On impose ainsi $\zeta_h = 1/\sqrt{2}$ et $\zeta_l = \sqrt{2}$. Ce choix permet d'assurer la quasi-orthonormalité de la transformée. Cependant, la transformée 5/3 est loin d'être orthogonale et ceci conduit à une efficacité de codage sous-optimale lors de l'utilisation d'algorithmes d'allocation optimale de débit. Les travaux d'Usevitch [154] préconisent l'utilisation d'une pondération adaptée lors du calcul de l'erreur de quantification, permettant ainsi une minimisation de l'erreur quadratique moyenne. En suivant cette approche dans une structure lifting, Rusert et Ohm proposent [57, 120] de modifier les coefficients ζ_h et ζ_l de façon à normaliser les énergies des filtres de synthèse $\widetilde{h}_0$ et $\widetilde{h}_1$. Ils montrent que cette modification permet la minimisation de l'erreur moyenne de reconstruction, sans toutefois avoir recours à une pondération externe. Dans le cas du filtre 5/3, ils obtiennent $\zeta_h = \sqrt{23/32}$, $\zeta_l = \sqrt{3/2}$ et observent une amélioration légère de l'efficacité de codage.

D'autres approches existent pour le choix de ζ_h et ζ_l. Dans le but de minimiser les fluctuations du PSNR des images décodées après filtrage temporel de 5/3, Tillier [142] montre sous une hypothèse haute résolution que le choix optimal consiste à prendre les valeurs $\zeta_h = \sqrt{30/19}$ et $\zeta_l = \sqrt{32/19}$. Des simulations expérimentales confirment ces résultats et produisent des courbes de PSNR plates, moyennant toutefois une perte légère du PSNR *moyen* calculé sur l'ensemble des images décodées.

L'approche retenue permet aussi de déterminer ces coefficients dans certains cas où les filtres ne sont pas invariants temporellement. Nous avons ainsi évoqué dans la section précédente des cas où l'opérateur de mise à jour traite des pixels qui ne sont pas connectés bidirectionnellement. Ces cas spéciaux où $\mathrm{M}^- = \emptyset$ ou $\mathrm{M}^+ = \emptyset$ influent sur le calcul de γ et δ. Par exemple, dans le cas 3 où $\mathrm{M}^- \neq \emptyset$ et $\mathrm{M}^+ = \emptyset$, il n'existe pas de pixel futur et donc $\delta = 0$. Comme on a toujours $\gamma + \delta = 1/2$, alors $\gamma = 1/2$ et les valeurs de α, β, ζ_h et ζ_l restent inchangées pour ce cas précis.

Filtrage temporel 5/3 compensé en mouvement : formulation finale

Comme vu dans les sections précédentes, la construction d'un filtre temporel 5/3 compensé en mouvement est loin d'être unique. Il existe ainsi plusieurs possibilités concernant le choix de la méthode d'estimation de mouvement, la façon de gérer les pixels non-connectés et connectés de façon multiple durant la mise à jour, les valeurs des coefficients de mise à l'échelle ζ_h et ζ_l... Ces choix relèvent souvent du résultat d'un compromis entre efficacité de codage et complexité. Dans la suite de nos travaux, nous avons ainsi adopté le filtre temporel 5/3 suivant, qui peut s'exprimer sous forme lifting par :

$$h_t^0(\boldsymbol{n}) = x_{2t+1}(\boldsymbol{n}) - \frac{1}{2}\big(\mathcal{C}(x_{2t}, \boldsymbol{v}_{2t+1}^{+})(\boldsymbol{n}) + \mathcal{C}(x_{2t+2}, \boldsymbol{v}_{2t+1}^{-})(\boldsymbol{n})\big) \tag{3.9}$$

$$l_t^0(\boldsymbol{n}) = x_{2t}(\boldsymbol{n}) + \gamma\,\mathcal{C}^{-1}(h_{t-1}, \boldsymbol{v}_{2t-1}^{-})(\boldsymbol{n}) + \delta\,\mathcal{C}^{-1}(h_t, \boldsymbol{v}_{2t+1}^{+})(\boldsymbol{n}) \tag{3.10}$$

$$h_t(\boldsymbol{n}) = 1/\sqrt{2}\; h_t^0(\boldsymbol{n}) \tag{3.11}$$

$$l_t(\boldsymbol{n}) = \sqrt{2}\; l_t^0(\boldsymbol{n}) \tag{3.12}$$

$$\text{avec} \begin{cases} \gamma = \delta = 1/4 & \text{si } \boldsymbol{n} \text{ est connecté des deux côtés} \\ \gamma = 1/2 \text{ et } \delta = 0 & \text{si } \boldsymbol{n} \text{ est connecté seulement à gauche} \\ \gamma = 0 \text{ et } \delta = 1/2 & \text{si } \boldsymbol{n} \text{ est connecté seulement à droite} \\ \gamma = 0 \text{ et } \delta = 0 & \text{si } \boldsymbol{n} \text{ n'est pas connecté} \end{cases}$$

où l'opérateur de compensation de mouvement spatial $\mathcal{C}$ agit sur une image x et un champ de mouvement muet $\boldsymbol{v}$. Il est défini par $\mathcal{C}(x, \boldsymbol{v})(\boldsymbol{n}) = x\big(\boldsymbol{n} - \boldsymbol{v}(\boldsymbol{n})\big)$ et met en jeu une compensation subpixellique avec recouvrement. L'opérateur de compensation inverse $\mathcal{C}^{-1}$ utilise la stratégie de mise à jour moyenne décrite en [145] et est défini par :

$$\mathcal{C}^{-1}(x, \boldsymbol{v})(\boldsymbol{m}) = \begin{cases} x\Big(\sum_{\boldsymbol{p}\in \mathrm{M}} x(\boldsymbol{p}) / \text{ Card M}\Big) & \text{où M} = \{\boldsymbol{p} \mid \boldsymbol{p} - \boldsymbol{v}(\boldsymbol{p}) = \boldsymbol{m}\} \\ 0 & \text{si Card M} = 0 \end{cases}$$

La décomposition successive des sous-bandes d'approximation nous permet d'obtenir une analyse temporelle 5/3 d'un groupe d'images d'une séquence vidéo. La Fig. 3.4 illustre une telle décomposition, effectuée sur une suite de 8 images et sur 3 niveaux temporels. On notera tout particulièrement les flèches en pointillés, décrivant l'utilisation d'images qui ne sont pas situées dans le groupe d'image courant. Leur présence s'explique par la taille du support du filtre 5/3 et pose un réel problème lors de l'implémentation de la transformée temporelle 5/3. La section suivante est consacrée à l'étude de ce problème et propose une solution pour y remédier.

Nous avons de plus illustré sur la Fig. 3.5 les sous-bandes $\{l_{k,j}\}$ et $\{h_{k,j}\}$ issues de la décomposition temporelle d'un extrait de la séquence *Foreman* sur 3 niveaux. Cette décomposition peut-être mise en correspondance avec la Fig. 3.4. Cependant, les sous-bandes ont été réorganisées à chaque niveau temporel en plaçant d'abord les sous-bandes d'approximation, suivies par les sous-bandes de détail. Cette disposition permet ainsi d'observer la décomposition successive des sous-bandes d'approximation à chaque niveau temporel. On remarquera la nature très différente des sous-bandes en fonction de leur type et de leur profondeur temporelle.

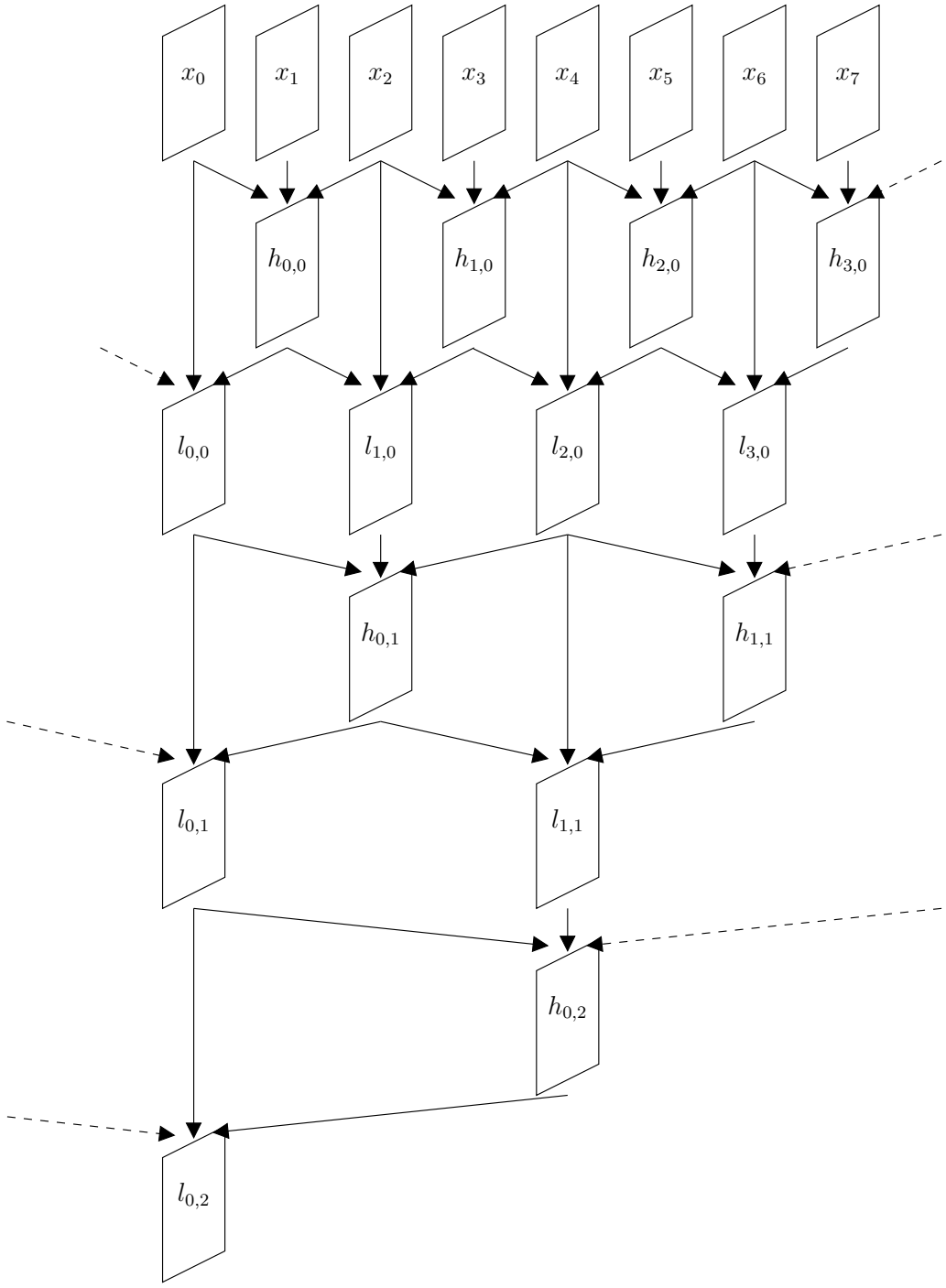


FIG. 3.4 – Analyse multirésolution temporelle 5/3 sur 3 niveaux d'une séquence vidéo.

3.1.4 Traitement au fil de l'eau

L'implémentation effective d'une transformée en ondelettes nécessite classiquement la connaissance préalable de la totalité du signal. Cette approche est valable dans le cas de signaux mono-dimensionnels de taille faible. Cependant, pour des raisons de place mémoire et de latence, elle devient discutable dans le cas d'images et irréalisable dans le

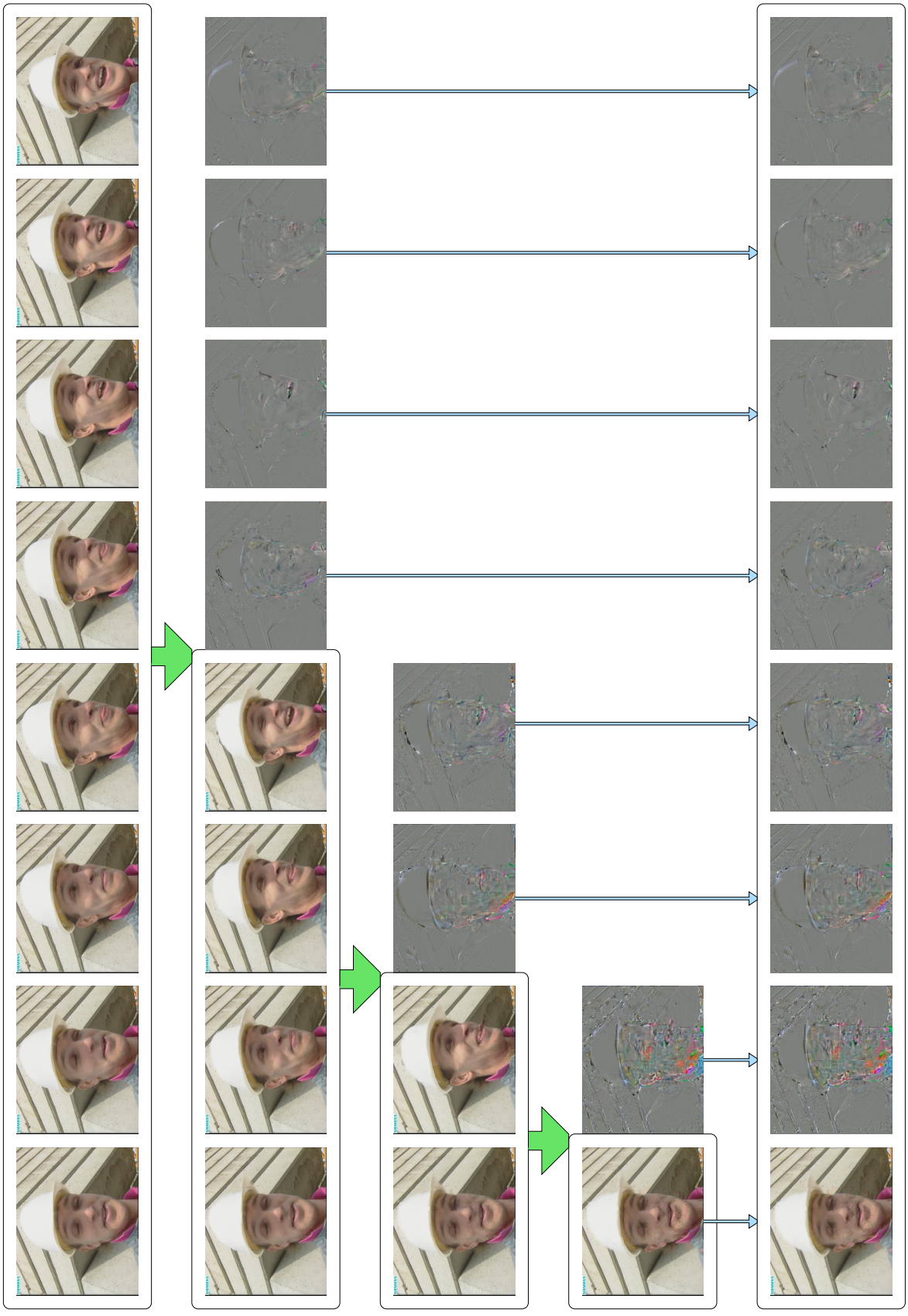


FIG. 3.5 – Analyse multirésolution temporelle 5/3 sur 3 niveaux d'un extrait de la séquence *Foreman*.

cas de séquences vidéo. Il n'est en effet pas réaliste de précharger les centaines d'images que composent une telle séquence avant de pouvoir la transformer. Il est alors légitime de se poser la question suivante : est-il possible d'implémenter une transformée de telle manière qu'elle décompose à la volée et au fur et à mesure les échantillons qu'elle reçoit ?

Le cas de la transformée temporelle de Haar est particulier. En effet, comme illustré précédemment dans la Fig. 2.6, la transformée de Haar opère localement sur les images et la transformation d'un GOP courant ne nécessite pas la connaissance d'images en dehors de ce GOP. Il est alors simple d'effectuer l'analyse temporelle d'un GOP sur place en appliquant juste les équations régissant la transformée de Haar.

Cependant, le cas de la transformée 5/3 est plus problématique. En observant l'analyse 5/3 illustrée en Fig. 3.4, on remarque que le calcul de la sous-bande $h_{3,0}$, synchronisée avec l'image x_7, nécessite la connaissance de l'image x_8. De même, la sous-bande $h_{1,1}$ nécessite l'image x_{10}, etc... En poursuivant, on remarque que le calcul de la décomposition temporelle 5/3 d'un GOP courant nécessite un contexte arrière et avant important. Il est possible d'implémenter la transformée temporelle 5/3 de cette façon mais cela nécessite un algorithme complexe et une grande quantité de mémoire.

Plusieurs solutions dans le cas 2D [39] et 3D [93] ont été apportées pour pouvoir effectuer la transformation en ondelettes à la volée, traitant ainsi au fil de l'eau les images ou échantillons entrants. En se basant sur la décomposition en banc de filtres de la transformée, ces auteurs préconisent l'utilisation d'un tampon (*buffer*) cyclique de filtrage où les échantillons sont consommés, filtrés et d'où l'on extrait les coefficients transformés. Le traitement est donc effectué à la volée et la seule mémoire requise est celle du tampon cyclique de filtrage.

Cependant, la formulation lifting de la transformée en ondelettes permet une implémentation encore plus compacte, nécessitant des tampons plus petits que ceux utilisés dans une décomposition en bancs de filtres. Cette approche lifting est utilisée dans notre schéma de codage et a également été employée par la suite dans le codeur Vidwav [164].

Il est de plus possible d'utiliser une structure modulaire pour réaliser une analyse sur plusieurs niveaux, où chaque module réalise la transformation élémentaire de deux échantillons en deux coefficients d'ondelettes. Dans une approche consommateur/producteur, il est ainsi possible de chaîner plusieurs modules afin de réaliser une décomposition sur plusieurs niveaux. Chaque module consomme alors des échantillons (ou des coefficients d'approximation) provenant du module précédent et produit des coefficients d'ondelettes qu'il transmet au module suivant.

Le fonctionnement d'un tel module est illustré dans le cas de la transformée 5/3 par la Fig. 3.6 où l'on observe l'évolution chronologique du buffer cyclique de filtrage au cours de la transformée. Le module effectue alors un cycle en accomplissant les étapes décrites dans l'algorithme suivant.

Algorithme de traitement au fil de l'eau

Initialisation Un buffer cyclique FIFO d'une taille fixe de n images est alloué, où n dépend de la largeur du support des opérateurs de prédiction et de mise à jour. Dans le cas de la transformée 5/3, $n = 4$. On ajoute alors deux premières images dans le buffer.

Ajout de deux images On ajoute les deux images suivantes du flux à transformer dans le buffer. Si cela n'est pas possible, en fin de flux par exemple, alors le module est placé dans un état *fin de flux*.

Prédiction Si le module n'est pas dans l'état *fin de flux* alors l'opérateur de prédiction est appliqué tel que donné par la décomposition lifting de la transformée. Sinon l'opérateur est appliqué après repliement sur les bords par symétrisation.

Mise à jour L'opérateur de mise à jour est appliqué de façon similaire à celui de prédiction, en tenant compte de l'état *fin de flux*. Selon la structure lifting de la transformée, il peut y avoir d'autres étapes de prédiction ou de mise à jour.

Extraction et mise à l'échelle Les deux premières images sont extraites et retirées du buffer, ce sont les images d'approximation et de détail résultant de la transformation. Le buffer est alors décalé de deux images. Les images transformées sont alors mises à l'échelle puis transmises au module suivant, qui peut être une autre instance du même module dans le cas d'une analyse sur plusieurs niveaux.

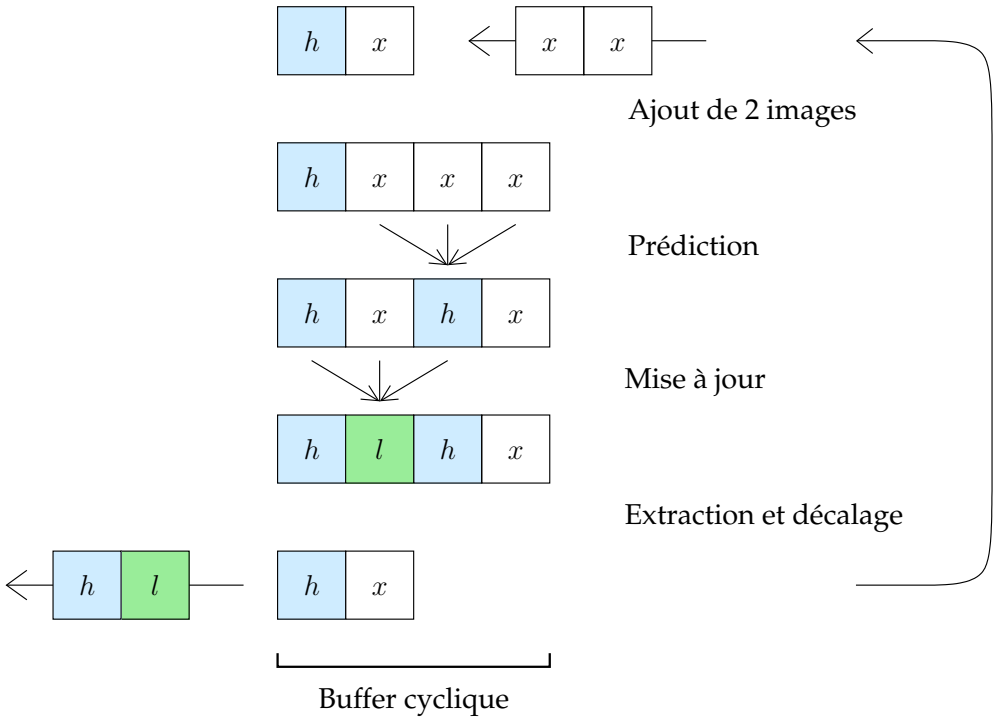


FIG. 3.6 – Schéma de fonctionnement du module de traitement au fil de l'eau de la transformée temporelle 5/3. On y observe l'évolution du buffer cyclique de filtrage.

3.2 Résultats expérimentaux

Cette section présente les résultats de codage expérimentaux observés lors de la mise en œuvre de la transformée temporelle 5/3 compensée en mouvement au sein du codec MC-EZBC, décrit en section 2.2.4. Des mesures d'efficacité objectives sont rapportées et les résultats sont comparés avec le filtre temporel de Haar selon Choi et Woods, rappelé en section 2.2.3. D'autres comparaisons avec des codecs vidéos couramment utilisés (MPEG-2, MPEG-4 Part. 2 et Windows Media 9) ont aussi été ajoutées.

3.2.1 Efficacité de codage

Dans un contexte de codage avec perte, compresser signifie baisser la quantité d'information ou le débit nécessaire à la description d'une vidéo d'une qualité donnée. Par conséquent, cela signifie aussi augmenter la qualité d'une description vidéo pour un débit donné. Il n'est cependant pas simple de formaliser ce concept. Ainsi, la définition de la notion de qualité pour une image ou pour une séquence vidéo possède un aspect psychovisuel et sensoriel difficile à modéliser, qui sort totalement du cadre de cette thèse. Nous utiliserons la moyenne du PSNR (*Peak Signal Noise Ratio*) ou rapport signal à bruit de crête calculée sur la composante de luminance Y des images décodées comme mesure objective de la qualité d'une vidéo décompressée. C'est une mesure simple à manipuler et qui correspond bien à la réalité visuelle perçue.

Les simulations ont été conduites sur les séquences couleur *Mobile* et *Foreman*, en utilisant le codec MC-EZBC avec le filtre temporel de Haar selon Choi et Woods, rappelé en section 2.2.3 et avec le filtre temporel 5/3 compensé en mouvement décrit dans ce chapitre. La décomposition temporelle des images a été faite sur 5 niveaux et le mouvement a été estimé au $1/8$-ème de pixel près.

Les résultats sur les codecs MPEG-2 et MPEG-4 ont été obtenus au moyen du logiciel Ffmpeg [20] en utilisant des paramètres de codage optimaux : utilisation d'une taille de GOP de 32 images, contrôle de débit précis assuré par deux passes d'encodage, estimation de mouvement au quart de pixel pour le codec MPEG-4. Les résultats obtenus avec les codecs Windows Media 9 proviennent de [44]. Les schémas de codage hybride présentés n'étant pas scalables, chaque simulation a nécessité un encodage et un décodage complet pour chaque débit tandis que les résultats obtenus avec le codec MC-EZBC n'ont nécessité qu'un seul encodage par séquence et par filtre.

Les Tab. 3.2 et 3.3 présentent les mesures de Y-PSNR obtenues pour plusieurs débits globaux, en utilisant le codec MC-EZBC munis des filtres temporels de Haar et 5/3, et en utilisant les codecs vidéo hybride MPEG-2, MPEG-4 Part. 2 et Windows Media 9.

Y-PSNR (in dB)	512 kbs	768 kbs	1024 kbs	1536 kbs	2048 kbs
Haar	**27.27**	29.78	31.35	33.61	35.30
5/3	26.53	**30.39**	**32.27**	**34.60**	**36.33**
MPEG-2	24.57	26.10	27.43	28.40	30.35
MPEG-4 Part. 2	25.75	28.50	29.65	31.21	33.07
Windows Media 9	25.6	26.5	27.1	28.0	28.5

TAB. 3.2 – Comparaison de l'efficacité de codage du codec MC-EZBC muni de différents filtres temporels et de plusieurs codecs hybrides sur la séquence *Mobile* CIF 30 Hz.

On observe tout d'abord que la transformée 5/3 donne les meilleurs résultats globaux sur les deux séquences et à tous les débits, à l'exception du débit 512 kbs. La transformée 5/3 surpasse ainsi la transformée de Haar sur les moyens et hauts débits : ceci peut s'expliquer par une meilleure prédiction temporelle, due à une estimation de mouvement bidirectionnelle. La transformée de Haar semble cependant plus efficace à bas débit car elle ne nécessite qu'un seul champ de mouvement, contrairement à la transformée 5/3 qui en nécessite deux. Comme ces champs sont incompressibles, ils ont tendance à prendre une place trop importante dans les bas débits.

On notera aussi la supériorité du schéma de codage $t + 2D$ par rapport aux schémas de codage hybride MPEG-2, MPEG-4 et WM-9, pourtant non-scalables. Le codec MPEG-4

Y-PSNR (in dB)	512 kbs	768 kbs	1024 kbs	1536 kbs	2048 kbs
Haar	34.47	36.17	37.53	39.39	40.89
5/3	34.27	**36.57**	**38.08**	**39.95**	**41.43**
MPEG-2	32.70	33.71	35.85	37.81	39.78
MPEG-4 Part. 2	33.60	34.95	36.59	38.94	40.55
Windows Media 9	**34.5**	36.0	36.7	38.1	38.7

TAB. 3.3 – Comparaison de l'efficacité de codage du codec MC-EZBC muni de différents filtres temporels et de plusieurs codecs hybrides sur la séquence *Foreman* CIF 30 Hz.

semble être le schéma le plus efficace des codeurs hybrides mais il reste nettement en deçà de la transformée 5/3, accusant un baisse atteignant jusqu'à 3 dB sur la séquence *Mobile* à 2048 kbs. Comme précédemment, le codec WM-9 est cependant meilleur à 512 kbs mais n'est pas scalable. Ces résultats montrent cependant globalement la bonne efficacité du schéma de codage $t + 2D$ muni de la transformée 5/3 comparé au codec MPEG-4.

3.2.2 Scalabilité temporelle

Afin d'apprécier l'impact que possède une transformation sur la scalabilité temporelle, nous avons comparé les images d'approximation obtenues avec une transformée temporelle de Haar et une transformée 5/3. La Fig. 3.7 illustre les images obtenues sur la séquence *Tempête* après une analyse temporelle sur quatre niveaux. On observe clairement le flou créé introduit par le filtre de Haar. Au contraire, l'image d'approximation obtenue par la transformée temporelle 5/3 possède une netteté et un piqué supérieur. Ceci peut s'expliquer par le fait que la transformée de Haar n'utilise qu'une seule image pour effectuer sa prédiction temporelle, contrairement au filtre 5/3 qui opère une prédiction bidirectionnelle.

FIG. 3.7 – Zooms sur une région d'une image d'approximation du quatrième niveau temporel obtenu avec une transformée temporelle de Haar (gauche) et une transformée 5/3 (droite) sur la séquence *Tempête* au format CIF.

3.3 Conclusion

Dans le cadre du schéma de codage vidéo $t + 2D$, l'utilisation du schéma lifting temporel nous a permis de construire une transformée temporelle 5/3 compensée en mouvement, mettant en jeu des opérateurs non-linéaires de compensation de mouvement avec chevauchement et d'interpolation subpixellique. C'est une transformée à reconstruction parfaite dont les opérateurs de prédiction et de mise à jour bidirectionnels ont été choisis pour maximiser son efficacité de codage.

Après sa mise en place au sein du codec MC-EZBC, la transformée temporelle 5/3 a permis d'atteindre des gains en PSNR atteignant 1 dB par rapport à la transformée de Haar. De plus, elle possède une efficacité de codage nettement supérieure à celle offerte par les codecs vidéo hybrides MPEG-2, MPEG-4 et Windows Media 9, pourtant non-scalables.

Bien que satisfaisante, l'efficacité de codage de la transformée temporelle 5/3 peut cependant être encore améliorée. Plusieurs pistes s'offrent à nous : en effet, lors de sa construction, diverses questions ont été soulevées sur le choix des champs de mouvement, sur la présence de zones non-connectées lors de l'étape de mise à jour ou sur le fait que certaines zones n'ont pas d'intérêt à bénéficier d'une prédiction bidirectionnelle. Ces pistes constituent des axes de recherches intéressants qui sont développés dans le chapitre suivant, consacré à l'optimisation de la transformée temporelle.

Chapitre 4

Optimisation du filtrage temporel

Tout au long de ce chapitre, nous présentons différentes stratégies d'optimisation de la transformée temporelle mise en jeu dans le schéma de codage vidéo $t + 2D$. Nous visons tout d'abord l'amélioration de l'efficacité objective du codeur vidéo. En se basant sur le filtre temporel $5/3$ décrit dans le chapitre précédent, nous présentons tout d'abord dans la section 4.1 un algorithme quasi-optimal de choix des champs de mouvement mis en jeu dans cette transformée. La mise en place de cet algorithme conduit alors à des résultats expérimentaux qui montrent un gain significatif par rapport à la stratégie adoptée dans la section précédente, validant ainsi l'approche retenue.

L'amélioration subjective de la qualité de codage est aussi une priorité dans la construction d'une transformée temporelle. En particulier, nous rapportons dans la section 4.2 la présence d'artefacts fantômes dans les séquences vidéo décodées à bas débit, rappelant des zones ou objets précédemment observés. La présence de ces artefacts est fortuite et est mal traduite par une mesure objective de la qualité comme le PSNR. Après avoir décrit et analysé les raisons de la présence de ces artefacts, nous présentons alors un nouveau filtre temporel, basé sur la transformée $5/3$ et construit dans l'optique de ne pas générer de tels artefacts. Les résultats expérimentaux observés après la mise en place de cette transformée sont visuellement convaincants. Nous montrons ensuite comment l'algorithme précédent de choix optimal des champs de mouvement peut être appliqué dans le cas de cette transformée. Nous observons alors un gain supplémentaire de l'efficacité de codage objective en terme de PSNR.

Les transformées temporelles classiquement utilisées dans les schémas $t + 2D$ possèdent un inconvénient qui n'a pas encore été mentionné : elles introduisent un retard non négligeable à l'encodage et au décodage des séquences visuelles. Cette latence est souvent trop importante pour permettre leur utilisation dans des applications en temps réel comme la vidéoconférence ou la vidéosurveillance. Nous présentons dans la section 4.3 une étude détaillée sur les retards introduits par différents filtres temporels et sur leurs causes. Nous proposons alors une stratégie de modification générale de la transformée temporelle, permettant de modérer voire d'annuler les retards qu'elle introduit et conduisant seulement à des pertes minimes en terme de débit-distorsion. Un exemple est donné dans le cas de la transformée $5/3$ et est illustré par des simulations expérimentales. Les résultats sont convaincants et concluent une solution offrant un large éventail de compromis entre délai et efficacité de codage, en fonction des besoins de l'application.

Nous avons pour l'instant seulement envisagé des transformées temporelles issues ou dérivées de l'ondelette de Haar et de l'ondelette $5/3$, de supports relativement courts. Au vu de l'amélioration constatée au chapitre précédent lors du passage du filtre de Haar au filtre $5/3$, on peut s'interroger sur le bénéfice apporté par un filtre basé sur une autre ondelette à support plus long. La construction d'une transformée temporelle avec une prédiction à plus long terme laisse ainsi entrevoir une meilleure décorrélation temporelle

des images. A cette fin, nous présentons dans la section 4.4 un filtre temporel basé sur l'ondelette de Daubechies-4. Ses performances ne sont cependant pas à la hauteur de son originalité et après avoir montré quelques résultats expérimentaux, nous expliquons les raisons de ses performances modestes.

4.1 Optimisation des vecteurs impliqués dans la prédiction

Les sous-bandes temporelles de détail constituent la majeure partie du flux binaire et une façon simple d'améliorer l'efficacité globale du codeur vidéo est de diminuer la complexité de ces images. On cherche ici à améliorer l'opérateur de prédiction mis en jeu dans la transformée temporelle 5/3 en optimisant les champs de vecteurs utilisés. La stratégie proposée dans cette section a conduit à la publication d'un article de conférence [105], repris dans un article de revue [106] plus général sur l'utilisation du schéma lifting compensé en mouvement en codage vidéo scalable.

4.1.1 Présentation du problème

On rappelle la transformée temporelle 5/3 utilisée par le schéma de codage présenté dans la section 3.1.3 qui s'exprime sous la forme lifting suivante :

$$h_t^0(\boldsymbol{n}) = x_{2t+1}(\boldsymbol{n}) - \frac{1}{2}\big(\mathcal{C}(x_{2t}, \boldsymbol{v}_{2t+1}^+)(\boldsymbol{n}) + \mathcal{C}(x_{2t+2}, \boldsymbol{v}_{2t+1}^-)(\boldsymbol{n})\big) \tag{4.1}$$

$$l_t^0(\boldsymbol{n}) = x_{2t}(\boldsymbol{n}) + \gamma\,\mathcal{C}^{-1}(h_{t-1}, \boldsymbol{v}_{2t-1}^-)(\boldsymbol{n}) + \delta\,\mathcal{C}^{-1}(h_t, \boldsymbol{v}_{2t+1}^+)(\boldsymbol{n}) \tag{4.2}$$

$$h_t(\boldsymbol{n}) = 1/\sqrt{2}\; h_t^0(\boldsymbol{n}) \tag{4.3}$$

$$l_t(\boldsymbol{n}) = \sqrt{2}\; l_t^0(\boldsymbol{n}) \tag{4.4}$$

$$\text{avec} \begin{cases} \gamma = \delta = 1/4 & \text{si } \boldsymbol{n} \text{ est connecté des deux côtés} \\ \gamma = 1/2 \text{ et } \delta = 0 & \text{si } \boldsymbol{n} \text{ est connecté seulement à gauche} \\ \gamma = 0 \text{ et } \delta = 1/2 & \text{si } \boldsymbol{n} \text{ est connecté seulement à droite} \\ \gamma = 0 \text{ et } \delta = 0 & \text{si } \boldsymbol{n} \text{ n'est pas connecté} \end{cases}$$

Nous nous intéressons ici uniquement à l'optimisation de la prédiction dans le but de diminuer la complexité des images de détail h_t. Pour simplifier les notations, nous omettons le coefficient de normalisation $1/\sqrt{2}$ dans nos raisonnements. En développant l'opérateur de compensation de mouvement $\mathcal{C}$, on peut alors réécrire l'équation (4.1) et détailler les coefficients des sous-bandes de détail :

$$h_t(\boldsymbol{n}) = x_{2t+1}(\boldsymbol{n}) - \frac{1}{2}\Big(x_{2t}\big(\boldsymbol{n} - \boldsymbol{v}_{2t+1}^+(\boldsymbol{n})\big) + x_{2t+2}\big(\boldsymbol{n} - \boldsymbol{v}_{2t+1}^-(\boldsymbol{n})\big)\Big) \tag{4.5}$$

On considère ici le problème de l'estimation bidirectionnelle des champs de mouvement avant $\boldsymbol{v}_{2t+1}^+$ et arrière $\boldsymbol{v}_{2t+1}^-$ impliqués dans la prédiction, de manière à minimiser la distorsion des images de détail h_t. Sous la réserve d'un choix judicieux d'une mesure de distorsion, on espère ainsi minimiser le coût de codage des images h_t. Il est par exemple raisonnable de penser que le choix de la norme ℓ_2 et donc la minimisation de l'énergie des images de détail h_t conduise à une réduction de leur coût de codage. On notera que cette approche a été poursuivie ultérieurement par Cagnazzo [27] dans un cas plus simple.

Compte tenu de la structure en blocs des champs de mouvement $\boldsymbol{v}_{2t+1}^+$ et $\boldsymbol{v}_{2t+1}^-$ due à la méthode choisie pour leur estimation, on choisit de minimiser la distorsion des images

de détail h_t, bloc par bloc. En nous concentrant sur la minimisation d'un bloc $\mathcal{B}$ courant appartenant à l'image x_{2t+1}, nous choisissons d'omettre l'indice spatial $\boldsymbol{n}$ pour alléger les notations et écrivons alors $\boldsymbol{v}^+ = \boldsymbol{v}^+_{2t+1}(\boldsymbol{n})$ et $\boldsymbol{v}^- = \boldsymbol{v}^-_{2t+1}(\boldsymbol{n})$. La minimisation de la distorsion des images de détail h_t revient ainsi à un problème de recherche d'optimum à deux paramètres. Elle peut se faire sous une contrainte de débit liée au coût des vecteurs $\lambda\big(R(\boldsymbol{v}^+) + R(\boldsymbol{v}^-)\big)$ ou non (en prenant $\lambda = 0$), et conduit à la minimisation du critère J général suivant :

$$J(\boldsymbol{v}^+, \boldsymbol{v}^-) = \sum_{\boldsymbol{n}\in\mathcal{B}} d\big(h_t(\boldsymbol{n})\big) + \lambda R(\boldsymbol{v}^+) + \lambda R(\boldsymbol{v}^-) \tag{4.6}$$

où $\mathcal{B}$ est un bloc de l'image courante x_{2t+1} à prédire, d une mesure de distorsion usuelle (erreur absolue ℓ_1, norme quadratique ℓ_2 , etc...) et R le coût de codage d'un vecteur. En minimisant la distorsion de tous les blocs des images de détail h_t comme illustré sur la Fig. 4.1, on espère ainsi minimiser leur complexité et donc faciliter leur codage.

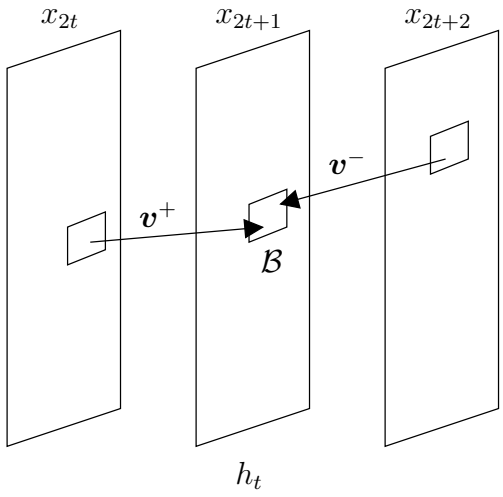


FIG. 4.1 – Opérateur de prédiction mis en jeu dans la transformée 5/3 et minimisation de la distorsion du bloc $\mathcal{B}$.

Estimation indépendante de $\boldsymbol{v}^+$ et $\boldsymbol{v}^-$

Dans la transformée temporelle 5/3 proposée dans la section 3.1 du chapitre précédent, les champs de mouvement avant $\boldsymbol{v}^+$ et arrière $\boldsymbol{v}^-$ sont estimés de façon indépendante. Le champ de mouvement avant $\boldsymbol{v}^+$ est ainsi calculé par la procédure d'appariement de blocs (*block-matching*) HVSBM, décrite en section 2.2.4, où chaque bloc de l'image courante x_{2t+1} est mis en correspondance avec un bloc de l'image de référence x_{2t}, de façon à minimiser le coût $D + \lambda R$ où D est une mesure de distorsion usuelle : la SAD (*Sum of Absolute Differences*). Le champ de mouvement arrière $\boldsymbol{v}^-$ est calculé de la même façon et indépendamment de $\boldsymbol{v}^+$ mais en prenant x_{2t+2} comme image de référence. Les vecteurs mouvements $\boldsymbol{v}^+$ et $\boldsymbol{v}^-$ vérifient donc :

$$\boldsymbol{v}^+ = \arg\min_{\boldsymbol{v}} \sum_{\boldsymbol{n}\in\mathcal{B}} \Big[d\big(x_{2t+1}(\boldsymbol{n}) - x_{2t}(\boldsymbol{n} - \boldsymbol{v}(\boldsymbol{n}))\big)\Big] + \lambda R(\boldsymbol{v}) \tag{4.7}$$

$$\boldsymbol{v}^- = \arg\min_{\boldsymbol{v}} \sum_{\boldsymbol{n}\in\mathcal{B}} \Big[d\big(x_{2t+1}(\boldsymbol{n}) - x_{2t+2}(\boldsymbol{n} - \boldsymbol{v}(\boldsymbol{n}))\big)\Big] + \lambda R(\boldsymbol{v}) \tag{4.8}$$

Pour la simple raison que les vecteurs $\boldsymbol{v}^+$ et $\boldsymbol{v}^-$ ont été estimés de manière indépendante, ils n'ont aucune raison à priori de minimiser le critère J précédemment défini et ne peuvent donc minimiser l'énergie du bloc $\mathcal{B}$ de l'image de détail h_t.

On souhaite ainsi estimer conjointement ce couple de vecteurs de façon à minimiser le critère J. Cependant, la minimisation directe de ce problème d'optimisation à deux paramètres est difficile et sa complexité quadratique est largement prohibitive. Nous proposons dans la section suivante une solution quasi-optimale, permettant de trouver un couple de vecteurs $\boldsymbol{v}^+$ et $\boldsymbol{v}^-$ constituant un minimum local de J .

4.1.2 Prédiction itérative bidirectionnelle jointe

La minimisation de J peut se faire par une suite de minimisations alternées du champ avant $\boldsymbol{v}^+$ et du champ arrière $\boldsymbol{v}^-$, en prenant compte des champs précédemment estimés. Nous avons ainsi présenté un algorithme itératif [105], capable de minimiser J et convergeant vers un minimum local. Un des intérêts de cet algorithme réside dans le fait qu'il ne nécessite pas la construction d'un nouvel estimateur de mouvements et repose sur un opérateur d'appariement de blocs quelconque que nous notons BM. Pour un bloc B courant et une image de référence x donnée, BM est défini comme un opérateur capable de fournir un vecteur $\boldsymbol{v}$=BM($\mathcal{B}$,x), pointant vers un bloc de l'image de référence et minimisant le coût $D + \lambda R$ associé au bloc $\mathcal{B}$.

Notre algorithme itératif de prédiction bidirectionnelle jointe permet de trouver les vecteurs $\boldsymbol{v}^+$ et $\boldsymbol{v}^-$ optimaux au sens de J et donc de minimiser le coût du bloc $\mathcal{B}$. Il est dit itératif car il repose sur la construction d'une suite de couples de vecteurs $\{\boldsymbol{v}_i^+, \boldsymbol{v}_i^-\}_{i\in\mathbb{N}}$, conduisant à la convergence du critère $\{J(\boldsymbol{v}_i^+, \boldsymbol{v}_i^-)\}_{i\in\mathbb{N}}$ vers un minimum local. L'algorithme s'énonce de la façon suivante :

Initialisation

Le vecteur avant $\boldsymbol{v}_0^+$ est obtenu par un appariement de blocs classique entre le bloc $\mathcal{B}$ de l'image courante x_{2t+1} et l'image de référence x_{2t}. On a alors $\boldsymbol{v}_0^+ = \text{BM}(\mathcal{B}, x_{2t})$.

Itération i, pour $i \geq 1$

- Le vecteur arrière $\boldsymbol{v}_i^-$ est obtenu par une procédure d'appariement de blocs entre un bloc virtuel $\mathcal{B}'$ et l'image de référence $x_{2t+2}/2$. Le bloc virtuel $\mathcal{B}'$ dépend du vecteur $\boldsymbol{v}_{i-1}^+$ précédent et est défini par $\mathcal{B}'(\boldsymbol{n}) = x_{2t+1}(\boldsymbol{n}) - x_{2t}\big(\boldsymbol{n} - \boldsymbol{v}_{i-1}^+(\boldsymbol{n})\big)/2$. Cette technique revient à faire une sorte d'appariement de blocs semi-compensé en mouvement. On a alors $\boldsymbol{v}_i^- = \text{BM}(\mathcal{B}', x_{2t+2}/2)$.
- De façon similaire, le vecteur avant $\boldsymbol{v}_i^+$ est obtenu par appariement de blocs entre entre le bloc virtuel $\mathcal{B}''$ semi-compensé en mouvement défini par $\mathcal{B}''(\boldsymbol{n}) = x_{2t+1}(\boldsymbol{n}) - x_{2t+2}\big(\boldsymbol{n} - \boldsymbol{v}_i^-(\boldsymbol{n})\big)/2$ et l'image $x_{2t}/2$. On a alors $\boldsymbol{v}_i^+ = \text{BM}(\mathcal{B}'', x_{2t}/2)$.

L'initialisation de l'algorithme revient ainsi à estimer d'abord le vecteur avant $\boldsymbol{v}_0^+$ par un appariement de blocs classique. Lors de la première itération, le vecteur arrière $\boldsymbol{v}_1^-$ est alors estimé grâce à la connaissance de $\boldsymbol{v}_0^+$. Ensuite, et à chaque itération, les vecteurs avant $\boldsymbol{v}_i^+$ et arrière $\boldsymbol{v}_i^-$ sont réestimés et permettent la convergence rapide du critère J vers un minimum local. On minimise ainsi le coût du bloc $\mathcal{B}$ au sens du critère J.

Chaque itération possède une complexité équivalente à deux recherches de blocs, sans compter l'initialisation. La complexité globale de l'algorithme avec n itérations est donc équivalente à $2n + 1$ procédures de recherche de blocs. Cependant, la complexité globale peut être réduite en diminuant à chaque itération le domaine de recherche des blocs. Ceci est justifié par le fait qu'il est probable que la direction du vecteur $\boldsymbol{v}_i^+$ soit proche de celle de $\boldsymbol{v}_{i-1}^+$ et que l'on peut ainsi réestimer $\boldsymbol{v}_i^+$ sur un domaine réduit. Comme dit précédemment, cet algorithme ne repose que sur l'utilisation d'une procédure d'appariement de blocs générique BM, rendant son implémentation grandement simplifiée. On remarquera enfin qu'il est possible de stopper l'algorithme au cours d'une itération, en s'arrêtant après l'estimation du vecteur arrière $\boldsymbol{v}_i^-$; on parlera alors de demi-itération.

Nous nous proposons désormais de montrer les propriétés de convergence de cet algorithme et de montrer qu'il est nécessairement meilleur qu'une stratégie consistant à faire une estimation indépendante des champs de mouvement. En utilisant les propriétés de l'estimateur de mouvement choisi, il est possible de montrer lors de l'initialisation que :

$$\boldsymbol{v}_0^+ = \arg\min_{\boldsymbol{v}} \sum_{\boldsymbol{n}\in\mathcal{B}} d\Big[x_{2t+1}(\boldsymbol{n}) - x_{2t}(\boldsymbol{n}-\boldsymbol{v})\Big] + \lambda R(\boldsymbol{v}) \tag{4.9}$$

Il est de même possible de montrer qu'à chaque itération, on vérifie :

$$\boldsymbol{v}_i^- = \arg\min_{\boldsymbol{v}} \sum_{\boldsymbol{n}\in\mathcal{B}} d\Big[x_{2t+1}(\boldsymbol{n}) - \frac{x_{2t}(\boldsymbol{n}-\boldsymbol{v}_{i-1}^+) + x_{2t+2}(\boldsymbol{n}-\boldsymbol{v})}{2}\Big] + \lambda R(\boldsymbol{v})$$

$$\boldsymbol{v}_i^+ = \arg\min_{\boldsymbol{v}} \sum_{\boldsymbol{n}\in\mathcal{B}} d\Big[x_{2t+1}(\boldsymbol{n}) - \frac{x_{2t}(\boldsymbol{n}-\boldsymbol{v}) + x_{2t+2}(\boldsymbol{n}-\boldsymbol{v}_i^-)}{2}\Big] + \lambda R(\boldsymbol{v})$$

A l'itération i, le critère $J(\boldsymbol{v}_i^+, \boldsymbol{v}_i^-)$ vaut donc :

$$J(\boldsymbol{v}_i^+, \boldsymbol{v}_i^-) = \sum_{\boldsymbol{n}\in\mathcal{B}} d\Big[x_{2t+1}(\boldsymbol{n}) - \frac{x_{2t}(\boldsymbol{n}-\boldsymbol{v}_i^+) + x_{2t+2}(\boldsymbol{n}-\boldsymbol{v}_i^-)}{2}\Big] + \lambda\big(R(\boldsymbol{v}_i^+) + R(\boldsymbol{v}_i^-)\big) \tag{4.10}$$

La poursuite de l'algorithme et la réalisation d'une demi-itération suivante nous permet d'obtenir le vecteur arrière suivant $\boldsymbol{v}_{i+1}^-$, qui vérifie :

$$\boldsymbol{v}_{i+1}^- = \arg\min_{\boldsymbol{v}} \sum_{\boldsymbol{n}\in\mathcal{B}} d\Big[x_{2t+1}(\boldsymbol{n}) - \frac{x_{2t}(\boldsymbol{n}-\boldsymbol{v}_i^+) + x_{2t+2}(\boldsymbol{n}-\boldsymbol{v})}{2}\Big] + \lambda R(\boldsymbol{v}) \tag{4.11}$$

Le vecteur $\boldsymbol{v}_{i+1}^-$ est donc le résultat d'une minimisation d'un terme de distorsion du critère $J(\boldsymbol{v}_i^+, \boldsymbol{v})$. Sous l'hypothèse que la direction du vecteur $\boldsymbol{v}_{i+1}^-$ soit proche de $\boldsymbol{v}_i^-$, il est légitime de supposer que leurs coûts sont très proches voire égaux $R(\boldsymbol{v}_{i+1}^-) \simeq R(\boldsymbol{v}_i^-)$. En combinant les équations (4.10) et (4.11), il est alors possible de montrer que $\boldsymbol{v}_{i+1}^-$ constitue un choix nécessairement meilleur que $\boldsymbol{v}_i^-$ au sens du critère J et de montrer ainsi que :

$$J(\boldsymbol{v}_i^+, \boldsymbol{v}_{i+1}^-) \leq J(\boldsymbol{v}_i^+, \boldsymbol{v}_i^-) \tag{4.12}$$

De la même façon, on peut montrer que $J(\boldsymbol{v}_{i+1}^+, \boldsymbol{v}_{i+1}^-) \leq J(\boldsymbol{v}_i^+, \boldsymbol{v}_i^-)$ et prouver ainsi que la suite $\{J(\boldsymbol{v}_i^+, \boldsymbol{v}_i^-)\}_{i\in\mathbb{N}}$ est décroissante, bornée et donc convergente. De plus, il est possible de montrer que les champs de mouvement estimés avec cet algorithme et avec seulement une demi-itération sont toujours meilleurs au sens du critère J que des champs estimés indépendamment. Ceci revient à montrer que $J(\boldsymbol{v}_0^+, \boldsymbol{v}_1^-) \leq J(\boldsymbol{v}_*^+, \boldsymbol{v}_*^-)$, en

notant $\boldsymbol{v}_*^+$ et $\boldsymbol{v}_*^-$ les vecteurs obtenus de façon indépendante, comme spécifié dans la section 4.1.1. Cette relation est obtenue au moyen des équations (4.8) et (4.10), de l'inégalité triangulaire et en remarquant que $\boldsymbol{v}_*^+ = \boldsymbol{v}_0^+$.

La poursuite d'une seule demi-itération nous fournit alors un algorithme d'estimation de mouvement bidirectionnel conjoint efficace, conduisant à une prédiction théoriquement toujours meilleure qu'une approche où les vecteurs sont estimés de manière indépendante. De plus, les deux approches ont une complexité identique et équivalente à deux recherches globales de mouvement, renforçant d'autant plus l'intérêt de cet algorithme d'estimation de mouvement bidirectionnel conjoint.

Il est à noter que cet algorithme n'est pas spécifique à une taille de blocs fixe car il fait appel à une procédure d'appariement de blocs BM générique. Il peut ainsi s'adapter simplement à d'autres procédures d'appariement à taille de blocs variable, comme l'algorithme HVSBM décrit en section 2.2.4 et utilisé dans notre schéma de codage.

On remarquera enfin que des travaux similaires ont été proposés indépendamment dans [162] dans le cadre d'un codeur hybride vidéo MPEG-2. Cependant, la méthode retenue par les auteurs possède une complexité plus élevée que notre algorithme car elle nécessite une initialisation indépendante des champs $\boldsymbol{v}_0^+$ *et* $\boldsymbol{v}_0^-$.

4.1.3 Prédiction bidirectionnelle à vecteur de mouvement unique

La transformée temporelle de Haar est mono-directionnelle mais ne nécessite le codage que d'un seul champ de mouvement. Au contraire, la transformée $5/3$ est bidirectionnelle et utilise deux champs de mouvement, lui permettant ainsi d'effectuer une prédiction temporelle de meilleure qualité. Cependant, cet avantage a un coût car il nécessite le codage d'un champ de mouvement supplémentaire. Nous avons pu ainsi observer dans la section 3.2 du chapitre précédent qu'à bas débit, la transformée de Haar possède une efficacité de codage supérieure à la transformée $5/3$, pénalisée par le surcoût de codage engendré par son deuxième champ de mouvement.

Nous souhaitons construire une transformée temporelle capable de concilier une prédiction bidirectionnelle tout en n'utilisant qu'un *seul* champ de mouvement. En faisant l'hypothèse d'un mouvement apparent souple et uniforme entre trois images consécutives, il est possible de construire une telle transformée en se basant sur le filtre temporel $5/3$. L'idée réside dans l'utilisation d'un champ de mouvement arrière obtenu par une simple opposition de signe du champ avant, conduisant ainsi à une transformée dont la prédiction s'écrit :

$$h_t(\boldsymbol{n}) = x_{2t+1}(\boldsymbol{n}) - \frac{1}{2}\Big(x_{2t}\big(\boldsymbol{n} - \boldsymbol{v}_{2t+1}(\boldsymbol{n})\big) + x_{2t+2}\big(\boldsymbol{n} + \boldsymbol{v}_{2t+1}(\boldsymbol{n})\big)\Big) \tag{4.13}$$

L'estimation du champ de mouvement $\boldsymbol{v}_{2t+1}$ minimisant l'énergie d'un bloc $\mathcal{B}$ de h_t et donc la minimisation du critère J devient ici un problème mono-dimensionnel plus simple à résoudre. La construction d'un nouvel estimateur de mouvement est cependant nécessaire.

Des travaux similaires ont été poursuivis dans [156] où les auteurs construisent une transformée bidirectionnelle à un seul champ de mouvement en utilisant un estimateur de mouvement minimisant la somme des erreurs quadratiques avant et arrière.

4.1.4 Résultats expérimentaux

Prédiction itérative et décroissance de la distorsion

Dans le contexte de l'implémentation du codeur vidéo décrit dans le chapitre précédent, nous souhaitons tout d'abord vérifier expérimentalement les propriétés de l'algorithme itératif d'estimation jointe. Celui-ci a été implémenté au sein de la transformée 5/3 présentée dans la section 3.1.3 pour optimiser le choix des champs de mouvement mis en jeu dans la prédiction. Nous avons alors étudié les sous-bandes temporelles issues de la décomposition sur 4 niveaux des séquences vidéo *Stefan* et *Mobile*, sans codage spatial ni quantification. Plusieurs simulations ont été effectuées en faisant varier le nombre d'itérations de l'algorithme de prédiction bidirectionnelle et en le comparant avec l'approche classique où les champs de mouvement sont estimés de façon indépendante. Les tableaux Tab. 4.1, 4.2, 4.3 et 4.4 montrent les résultats obtenus en présentant la norme ℓ_1 et l'énergie moyenne (norme ℓ_2) observées sur les sous-bandes temporelles de détail, calculées sur la composante Y à différents niveaux temporels.

Norme ℓ_1	Indépendante	0.5 it	1 it	1.5 it	2 it
Niveau 1	4.70	4.21	4.04	4.03	4.01
Niveau 2	7.78	6.99	6.72	6.71	6.67
Niveau 3	11.94	10.65	10.22	10.24	10.16
Niveau 4	17.46	15.60	15.01	15.03	14.92

TAB. 4.1 – Norme ℓ_1 des images de détail de la décomposition temporelle de la séquence *Stefan* CIF 30 Hz obtenue en utilisant une estimation bidirectionnelle indépendante et en utilisant l'algorithme itératif proposé.

Énergie moyenne	Indépendante	0.5 it	1 it	1.5 it	2 it
Niveau 1	79.33	62.67	57.04	56.95	56.22
Niveau 2	194.54	154.21	141.57	141.59	139.76
Niveau 3	433.38	339.09	310.11	312.98	308.29
Niveau 4	893.81	705.19	649.74	653.49	642.21

TAB. 4.2 – Énergie moyenne des images de détail de la décomposition temporelle de la séquence *Stefan* CIF 30 Hz obtenue en utilisant une estimation bidirectionnelle indépendante et en utilisant l'algorithme itératif proposé.

Norme ℓ_1	Indépendante	0.5 it	1 it
Level 1	2.76	2.49	2.39
Level 2	5.18	4.66	4.47
Level 3	8.87	8.17	7.89
Level 4	14.66	13.97	13.48

TAB. 4.3 – Norme ℓ_1 des images de détail de la décomposition temporelle de la séquence *Mobile* CIF 30 Hz obtenue en utilisant une estimation bidirectionnelle indépendante et en utilisant l'algorithme itératif proposé.

Énergie moyenne	Indépendante	0.5 it	1 it
Level 1	31.80	23.32	20.81
Level 2	98.65	75.32	68.46
Level 3	257.48	209.46	193.39
Level 4	652.75	572.28	529.50

TAB. 4.4 – Énergie moyenne des images de détail de la décomposition temporelle de la séquence *Mobile* CIF 30 Hz obtenue en utilisant une estimation bidirectionnelle indépendante et en utilisant l'algorithme itératif proposé.

Conformément à nos attentes, on remarque que l'algorithme proposé avec une demi-itération conduit à des énergies inférieures de près de 20 % par rapport à celles obtenues avec une estimation indépendante, pour une complexité équivalente. De plus, on observe une décroissance nette des normes ℓ_1 et ℓ_2 des trames de détail à chaque itération, atteignant jusqu'à 35% après 2 itérations. Ces observations sont en accord avec les propriétés théoriques de décroissance (4.12) énoncées à la fin de la section 4.1.2.

Efficacité de codage avec le codec MC-EZBC

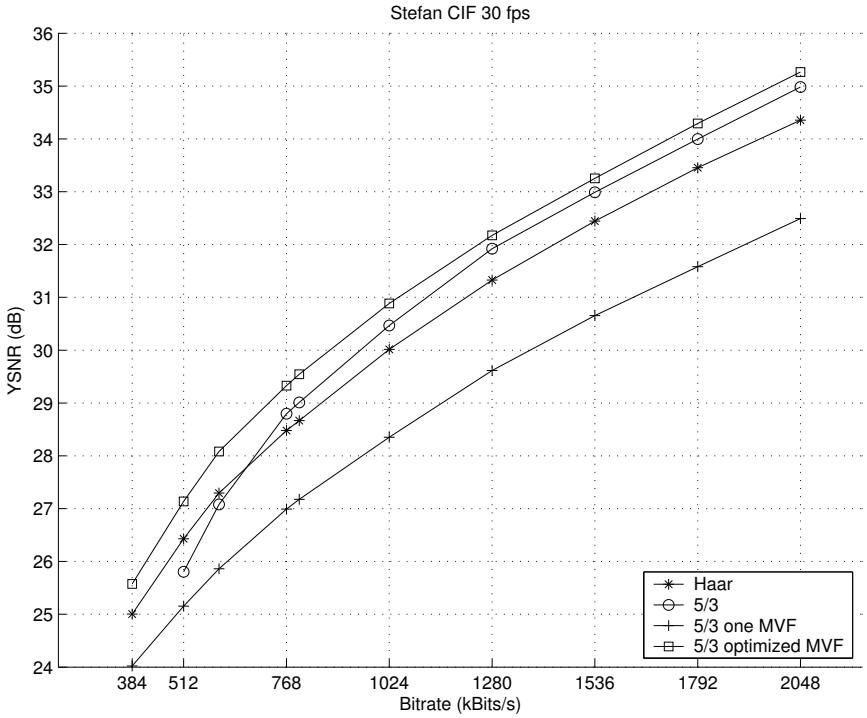
Afin d'évaluer le gain qu'apportent les deux méthodes de prédiction présentées, nous avons réalisé des simulations complètes de codage vidéo en utilisant le schéma de codage original présenté dans le chapitre précédent. Nous avons considéré les séquences vidéo couleur *Stefan*, *Foreman*, *Mobile* et *Tempête* au format CIF 30 Hz, choisies pour la variété de mouvements et de textures qu'elles offrent. Les séquences ont été décomposées sur 4 niveaux temporels et les champs de mouvement ont été estimés au 1/8ème de pixel près.

Les séquences vidéos ont été encodées entièrement, signifiant que le bitstream contient les composantes de luminance Y et de chrominances U et V de chaque image, les champs de mouvements et les informations d'en-tête. L'efficacité de codage est exprimée en terme de YSNR ou Y-PSNR, défini comme la moyenne des PSNR de la composante Y des images décodées. Les Fig. 4.2, 4.3, 4.4 et 4.5 présentent les résultats de codage obtenus en comparant les transformées temporelles suivantes :

- Transformée de Haar
- Transformée 5/3
- Transformée 5/3 à vecteur de mouvement unique, notée *5/3 one MVF*
- Transformée 5/3 avec prédiction bidirectionnelle jointe, notée *5/3 optimized MVF*

Les simulations utilisant la transformée 5/3 avec prédiction bidirectionnelle jointe des champs de mouvement ont été réalisées avec une itération, en utilisant comme mesure de distorsion d la norme ℓ_1 ou SAD. Cette transformée a donc une complexité équivalente à 3 étapes de recherche de mouvements. Par comparaison, les transformées de Haar et 5/3 à vecteur de mouvement unique possèdent une complexité équivalente à une seule étape de recherche. Pour sa part, la transformée 5/3 classique possède une complexité équivalente à 2 procédures de recherche de mouvement.

Comparons tout d'abord la transformée de Haar et la transformée 5/3. Comme précédemment, nous remarquons que la transformée 5/3 est bien plus efficace que celle de Haar à moyen et haut débits, où elle surpasse cette dernière d'environ 1 dB. Ceci peut être expliqué par une meilleure prédiction temporelle due à l'estimation bidirectionnelle du mouvement mais aussi par une mise à jour bidirectionnelle durant le calcul de la sous-

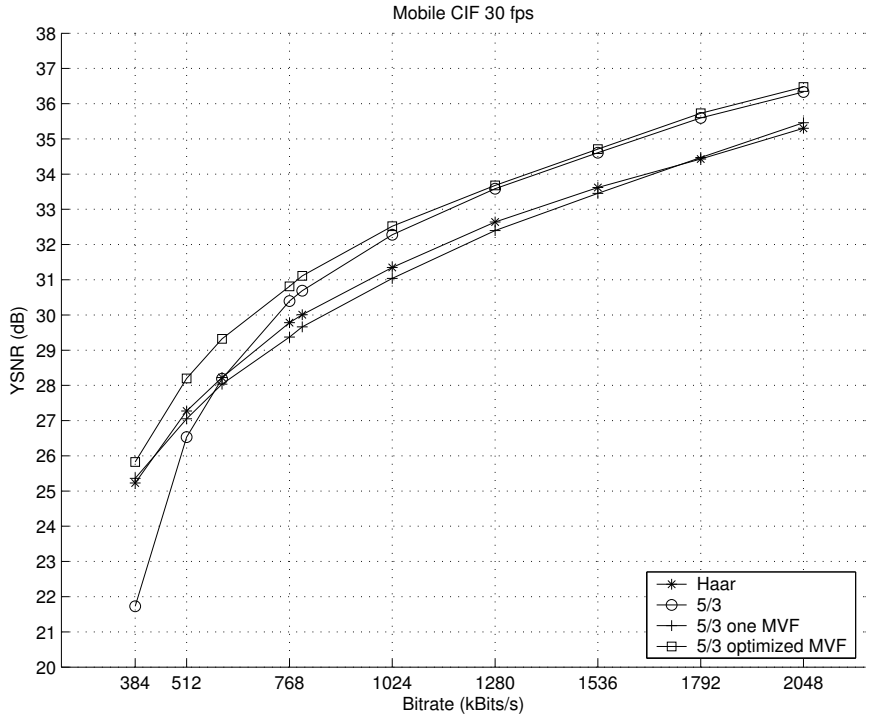


YSNR (in dB)	512 kbs	768 kbs	1024 kbs	1536 kbs	2048 kbs
Haar	26.42	28.47	30.01	32.44	34.35
5/3	25.80	28.79	30.46	32.98	34.98
5/3 one MVF	25.15	26.99	28.35	30.65	32.49
5/3 optimized MVF	27.13	29.32	30.88	33.25	35.26

FIG. 4.2 – Courbes et tableaux de débit-distorsion obtenus pour différents filtres temporels et différents débits sur la séquence *Stefan* CIF 30 Hz.

bande d'approximation. Cependant, la différence de gain est moindre sur des séquences comme *Foreman* où les mouvements complexes rotatoires de la tête du personnage réduisent les bénéfices apportés par les opérateurs bidirectionnels. Cependant, la transformée de Haar est plus efficace à faible débit car elle ne nécessite qu'un seul champ de mouvement là où la transformée 5/3 en nécessite deux. Comme ces champs sont codés sans perte, ils sont incompressibles et fixent ainsi une limite au débit minimal à laquelle une séquence vidéo peut être encodée. Le schéma 5/3 nécessite alors un débit minimal nécessairement plus important que celui de Haar et n'est donc généralement pas le plus efficace dans les très bas débits.

La transformée à vecteur de mouvement unique est un compromis entre la transformée de Haar et la transformée 5/3 : elle ne nécessite qu'un seul champ de mouvement et bénéficie cependant d'opérateurs bidirectionnels. Il est raisonnable de penser qu'elle compense les désavantages des transformées de Haar et 5/3. Ceci est confirmé expérimentalement sur la séquence *Tempête* où l'on observe des résultats supérieurs à la transformée de Haar dans les bas débits et des résultats similaires au filtre 5/3 dans les moyen et haut débits. Ceci reste vrai sur *Mobile* dans les bas débits mais pas dans les débits supérieurs où la transformée 5/3 se montre plus efficace. Cependant, sur les séquences possédant une forte activité de mouvement comme *Stefan* et *Foreman*, les résultats ne sont

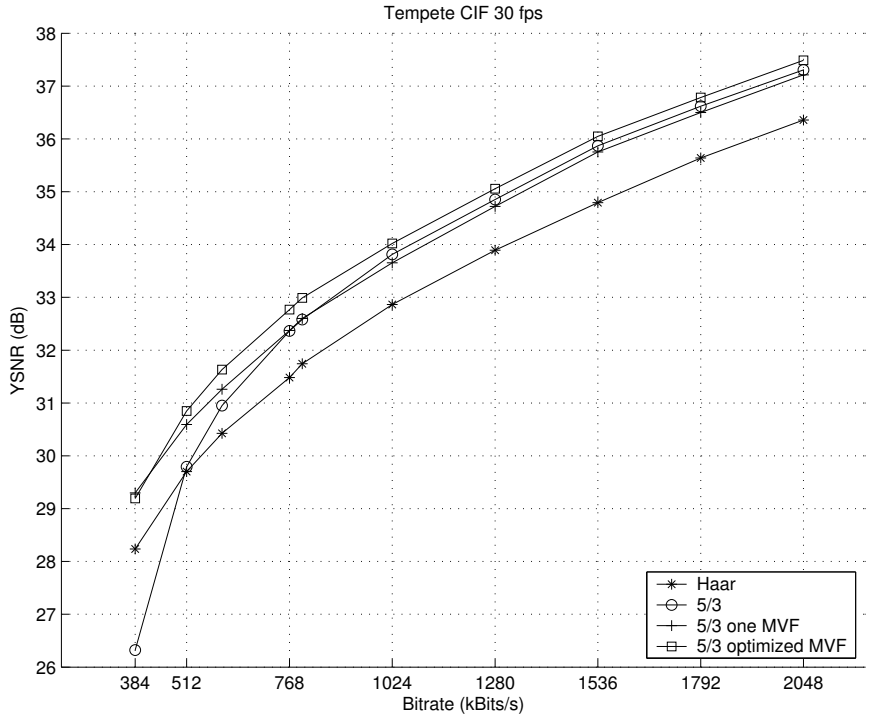


YSNR (in dB)	512 kbs	768 kbs	1024 kbs	1536 kbs	2048 kbs
Haar	27.27	29.78	31.35	33.61	35.30
5/3	26.53	30.39	32.27	34.60	36.33
5/3 one MVF	27.05	29.37	31.03	33.45	35.45
5/3 optimized MVF	28.19	30.81	32.52	34.70	36.47

FIG. 4.3 – Courbes et tableaux de débit-distorsion obtenus pour différents filtres temporels et différents débits sur la séquence *Mobile* CIF 30 Hz.

pas encourageants et sont moins bons à tous les débits que les autres transformées temporelles. En effet, ces séquences contiennent des mouvements rapides et complexes qui ne satisfont pas la contrainte d'un mouvement apparent souple et uniforme, attendue par la transformée à vecteur de mouvement unique. Il en résulte une plus grande erreur de prédiction temporelle, desservant ainsi l'efficacité de codage de la transformée 5/3 à vecteur de mouvement unique.

Nous comparons maintenant la transformée temporelle 5/3 avec prédiction itérative bidirectionnelle jointe par rapport aux autres transformées. Il apparaît clairement qu'elle donne *systématiquement* les meilleurs résultats sur toutes les séquences et à tous les débits, comparé aux autres transformées. Nous observons ainsi des gains moyens d'environ 0.5 dB avec des pointes à plus de 1.3 dB sur les séquences *Stefan* et *Mobile*. Ceci montre que l'algorithme d'estimation *jointe* des champs de mouvement avant et arrière améliore significativement la prédiction temporelle à moyen et haut débit. De plus, l'algorithme augmente la cohérence des champs de mouvement, expliquant ainsi le gain visible dans les bas débits, où une majeure partie du budget de codage est consacré aux vecteurs de mouvement. La transformée temporelle 5/3 avec prédiction jointe des champs de mouvement apparaît donc compétitive même à bas débit, comparée à la transformée de Haar. En effet, dans l'implémentation actuelle de l'algorithme itératif, une étape d'optimisation



YSNR (in dB)	512 kbs	768 kbs	1024 kbs	1536 kbs	2048 kbs
Haar	29.70	31.48	32.86	34.79	36.35
5/3	29.79	32.36	33.81	35.86	37.30
5/3 one MVF	29.29	32.37	33.65	35.75	37.21
5/3 optimized MVF	30.84	32.76	34.01	36.04	37.48

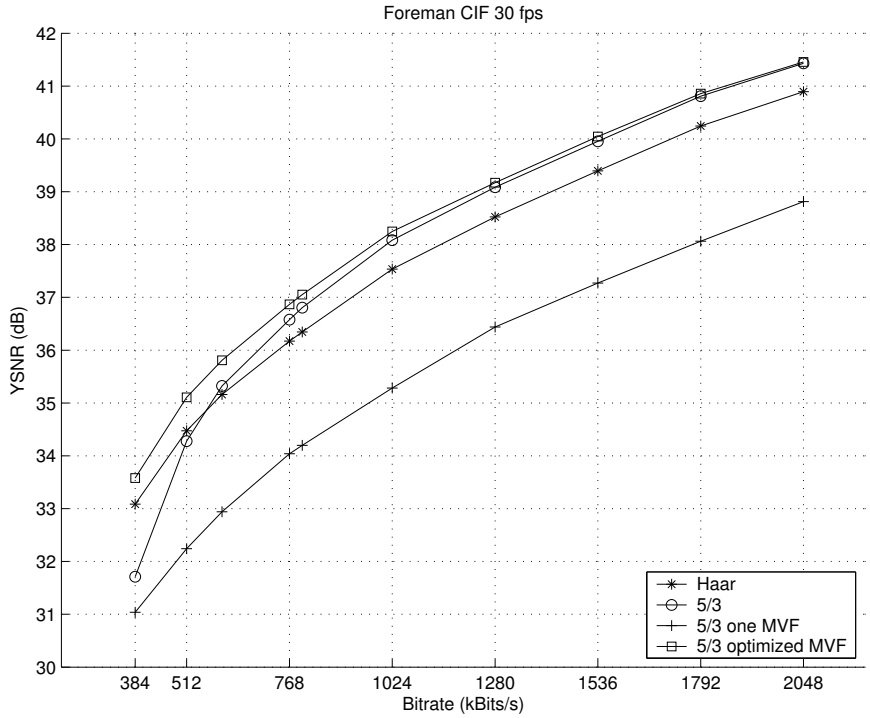
FIG. 4.4 – Courbes et tableaux de débit-distorsion obtenus pour différents filtres temporels et différents débits sur la séquence *Tempête* CIF 30 Hz.

débit-distorsion du champ de mouvement comme décrite dans la section 2.2.4 est réalisée à chaque itération par élagage. Les deux champs de mouvement ainsi obtenus sont alors nettement plus réguliers que ceux obtenus par une approche indépendante et une partie des gains de codage peut être expliqué par ce phénomène. Cette transformée possède ainsi une meilleure prédiction temporelle *et* nécessite la même quantité d'information que la transformée de Haar pour encoder ses champs de mouvement. Ceci explique pourquoi elle donne de meilleurs résultats que les autres transformées, même à bas débit.

Performance de décorrélation temporelle

Comme vu dans la section 2.2.5, le codage efficace des champs de mouvement avec une pleine exploitation de leurs dépendances spatio-temporelles est un sujet ouvert et actif, abordé par de nombreux travaux [150, 152]. Il n'est ainsi pas simple d'apprécier la performance de décorrélation opérée par une transformée temporelle à partir de sa seule efficacité de codage, du fait de la corrélation entre cette dernière et l'efficacité de codage des champs de mouvement.

Afin d'apprécier l'efficacité de la décorrélation temporelle opérée par les transformées 5/3 classique et 5/3 avec estimation jointe des champs de mouvement, nous avons procédé à des simulations de codage en excluant du budget d'encodage le débit alloué au



YSNR (in dB)	512 kbs	768 kbs	1024 kbs	1536 kbs	2048 kbs
Haar	34.47	36.17	37.53	39.39	40.89
5/3	34.27	36.57	38.08	39.95	41.43
5/3 one MVF	32.24	34.03	35.28	37.27	38.81
5/3 optimized MVF	35.10	36.86	38.24	40.04	41.45

FIG. 4.5 – Courbes et tableaux de débit-distorsion obtenus pour différents filtres temporels et différents débits sur la séquence *Foreman* CIF 30 Hz.

codage des champs de mouvement. Nous nous plaçons ainsi dans une situation idéale où les meilleurs champs de mouvement sont utilisés pour la compensation de mouvement et où leur coût de codage est nul ou négligeable. Dans le codec MC-EZBC, de tels champs de mouvement sont obtenus par la suppression de l'étape d'élagage. Ils contiennent un seul vecteur de mouvement pour chaque bloc de 4×4 pixels et sont donc presque denses. Sous ces hypothèses et en négligeant le coût de codage de ces champs, nous obtenons les résultats de codage sur la séquence *Mobile* illustrés par le Tab. 4.5.

YSNR (en dB)	512 kbs	768 kbs	1024 kbs	1536 kbs	2048 kbs
5/3	32.76	34.12	35.13	36.89	38.28
5/3 optimized MVF	33.00	34.38	35.43	37.22	38.67

TAB. 4.5 – Mesures de distorsion obtenues pour différents filtres temporels sur la séquence *Mobile* CIF 30 Hz, sans considérer le coût des champs de mouvement. La compensation de mouvement a été effectuée avec des champs quasiment denses.

Nous observons un gain en PSNR d'environ 0.3-0.4 dB sur tous les débits, en faveur de la transformée avec optimisation jointe des champs de mouvement. Ceci est comparable avec les résultats d'efficacité de codage réel à haut débit, observés précédemment lorsque

le coût de codage des champs de mouvement était inclus dans le débit total. Dans les bas débits, la différence d'efficacité de codage entre les deux transformées était nettement plus importante que 0.3 dB. Ces résultats justifient une fois de plus la propension de la transformée 5/3 optimisée jointe à lisser les champs de mouvement et à rendre leur codage moins coûteux. Ceci conclue et montre que l'algorithme itératif d'optimisation des champs de mouvement influence l'efficacité de codage en assurant à la fois une meilleure décorrélation temporelle tout en réduisant le débit nécessaire à l'encodage des champs.

Efficacité de codage avec le codec Vidwav

L'algorithme de recherche jointe itérative a de plus été intégré sur le codeur vidéo MPEG-Vidwav [97], qui utilise l'estimateur de mouvement mis en œuvre dans le codec H.264 . La mise en place de l'algorithme a cependant été effectuée seulement sur les blocs de taille 16×16 pixels, sachant que ces derniers constituent plus de 70% des décisions de modes de prédiction temporelle. De plus, l'algorithme de recherche jointe n'est utilisé qu'avec une demi-itération, n'augmentant pas ainsi la complexité du filtre temporel comparé à une transformée 5/3 classique. Les simulations expérimentales ont été conduites sur les séquences *Mobile* et *Soccer* en utilisant les conditions de scalabilité spatiales, temporelles et en débit définies dans le descriptif [25] des activités exploratoires du groupe de travail MPEG-Vidwav de Palma.

Les résultats de simulations obtenus sur les séquences *Mobile* et *Soccer* sont présentés dans les Tab. 4.6 et 4.7. Nous observons des gains en PSNR faibles d'environ 0.05 dB, loin des gains d'environ 0.7 dB obtenus avec le schéma de codage MC-EZBC. Ceci peut s'expliquer par le fait que le codec Vidwav utilise l'algorithme d'estimation de mouvement à modes de prédiction du codec H.264. En effet, cet algorithme choisit pour chaque bloc le meilleur mode de prédiction de façon à minimiser le coût du bloc. Or le mode de prédiction le plus observé lors de nos simulations est le mode bidirectionnel 16×16 utilisant un couple de vecteurs mouvements déduits des vecteurs des blocs voisins. Bien que notre algorithme réduise le coût d'un bloc, il nécessite cependant l'encodage systématique de nouveaux vecteurs. Il ne rivalise alors que rarement avec ce mode où les vecteurs ne sont pas encodés, expliquant les gains faibles observés.

YSNR (en dB)	QCIF 15 Hz 96 kbs	QCIF 15 Hz 128 kbs	CIF 15 Hz 256 kbs	CIF 30 Hz 384 kbs
5/3	28.93	30.82	28.14	29.30
5/3 optimisé	28.96	30.88	28.18	29.35

TAB. 4.6 – Mesures de distorsion obtenues avec le codec Vidwav en utilisant ou non l'algorithme de recherche bidirectionnelle optimal sur la séquence *Mobile* CIF 30 Hz.

YSNR (en dB)	QCIF 15 Hz 96 kbs	QCIF 15 Hz 128 kbs	CIF 30 Hz 256 kbs	CIF 30 Hz 384 kbs	4CIF 60 Hz 3072 kbs
5/3	31.67	35.65	31.78	35.00	36.57
5/3 optimisé	31.69	35.67	31.83	35.06	36.63

TAB. 4.7 – Mesures de distorsion obtenues avec le codec Vidwav en utilisant ou non l'algorithme de recherche bidirectionnelle optimal sur la séquence *Soccer* 4CIF 60 Hz.

De plus, nous avons souhaité évaluer de façon objective les performances du codec Vidwav muni de l'algorithme d'estimation jointe comparé au codec SVC, en cours de normalisation par le groupe ITU/MPEG JVT (*Joint Video Team*). A cette fin, nous avons choisi un extrait de la séquence vidéo haute définition *Vintage Car* de résolution 704×896 à 30 Hz que nous avons encodé avec le codec Vidwav muni de l'algorithme d'estimation jointe et avec le codec SVC JSVM 2.0. En suivant un scénario de scalabilité spatiale, temporelle et en qualité imposé, nous obtenons les résultats de codage suivants, exprimés en terme de PSNR moyen calculé sur les images décodées et présentés dans le Tab. 4.8.

YSNR (en dB)	176×224 15 Hz 96 kbs	352×448 30 Hz 384 kbs	704×896 30 Hz 1024 kbs
Vidwav + Joint	31.19	32.14	33.84
SVC JSVM 2.0	33.12	33.30	32.70

TAB. 4.8 – Mesures de distorsion obtenues pour plusieurs points de scalabilité en utilisant le codec Vidwav muni de l'algorithme de recherche jointe bidirectionnelle et le codec SVC JSVM 2.0 sur la séquence *Vintage Car* 704×896 à 30 Hz.

Le codec Vidwav muni de l'algorithme d'estimation jointe affiche de bonnes performances à la résolution nominale de la séquence vidéo où il surpasse le schéma de codage SVC d'environ 1.1 dB. Cependant, il offre une efficacité de codage moins bonne dans des résolutions inférieures. Les résultats obtenus en utilisant une itération complète n'offrent qu'une amélioration de 0.01 dB et n'ont pas été présentés. Afin d'étudier les raisons de la contre-performance observée à bas débit, nous avons tracé sur les Figs. 4.6 et 4.7 l'évolution du PSNR des images reconstruites aux débits respectifs de 1024 kbs et 384 kbs.

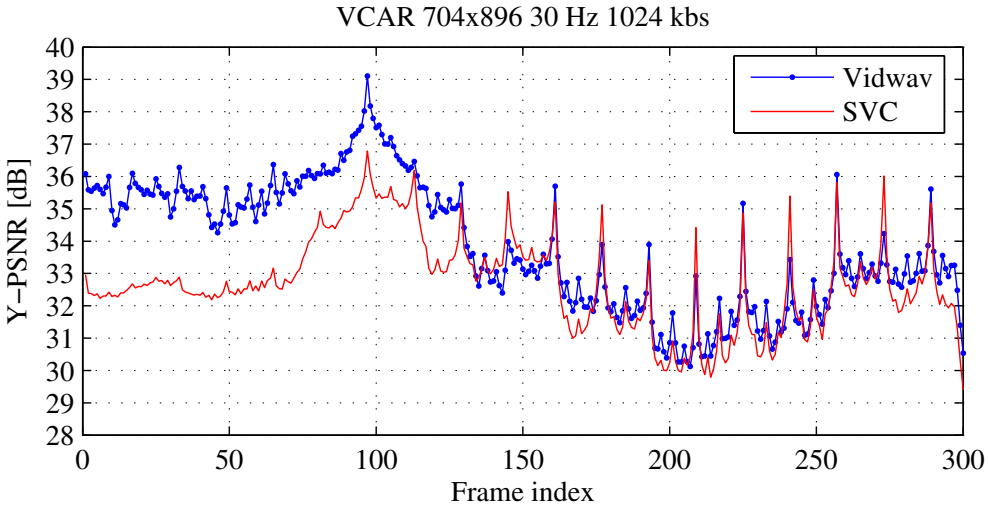


FIG. 4.6 – Comparaison de l'évolution du PSNR des images reconstruites de la séquence *Vintage Car* décodée à la résolution 704×896 à 30 Hz avec un débit de 1024 kbs, avec les codecs Vidwav et SVC.

On observe un comportement similaire dans l'évolution du PSNR des séquences décodées aux deux résolutions spatiales : le codec Vidwav surpasse le codec SVC d'environ 3 dB sur le premier tiers de la séquence. Sur les deux tiers restant et dans le cas de la résolution 704×896, le codec Vidwav offre un PSNR proche de celui du codec SVC. Cependant,

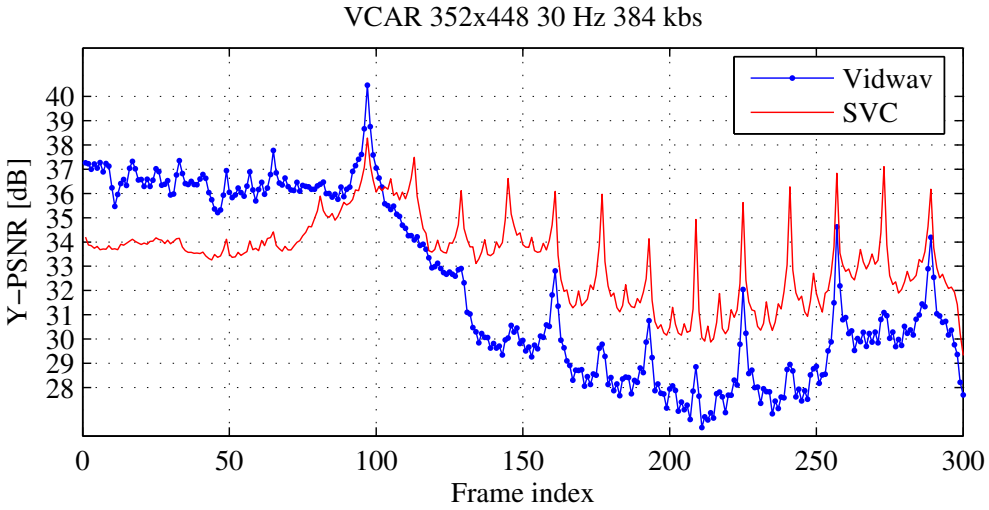


FIG. 4.7 – Comparaison de l'évolution du PSNR des images reconstruites de la séquence *Vintage Car* décodée à la résolution 352×448 à 30 Hz avec un débit de 384 kbs, avec les codecs Vidwav et SVC.

dans le cas de la résolution 352×448, le PSNR obtenu avec le codec Vidwav reste nettement en deçà, avec une baisse d'environ 4 dB. Le mouvement faible et uniforme présent dans le premier tiers de la séquence *Vintage Car* et la meilleure aptitude du codec SVC à gérer des champs de mouvement scalables peuvent justifier cet écart de PSNR. Cela explique ainsi la moins bonne efficacité du codec Vidwav observée à la résolution 352×448.

Enfin, nous avons illustré sur la Fig. 4.8 des images reconstruites par les deux codecs vidéos de la séquence *Vintage Car* à la résolution 704×896 avec un débit de 1024 kbs. On observe clairement plus de détails sur l'image reconstruite avec le codec Vidwav muni de l'estimation jointe. Le gravier, le feuillage de fond et les contours de la voiture y sont plus nets ; l'image présente ainsi un piqué supérieur à celle obtenue par le codec SVC.

4.1.5 Conclusion

L'étude de l'opérateur de prédiction impliqué dans la transformée temporelle 5/3 a permis d'élaborer deux stratégies pour améliorer son efficacité de décorrélation. Tout d'abord, la constatation que les champs de vecteurs bidirectionnels n'étaient pas choisis de façon à minimiser l'erreur de prédiction temporelle nous a conduit à construire un algorithme d'estimation conjointe du mouvement. Cet algorithme possède de nombreux avantages. Tout d'abord, il apporte des gains substantiels en terme de PSNR allant de 0.5 dB à plus de 2 dB sur une large gamme de débits et de séquences vidéos. Nous avons ainsi montré qu'il surpasse systématiquement le filtre temporel 5/3 et le filtre de Haar, même à bas débit. De plus, c'est un algorithme itératif qui possède une complexité variable, en fonction des besoins d'une application. Son utilisation avec une demi-itération possède ainsi une complexité équivalente au filtre temporel 5/3 et conduit à une meilleure efficacité de codage. Enfin, il est simple à mettre en œuvre et ne nécessite pas la construction d'un nouvel estimateur de mouvements.

En souhaitant concilier une prédiction bidirectionnelle tout en n'utilisant qu'un seul champ de mouvement, nous avons de plus construit une transformée 5/3 utilisant deux

FIG. 4.8 – Reconstruction d'une image issue du codage de la séquence *Vintage Car* de résolution 704×896 à 30 Hz pour un débit de 1024 kbs avec le codec Vidwav muni de l'estimation jointe (gauche) et avec le codec SVC JSVM 2.0 (droite).

champs de mouvement avant et arrière opposés. Cependant, bien que cette transformée offre une bonne efficacité de codage en présence d'un mouvement apparent uniforme et à bas débits, elle ne rivalise pas avec la polyvalence de la transformée 5/3 avec estimation conjointe du mouvement.

4.2 Transformée temporelle 5/3 de sens uniforme

La section précédente montre comment construire une transformée temporelle en choisissant les champs de mouvement de façon à minimiser la distorsion des sous-bandes temporelles de détail. On observe alors une augmentation *objective* de l'efficacité de codage en terme de PSNR. Cependant, bien que ce dernier soit une mesure correcte de la qualité objective, il traduit parfois mal l'apparition d'artefacts, d'effets d'anneaux (*ringing*) ou de discontinuités très visibles qui peuvent apparaîtrent dans les séquences vidéos décompressées, sans que le PSNR en soit affecté.

En particulier, un inconvénient majeur des transformations temporelles de type Haar ou 5/3 est leur propension à introduire des artefacts fantômes dans les sous-bandes d'approximation. Ces artefacts nuisent à l'efficacité globale du schéma de codage et dégradent la qualité visuelle des images décodées à bas débit. Nous nous proposons dans la section 4.2.1 d'étudier les causes de ces artefacts et commentons quelques propositions faites dans la littérature pour y remédier. Nous introduisons alors dans la section 4.2.2 la transformée temporelle 5/3 uniforme, basée sur le filtre 5/3 classique qui, par construction même, ne créé pas de tels artefacts. On montre expérimentalement que cette transformée

temporelle améliore nettement la qualité visuelle des sous-bandes d'approximation et augmente l'efficacité globale de codage. Enfin, ces résultats encourageants ont conduit à la publication d'un article de conférence [99].

Dans la continuation de nos méthodes développées dans la section précédente, nous décrivons en section 4.2.3 un algorithme de calcul optimal des champs de vecteurs mis en jeux dans la transformée 5/3 uniforme. Nous observons alors expérimentalement une nouvelle amélioration des performances et relatons nos travaux dans [98].

4.2.1 Artefacts fantômes et mise à jour

Notre schéma de codage muni de la transformée temporelle 5/3 a tendance à produire des artefacts très visibles dans certaines séquences à bas débit ou dans les images d'approximation. Ces artefacts ressemblent à des réminiscences locales d'objets présents dans des images antérieures ou postérieures à l'image courante et sont nommés pour cette raison, artefacts fantômes (*ghosting artefacts*). Ils sont particulièrement visibles sur les sous-bandes temporelles d'approximation de la séquence *Stefan*, illustrées en Fig. 4.9. On y voit ainsi clairement la présence de plusieurs pieds et de plusieurs balles de tennis.

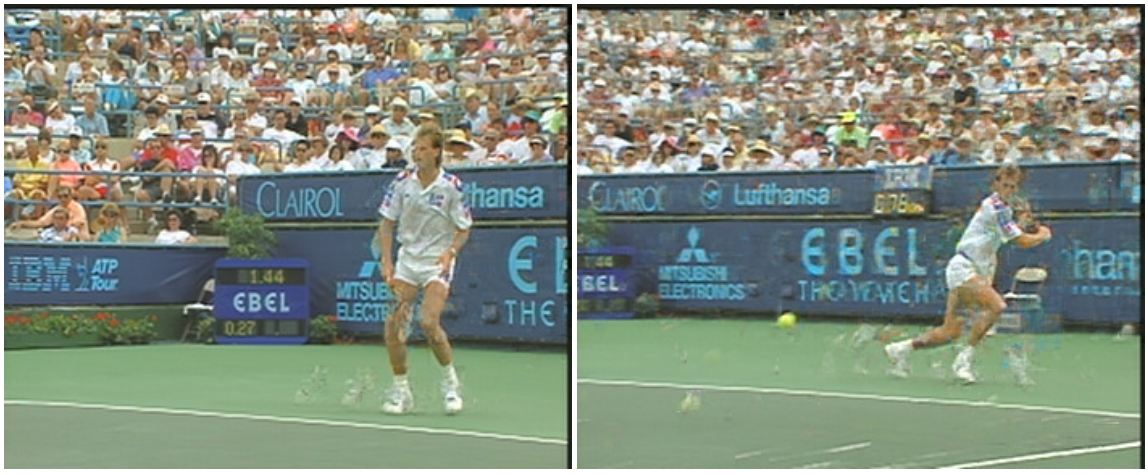


FIG. 4.9 – Présence d'artefacts fantômes sur les sous-bandes d'approximation issues du troisième niveau de la décomposition temporelle 5/3 de la séquence CIF 30 Hz *Stefan*.

Ces artefacts fantômes sont gênants pour plusieurs raisons. Tout d'abord, ils créent des discontinuités locales qui perturbent visuellement les images et induisent l'apparition de grands coefficients d'ondelettes. Ils augmentent ainsi le coût de codage des images et entraînent une diminution globale des performances du codec vidéo. De plus, ces artefacts dégradent la qualité visuelle des images d'approximation et nuisent ainsi à la scalabilité temporelle du schéma. Enfin, ils se propagent dans les niveaux temporels suivants et diminuent l'efficacité de la prédiction temporelle effectuée dans les étages supérieurs.

La présence de ces artefacts est due à l'étape de mise à jour du filtre temporel 5/3. En effet, comme vu dans la section 3.1.3, la pseudo-inversion de l'opérateur de compensation de mouvement $\mathcal{C}$ crée durant cette étape une mosaïque hétérogène de zones non-connectées, simplement connectées ou connectées de façon multiple. Le filtrage passe-bas subséquent engendre alors des zones de caractéristiques visuelles différentes, créant les artefacts.

Plusieurs approches ont été proposées pour limiter la présence des artefacts fantômes. Après avoir observé leur existence, Reichel [117] introduit par exemple une transformée de Haar non compensée en mouvement qui décide dynamiquement pour chaque pixel si il est transformé ou non, en fonction d'un critère de seuil basé sur la valeur des coefficients de détail quantifié du niveau temporel précédent. Cette solution donne des résultats visuellement probants mais n'offre pas une bonne efficacité de codage. De plus, le recours fait à un quantificateur dans la boucle d'encodage rend ce schéma de codage vidéo non scalable. Une approche plus efficace a été aussi proposée par Song [133] où les auteurs présentent une étape de mise à jour adaptative en utilisant un critère de seuil sur les coefficients, basé sur un modèle numérique simple de la vision humaine.

Enfin, on remarquera que la présence de zones non filtrées est intimement liée à la non-inversibilité des champs de mouvement durant l'étape de mise à jour. Ce problème a tout d'abord été observé par Secker et Taubman [126] où les auteurs préconisent l'utilisation d'un modèle de mouvement non basé sur des blocs mais sur une grille triangulaire déformable, presque toujours inversible (sauf en cas de retournement de maille). Ce type de modèle est cependant coûteux à encoder et difficile à estimer. Konrad [69] a étudié le problème de façon plus générale et montre l'existence d'une transformée temporelle de Haar *transverse* où le mouvement ne nécessite pas d'inversion durant l'étape de mise à jour. Cependant, cette approche n'est pas généralisable dans le cas du filtre 5/3. Poursuivant ces travaux, André [13] promeut alors l'utilisation du filtre temporel LS (2,0), obtenu par suppression de l'étape de mise à jour de la transformée temporelle 5/3 et permettant ainsi d'éliminer les artefacts fantômes de façon draconienne. Cependant, cette amputation peut aussi engendrer une baisse significative de l'efficacité de codage, pouvant atteindre 1 dB lors de son utilisation sur des séquences fluides comme *Mobile*. Des résultats expérimentaux détaillés illustrant les performances du filtre temporel 5/3 sans mise à jour sont ainsi présentés dans la section 4.2.4.

4.2.2 Transformée temporelle 5/3 de sens de mouvement uniforme

Avant de décrire les détails de la construction de la transformée 5/3 de sens de mouvement uniforme, nous nous proposons tout d'abord d'étudier quelques propriétés de connectivité de la transformée temporelle 5/3.

Transformée temporelle 5/3 classique et connectivité

Rappelons les équations de la transformée temporelle 5/3, dont les opérateurs sont illustrés par la Fig. 4.10 :

$$h_t^0(\boldsymbol{n}) = x_{2t+1}(\boldsymbol{n}) - \frac{1}{2}\big(\mathcal{C}(x_{2t}, \boldsymbol{v}_{2t+1}^+)(\boldsymbol{n}) + \mathcal{C}(x_{2t+2}, \boldsymbol{v}_{2t+1}^-)(\boldsymbol{n})\big) \tag{4.14}$$

$$l_t^0(\boldsymbol{n}) = x_{2t}(\boldsymbol{n}) + \gamma\,\mathcal{C}^{-1}(h_{t-1}, \boldsymbol{v}_{2t-1}^-)(\boldsymbol{n}) + \delta\,\mathcal{C}^{-1}(h_t, \boldsymbol{v}_{2t+1}^+)(\boldsymbol{n}) \tag{4.15}$$

$$h_t(\boldsymbol{n}) = 1/\sqrt{2}\; h_t^0(\boldsymbol{n}) \tag{4.16}$$

$$l_t(\boldsymbol{n}) = \sqrt{2}\; l_t^0(\boldsymbol{n}) \tag{4.17}$$

$$\text{avec} \begin{cases} \gamma = \delta = 1/4 & \text{si } \boldsymbol{n} \text{ est connecté des deux côtés} \\ \gamma = 1/2 \text{ et } \delta = 0 & \text{si } \boldsymbol{n} \text{ est connecté seulement à gauche} \\ \gamma = 0 \text{ et } \delta = 1/2 & \text{si } \boldsymbol{n} \text{ est connecté seulement à droite} \\ \gamma = 0 \text{ et } \delta = 0 & \text{si } \boldsymbol{n} \text{ n'est pas connecté} \end{cases}$$

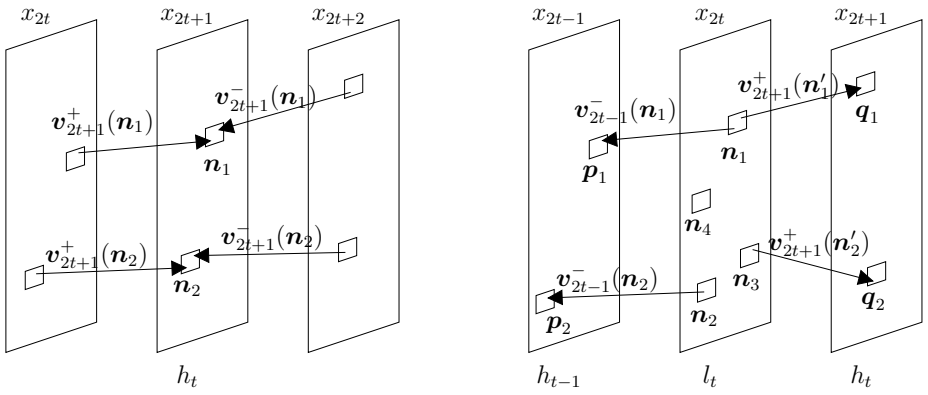


FIG. 4.10 – Opérateur de prédiction (gauche) et de mise à jour (droite) mis en jeu dans la transformée temporelle 5/3.

Lors de la prédiction décrite par l'équation (4.14), chaque pixel de l'image x_{2t+1} est toujours connecté à un seul pixel de l'image précédente x_{2t} et à un seul autre de l'image suivante x_{2t+2}. Chaque pixel de x_{2t+1} sera ainsi toujours prédit bidirectionnellement. C'est une propriété importante qui améliore généralement l'efficacité de codage, en particulier en présence d'un mouvement fluide. C'est aussi une des raisons pour laquelle la transformée 5/3 assure une meilleure décorrélation temporelle que la transformée de Haar. Cependant, lorsqu'une zone de x_{2t+1} ne peut être prédite bidirectionnellement, par exemple lors d'une occlusion ou d'une coupure de plan (*scene cut*), cette propriété n'est pas souhaitable car un des pixels prédicteur sera incorrect.

Lors de l'étape de mise à jour décrite par l'équation (4.15), les possibilités de connectivité sont nettement plus grandes. Ainsi, chaque pixel de l'image x_{2t} peut être connecté à zéro, un ou plusieurs pixels de l'image précédente x_{2t-1} et à zéro, un ou plusieurs pixels de l'image suivante x_{2t+1}. En fonction de l'état de connectivité d'un pixel et comme précisé dans la section 3.1.3, il sera alors filtré bidirectionnellement s'il est connecté dans l'image précédente et dans l'image suivante, monodirectionnellement si il est connecté dans une seule de ces images et ne sera pas filtré si il n'est pas connecté.

Le tableau 4.9 résume dans le cas de la transformée temporelle 5/3 les relations entre l'état de connexion d'un pixel et le filtrage qu'il subit lors des étapes de prédiction et de mise à jour. Il se lit comme ceci : lors de l'étape de prédiction, tous les pixels sont connectés bidirectionnellement et sont prédits par le prédicteur 5/3. Il ne peut y avoir de pixels non-connectés ou simplement connectés lors de cette étape. De plus, lors de la mise à jour, les pixels non-connectés ne sont pas filtrés, les pixels connectés sur une seule image sont filtrés par un filtre de type Haar et les pixels connectés bidirectionnellement sont filtrés par un filtre 5/3.

État de connexion d'un pixel	0	1	2
Prédiction P	N/A	N/A	Prédiction 5/3
Mise à jour U	Pas de filtrage	Filtrage Haar	Filtrage 5/3

TAB. 4.9 – Relations entre l'état de connexion d'un pixel non connecté (0), connecté sur une seule des deux images (1) et connecté sur les deux images (2) et le filtrage qu'il subira dans la transformée temporelle 5/3 classique.

Comme vu précédemment, l'opération de mise à jour induit la création de zones non-filtrées et filtrées au sein des sous-bandes temporelles d'approximation l_t. Ces zones partagent des caractéristiques différentes et leur mélange créé une mosaïque de régions inhomogènes, causant l'apparition des artefacts fantômes mentionnés plus haut.

Transformée temporelle 5/3 uniforme

Nous souhaitons donc construire une transformée temporelle où chaque pixel soit *toujours* connecté au moins à un autre, lors des étapes de prédiction et de mise à jour. Ceci est possible en introduisant deux autres champs de mouvement lors de la mise à jour mais cette solution ne se relève pas rentable, à cause du surcoût engendré par le codage des champs supplémentaires.

Une façon simple pour parvenir à cette propriété consiste à modifier la transformée temporelle 5/3 classique en utilisant deux champs de mouvement orientés dans la même direction, comme illustré par la Fig. 4.11. Cette modification nous conduit à la construction d'une nouvelle transformée, nommée transformée temporelle 5/3 uniforme, en raison du sens uniforme des champs de mouvement qu'elle met en jeu.

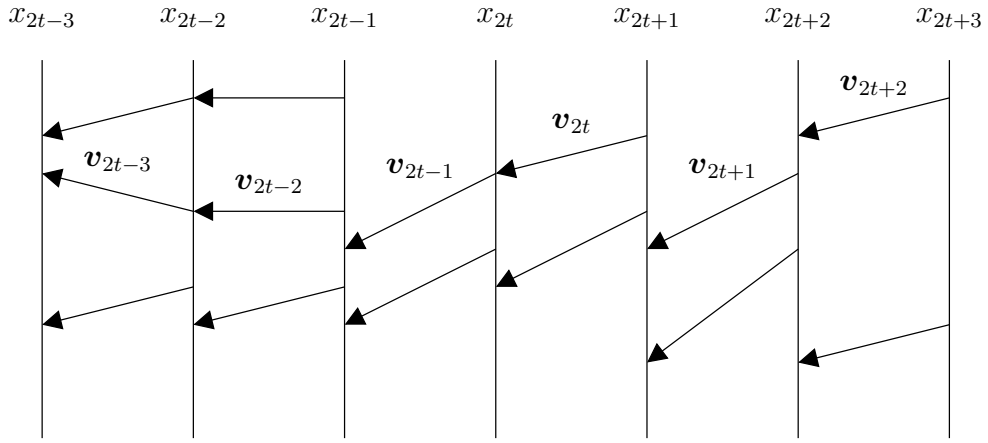


FIG. 4.11 – Orientation du mouvement dans la transformée 5/3 uniforme.

La notation indicée $+$ et $-$ n'étant plus nécessaire pour indiquer le sens de prédiction des champs de mouvement $\boldsymbol{v}$, ces derniers sont désormais notés $\boldsymbol{v}_t$ où t peut prendre des valeurs paires ou impaires. On définit ainsi un champ de vecteur mouvement arrière $\boldsymbol{v}_t$, prédisant chaque image x_t à partir de l'image suivante x_{t+1}. Le choix de la direction arrière est arbitraire et aurait pu être fait dans la direction avant.

La transformée temporelle 5/3 uniforme peut alors s'exprimer sous forme lifting par les opérateurs de prédiction et de mise à jour décrits par les Figs. 4.12 et 4.13. Elle est alors régie par les équations suivantes :

$$h_t(\boldsymbol{n}) = x_{2t+1}(\boldsymbol{n}) + \alpha\,\mathcal{C}^{-1}(x_{2t}, \boldsymbol{v}_{2t})(\boldsymbol{n}) + \beta\,\mathcal{C}(x_{2t+2}, \boldsymbol{v}_{2t+1})(\boldsymbol{n}) \tag{4.18}$$

$$\text{avec} \begin{cases} \alpha = \beta = -1/2 & \text{si } \boldsymbol{n} \text{ est connecté des deux côtés} \\ \alpha = 0 \text{ et } \beta = 1 & \text{si } \boldsymbol{n} \text{ n'est connecté qu'à gauche} \end{cases}$$

$$l_t(\boldsymbol{n}) = x_{2t}(\boldsymbol{n}) + \delta\,\mathcal{C}^{-1}(h_{t-1}, \boldsymbol{v}_{2t-1})(\boldsymbol{n}) + \gamma\,\mathcal{C}(h_t, \boldsymbol{v}_{2t})(\boldsymbol{n}) \tag{4.19}$$

$$\text{avec} \begin{cases} \delta = \gamma = 1/4 & \text{si } \boldsymbol{n} \text{ est connecté des deux côtés} \\ \delta = 0 \text{ et } \gamma = 1/2 & \text{si } \boldsymbol{n} \text{ n'est connecté qu'à gauche} \end{cases}$$

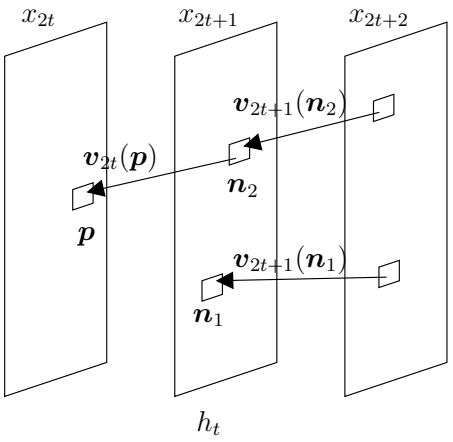


FIG. 4.12 – Opérateur de prédiction mis en jeu dans la transformée 5/3 uniforme. Le pixel $\boldsymbol{n}_1$ est simplement connecté tandis que le pixel $\boldsymbol{n}_2$ est connecté des deux côtés.

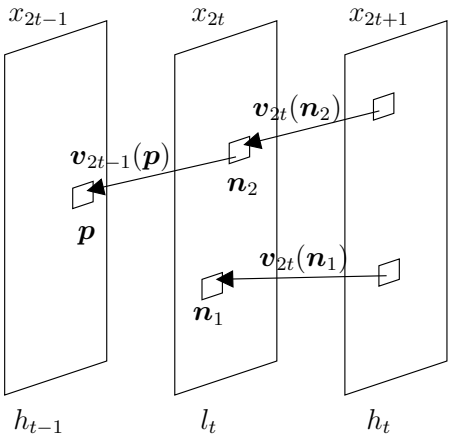


FIG. 4.13 – Opérateur de mise à jour utilisé dans la transformée 5/3 uniforme. Le pixel $\boldsymbol{n}_1$ est simplement connecté tandis que le pixel $\boldsymbol{n}_2$ est connecté des deux côtés.

Cette transformation possède une propriété importante : chaque pixel est toujours connecté à un autre dans l'image précédente, durant l'étape de prédiction et durant la mise à jour. Contrairement à la transformée temporelle 5/3 classique, cette caractéristique assure ainsi ne jamais avoir de zone découverte durant l'étape de mise à jour et permet alors à chaque pixel de toujours bénéficier au moins d'un filtrage temporel passe-bas mono-directionnel.

Lors de la prédiction décrite par l'équation (4.18), chaque pixel simplement connecté est prédit monodirectionnellement par un filtre de Haar et chaque pixel connecté bidirectionnellement est prédit par un filtre de type 5/3. Ainsi, comparé au filtre temporel 5/3 classique, la prédiction n'est pas toujours bidirectionnelle. Ceci est souvent un avan-

tage car le manque de connectivité dans une direction est souvent lié à l'occultation ou à l'apparition d'un objet, qui ne peuvent ainsi être prédit que dans une direction.

L'étape de mise à jour décrite par l'équation (4.19) est très similaire à celle de la prédiction. Chaque pixel subit ainsi un filtrage passe-bas mono ou bidirectionnel et il n'y a donc pas de pixels non-filtrés comme dans le filtre temporel 5/3 classique. Cette mise à jour est donc plus régulière et doit conduire à des images filtrées plus homogènes que celle obtenues avec le filtre temporel 5/3 classique. Comme évoqué précédemment dans la section 4.2.1, cette propriété doit contribuer à réduire la source majeure de création d'artefacts fantômes dans la transformée temporelle.

Le tableau 4.10 dresse les propriétés de connectivité de la transformé temporelle 5/3 uniforme. Mis en correspondance avec celui de la transformée 5/3 classique présenté en Fig. 4.9, il montre l'intérêt principal de la transformée 5/3 uniforme : tous les pixels sont au moins prédits et mis à jour par un pixel de l'image précédente. Ceci permet ainsi de résoudre élégamment le problème de gestion des occlusions lors de la prédiction et des zones non-connectées lors de la mise à jour.

Connectivité	0	1	2
Prédiction P	N/A	Prédiction Haar	Prédiction 5/3
Mise à jour U	N/A	Filtrage Haar	Filtrage 5/3

TAB. 4.10 – Relations entre l'état de connexion d'un pixel non connecté (0), connecté sur une seule des deux images (1) et connecté sur les deux images (2) et le filtrage qu'il subira dans la transformée temporelle 5/3 uniforme.

On notera qu'une variante de cette transformée temporelle, utilisant des champs de vecteurs orientés dans la même direction, a été étudiée indépendamment par Golwelkar et Woods [54]. Cependant, leur volonté sous-jacente semblait être avant tout de mettre en œuvre une transformée temporelle bidirectionnelle sans aborder le problème des artefacts fantômes. Ils mettent ainsi surtout en avant leur habilité à pouvoir traiter les images au fil de l'eau mais n'établissent pas de comparaison avec la transformée temporelle 5/3 classique. Enfin, les auteurs présentent des résultats expérimentaux nettement en deçà de ceux observés dans la section 4.2.4.

4.2.3 Prédiction bidirectionnelle optimale des zones découvertes

Nous abordons dans cette section le problème de l'estimation optimale des champs de mouvement impliqués dans la transformée 5/3 uniforme. En suivant la même approche que celle décrite dans la section 4.1 dans le cas de la transformée 5/3 classique, on s'intéresse à la minimisation d'un critère J basé sur les images de détail h_t, en espérant ainsi réduire leur coût de codage. Développons le critère J dans le cas de la transformée 5/3 uniforme décrite par l'équation de prédiction (4.18)et illustrée par la Fig. 4.12 :

$$
\begin{aligned}
J(\boldsymbol{v}_{2t}, \boldsymbol{v}_{2t+1}) &= \sum_{\boldsymbol{n}\in\mathcal{B}} d\Big[h_t(\boldsymbol{n})\Big] + \lambda\big(R(\boldsymbol{v}_{2t}) + R(\boldsymbol{v}_{2t+1})\big) \qquad (4.20)\\
&= \sum_{\boldsymbol{n}\in\mathcal{B}_1} d\Big[x_{2t+1}(\boldsymbol{n}) - \mathcal{C}(x_{2t+2}, \boldsymbol{v}_{2t+1})(\boldsymbol{n})\Big] + \lambda\big(R(\boldsymbol{v}_{2t}) + R(\boldsymbol{v}_{2t+1})\big)\\
&\quad + \sum_{\boldsymbol{n}\in\mathcal{B}_2} d\Big[x_{2t+1}(\boldsymbol{n}) - \frac{\mathcal{C}^{-1}(x_{2t}, \boldsymbol{v}_{2t})(\boldsymbol{n}) + \mathcal{C}(x_{2t+2}, \boldsymbol{v}_{2t+1})(\boldsymbol{n})}{2}\Big]
\end{aligned}
$$

où d est une mesure de distorsion quelconque, λ la contrainte de Lagrange, R le coût de codage d'un vecteur, $\mathcal{C}$ l'opérateur de compensation de mouvement et $\mathcal{B}$ est un bloc de l'image courante x_{2t+1} à prédire. Le bloc $\mathcal{B}$ est subdivisé en deux sous-ensembles $\mathcal{B} = \mathcal{B}_1 + \mathcal{B}_2$ où $\mathcal{B}_1$ est l'ensemble des points connectés uniquement sur l'image suivante et $\mathcal{B}_2$ l'ensemble des points connectés bidirectionnellement.

Il est possible de minimiser directement J, revenant ainsi à estimer conjointement $\boldsymbol{v}_{2t}$ et $\boldsymbol{v}_{2t+1}$. Cependant, la minimisation d'un problème à deux paramètres possède une complexité quadratique dont le coût de calcul est prohibitif. Il nous faut donc chercher une solution sous-optimale.

Nous nous proposons ici d'estimer d'abord le champ de mouvement $\boldsymbol{v}_{2t}$ puis d'estimer le champ $\boldsymbol{v}_{2t+1}$, en fonction de $\boldsymbol{v}_{2t}$ de façon à minimiser J. C'est une minimisation alternée qui peut être répétée itérativement de manière à converger vers un optimum local, de façon similaire à l'algorithme décrit dans la section 4.1.2. De plus, il est aisé de montrer que cette approche est nécessairement meilleure qu'une estimation indépendante de $\boldsymbol{v}_{2t}$ et $\boldsymbol{v}_{2t+1}$.

Algorithme

Considérons la Fig. 4.12. L'estimation de $\boldsymbol{v}_{2t}$ est faite tout d'abord par un algorithme d'appariement de blocs classique en prenant x_{2t} comme image courante et x_{2t+1} comme image de référence. La compensation de l'image x_{2t+1} par le champ $\boldsymbol{v}_{2t}$ fournit ainsi une bonne approximation de x_{2t}. De plus, en parcourant le champ $\boldsymbol{v}_{2t}$ à l'envers, on peut obtenir une approximation de x_{2t+1} par compensation inverse de l'image x_{2t}, donnée par $\mathcal{C}^{-1}(x_{2t}, \boldsymbol{v}_{2t})$ et illustré par la Fig. 4.14.

FIG. 4.14 – Compensation inverse d'une image $\mathcal{C}^{-1}(x_{2t}, \boldsymbol{v}_{2t})$ provenant de la séquence *Foreman*. Les zones noires représentent les zones découvertes.

Du fait de la non-inversibilité de $\mathcal{C}$, la compensée inverse $\mathcal{C}^{-1}(x_{2t}, \boldsymbol{v}_{2t})$ n'est pas définie partout. Elle comporte quelques zones découvertes car tous les pixels de x_{2t+1} ne sont pas reliés à x_{2t}. Cependant, ces régions ne correspondent pas nécessairement à des zones occlues par des déplacements d'objets. Au vu de la Fig. 4.14, il est ainsi raisonnable de penser que ces zones puissent être prédites par l'image suivante x_{2t+2}. De plus, chaque pixel de l'image x_{2t+1} est connecté à un pixel de l'image x_{2t+2}. On peut ainsi aisément prédire x_{2t+1} par la compensée directe $\mathcal{C}(x_{2t+2}, \boldsymbol{v}_{2t+1})$.

Nous souhaitons donc poursuivre cette idée : prédire l'image x_{2t+1} d'une part à partir de $\mathcal{C}^{-1}(x_{2t}, \boldsymbol{v}_{2t})$, malgré les zones découvertes qu'elle comporte et d'autre part à partir de

$\mathcal{C}(x_{2t+2}, \boldsymbol{v}_{2t+1})$. Ceci revient à effectuer tout d'abord une première prédiction incomplète par $\mathcal{C}^{-1}(x_{2t}, \boldsymbol{v}_{2t})$. Chaque pixel de x_{2t+1} bénéficiera alors d'une prédiction supplémentaire par $\mathcal{C}(x_{2t+2}, \boldsymbol{v}_{2t+1})$.

On cherche ainsi à estimer $\boldsymbol{v}_{2t+1}$ en souhaitant prédire $x_{2t+1} - \mathcal{C}^{-1}(x_{2t}, \boldsymbol{v}_{2t})$ à partir de $\mathcal{C}(x_{2t+2}, \boldsymbol{v}_{2t+1})$, tout en tenant compte des zones découvertes. Cependant, comment lancer une procédure d'appariement de blocs lorsque l'image courante comporte des trous ? Ceci peut être résolu grâce à l'algorithme suivant.

Estimation de $\boldsymbol{v}_{2t+1}$ sous la connaissance de $\boldsymbol{v}_{2t}$

La connaissance de $\boldsymbol{v}_{2t}$ nous permet de savoir si un pixel sera connecté avec la seule image x_{2t+2} ou avec les deux images x_{2t} et x_{2t+2}. Ainsi les pixels non-définis de $\mathcal{C}^{-1}(x_{2t}, \boldsymbol{v}_{2t})$ sont ceux connectés uniquement avec x_{2t+2} et appartiennent donc à $\mathcal{B}_1$, les autres appartenant à $\mathcal{B}_2$. Tout comme dans la section 4.1.2, on peut définir une image intermédiaire semi-compensée en mouvement a, représentant la première passe de prédiction :

$$a(\boldsymbol{n}) = \begin{cases} \frac{1}{2} x_{2t+1}(\boldsymbol{n}) & \text{si } \boldsymbol{n} \in \mathcal{B}_1 \\ x_{2t+1}(\boldsymbol{n}) - \frac{1}{2}\mathcal{C}^{-1}(x_{2t}, \boldsymbol{v}_{2t})(\boldsymbol{n}) & \text{si } \boldsymbol{n} \in \mathcal{B}_2 \end{cases} \tag{4.21}$$

Avant de pouvoir prédire l'image intermédiaire a par x_{2t+2}, il nous faut un algorithme d'estimation de mouvement tenant compte d'une pondération pour chaque pixel car les pixels de a appartenant à $\mathcal{B}_1$ ont une dynamique deux fois moindre que ceux appartenant à $\mathcal{B}_2$. Définissons alors une métrique $\mathcal{M}$ pour l'estimateur de mouvement entre deux images x et y en faisant intervenir le masque de pondération w :

$$\mathcal{M}(x, y) = \sum_{\boldsymbol{n} \in \mathcal{B}} d\Big[w(\boldsymbol{n}) \big(x(\boldsymbol{n}) - y(\boldsymbol{n}) \big) \Big] \tag{4.22}$$

$$\text{où } w(\boldsymbol{n}) = \begin{cases} 2 & \text{si } \boldsymbol{n} \in \mathcal{B}_1 \\ 1 & \text{si } \boldsymbol{n} \in \mathcal{B}_2 \end{cases} \tag{4.23}$$

On peut alors montrer que l'estimation du champ de mouvement $\boldsymbol{v}_{2t+1}$ par une procédure d'appariement de blocs munie de la métrique $\mathcal{M}$, en prenant a comme image courante et $x_{2t}/2$ comme image de référence, est équivalente à la minimisation du critère J pour le bloc $\mathcal{B}$, connaissant $\boldsymbol{v}_{2t}$. En effet, en utilisant l'opérateur d'appariement de blocs BM défini en section 4.1.2, on montre que :

$$\begin{aligned}
\boldsymbol{v}_{2t+1} &= \text{BM}_{\mathcal{M}}(a, \frac{1}{2} x_{2t}) \\
&= \arg\min_{\boldsymbol{v}_{2t+1}} \sum_{\boldsymbol{n} \in \mathcal{B}} d\Big[w(\boldsymbol{n}) \big(a(\boldsymbol{n}) - \frac{1}{2}\mathcal{C}(x_{2t}, \boldsymbol{v}_{2t+1})(\boldsymbol{n}) \big) \Big] + \lambda \big(R(\boldsymbol{v}_{2t}) + R(\boldsymbol{v}_{2t+1}) \big) \\
&= \sum_{\boldsymbol{n} \in \mathcal{B}_1} d\Big[x_{2t+1}(\boldsymbol{n}) - \mathcal{C}(x_{2t+2}, \boldsymbol{v}_{2t+1})(\boldsymbol{n}) \Big] + \lambda \big(R(\boldsymbol{v}_{2t}) + R(\boldsymbol{v}_{2t+1}) \big) \\
&\quad + \sum_{\boldsymbol{n} \in \mathcal{B}_2} d\Big[x_{2t+1}(\boldsymbol{n}) - \frac{\mathcal{C}^{-1}(x_{2t}, \boldsymbol{v}_{2t})(\boldsymbol{n}) + \mathcal{C}(x_{2t+2}, \boldsymbol{v}_{2t+1})(\boldsymbol{n})}{2} \Big] \\
&= \arg\min_{\boldsymbol{v}} J(\boldsymbol{v}_{2t}, \boldsymbol{v})
\end{aligned}$$

On obtient alors un champ optimal $\boldsymbol{v}_{2t+1}$ qui minimise l'énergie des images de détail h_t. Ne nécessitant que deux procédures de recherche de blocs, la transformée temporelle

5/3 uniforme ainsi définie a une complexité équivalente à celle d'une transformée 5/3 classique ou d'une transformée 5/3 avec une prédiction optimisée jointe à une demi-itération, telle que décrite dans la section 4.1.

Enfin, il est intéressant de remarquer que l'algorithme ci-dessus peut être étendu de façon itérative comme dans le cas de la prédiction jointe itérative, en réestimant successivement $\boldsymbol{v}_{2t}$ avec la connaissance de $\boldsymbol{v}_{2t+1}$ puis en réestimant $\boldsymbol{v}_{2t+1}$, etc. On converge ainsi vers un minimum local. Cependant, les résultats expérimentaux montrent que le gain obtenu par cette approche itérative est marginal, pour une complexité accrue. Nous nous cantonnerons donc à la transformée 5/3 uniforme précédemment définie.

4.2.4 Résultats expérimentaux

Réduction des artefacts fantômes

La série d'images présentée en Fig. 4.15 illustre les sous-bandes temporelles d'approximation issues de la décomposition de la séquence vidéo *Stefan* sur quatre niveaux temporels, obtenues avec le filtre 5/3 uniforme et le filtre 5/3 classique. Les images issues de la décomposition uniforme sont visiblement de meilleure qualité et ne présentent pas d'artefacts fantômes. On remarque ainsi que de nombreux gribouillis, présents aux abords des objets en mouvement dans le cas du filtre temporel 5/3 classique, sont absents dans le cas du filtre 5/3 uniforme. L'amélioration de l'aspect visuel des sous-bandes temporelles d'approximation permet d'augmenter ainsi la qualité de la scalabilité temporelle du schéma de codage.

Connectivité

Le Tab. 4.11 présente des résultats intéressants sur l'état de connectivité des pixels lors de l'étape de mise à jour de la transformée temporelle 5/3 uniforme et de la transformée 5/3 classique. Conformément aux propriétés de la transformée 5/3 uniforme, le pourcentage de pixels non-connectés, donc non-filtrés est nul à tous les niveaux. Par comparaison, ce taux atteint près de 10 % dans le dernier niveau temporel de la transformée 5/3 classique. De plus, on remarque que les taux de pixels simplement connectés et connectés bidirectionnellement sont plus importants dans le cas uniforme que dans le cas classique, à tous les niveaux temporels. Il en résulte une prédiction et un filtrage passe-bas de meilleure qualité.

Efficacité de codage

Afin d'évaluer son efficacité de codage, la transformée 5/3 uniforme a été mise en place au sein du codec MC-EZBC. Des simulations ont alors été conduites sur les séquences *Mobile*, *Tempête* et *City*, en utilisant une décomposition temporelle sur 5 niveaux et une estimation du mouvement au 1/8ème de pixel près. Les résultats exprimés en terme de Y-PSNR sont présentés dans les Tabs. 4.12, 4.13 et 4.14 et sont mis en comparaison avec le filtre temporel 5/3 classique, muni de la prédiction optimisée jointe itérative décrite dans la section 4.1.2 et avec le filtre temporel 5/3 classique sans mise à jour.

Afin de comparer notre schéma de codage avec un codec normatif à l'état-de-l'art, nous avons ajouté les performances débit-distorsion obtenues avec le codec H.264/AVC, dont les caractéristiques sont rappelées dans la section 2.1.2. Les conditions de simulations sont celles utilisées lors de l'appel à propositions MPEG [8] : utilisation du JSVM 7.3 avec

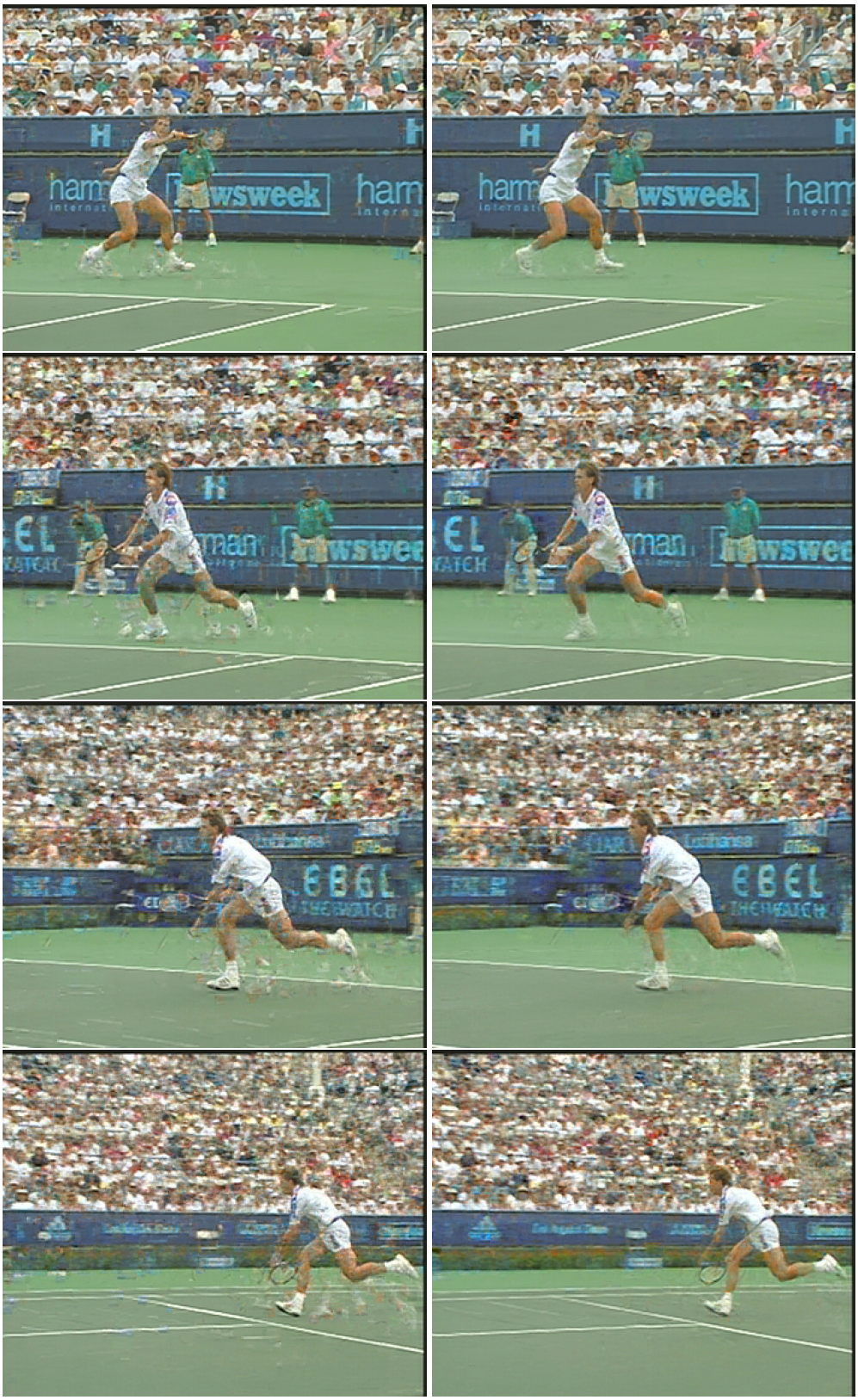


FIG. 4.15 – Images d'approximation du quatrième niveau issues de la décomposition temporelle de la séquence *Stefan* CIF 30 Hz obtenue avec le filtre 5/3 classique (à gauche) et avec le filtre 5/3 uniforme (à droite).

Transformée 5/3 classique	0	1	2
Niveau temporel 1	0.67	6.86	92.47
Niveau temporel 2	2.23	13.79	83.98
Niveau temporel 3	5.46	22.80	71.74
Niveau temporel 4	10.73	29.62	59.65

Transformée 5/3 uniforme	0	1	2
Niveau temporel 1	0.00	4.13	95.87
Niveau temporel 2	0.00	9.01	90.99
Niveau temporel 3	0.00	17.42	82.58
Niveau temporel 4	0.00	26.67	73.33

TAB. 4.11 – Pourcentage de pixels non-connectés (0), simplement connectés (1) et connectés bidirectionnellement (2) durant l'étape de mise à jour de la transformée 5/3 uniforme et de la 5/3 classique, à différents niveaux temporels. Ces résultats ont été obtenus sur la décomposition temporelle de la séquence *Foreman* CIF 30 Hz sur 4 niveaux.

optimisation débit-distorsion, codage CABAC, contrôle de débit et utilisation de 5 images de référence.

YSNR (en dB)	512 kbs	768 kbs	1024 kbs	1536 kbs
Filtre 5/3 uniforme	30.23	32.39	33.85	36.07
Filtre 5/3 classique optimisé	29.64	31.80	33.22	35.08
Filtre 5/3 sans mise à jour	28.82	31.04	32.55	34.86
H.264/AVC	29.90	31.88	33.68	35.27

TAB. 4.12 – Courbes de débit-distorsion obtenues pour différents filtres temporels et différents débits sur la séquence *Mobile* CIF 30 Hz.

YSNR (en dB)	512 kbs	768 kbs	1024 kbs	1536 kbs
Filtre 5/3 uniforme	32.04	33.82	35.03	36.99
Filtre 5/3 classique optimisé	31.44	33.23	34.41	36.36
Filtre 5/3 sans mise à jour	30.62	32.46	33.75	35.73
H.264/AVC	32.31	34.01	35.15	37.07

TAB. 4.13 – Courbes de débit-distorsion obtenues pour différents filtres temporels et différents débits sur la séquence *Tempête* CIF 30 Hz.

YSNR (en dB)	3000 kbs	6000 kbs
Filtre 5/3 uniforme	36.70	38.43
Filtre 5/3 classique optimisé	36.59	38.25
Filtre 5/3 sans mise à jour	35.59	37.14
H.264/AVC	36.20	37.80

TAB. 4.14 – Courbes de débit-distorsion obtenues pour différents filtres temporels et différents débits sur la séquence *City* 4CIF 60 Hz.

On observe les très bonnes performances du filtre 5/3 uniforme comparé au filtre 5/3 classique optimisé, avoisinant des gains de 0.5 dB en moyenne et atteignant jusqu'à 1 dB sur *Mobile* à 1536 kbs, pour une complexité équivalente. On remarque aussi les bons résultats que la transformée offre en comparaison du codec H.264/AVC, sachant que ce dernier n'offre aucune forme de scalabilité. Les simulations réalisées avec le filtre 5/3 sans mise à jour montrent de plus l'importance de cette étape en terme de PSNR dans la transformée temporelle. L'augmentation de l'efficacité objective de codage apportée par la transformée 5/3 uniforme est donc réelle, bien qu'elle n'ait pas été construite explicitement dans cette optique. Il est probable que les gains en PSNR observés soient dus à la meilleure qualité des images d'approximations obtenues au cours de la décomposition temporelle, augmentant ainsi mécaniquement l'efficacité de la prédiction temporelle dans les étages supérieurs.

4.2.5 Conclusion

En étudiant l'origine des artefacts fantômes apparaissant dans les séquences décodées à bas débit, nous avons été amenés à construire une transformée temporelle basée sur la transformée 5/3 dont les champs sont orientés dans la même direction : la transformée 5/3 uniforme. Cette construction particulière assure que tous les pixels soient connectés lors des étapes de prédiction et de mise à jour temporelle. Il en résulte un filtrage passe-haut et passe-bas plus homogène, conduisant à des sous-bandes temporelles d'approximation dépourvues d'artefacts. En suivant la même approche que celle suivie pour la transformée 5/3 classique, nous avons de plus construit un algorithme optimal d'estimation des champs de mouvement mis en jeu dans la transformée 5/3 uniforme, minimisant l'erreur de prédiction temporelle. On montre alors expérimentalement que la transformée temporelle 5/3 uniforme améliore nettement la qualité visuelle des sous-bandes d'approximation tout en augmentant l'efficacité globale de codage de façon significative.

4.3 Modération de la latence

Bien que la recherche de l'efficacité de codage soit primordiale dans la construction d'une transformée temporelle, elle ne doit pas masquer d'autres problématiques couramment rencontrées lors de la mise en situation *effective* d'un codec vidéo. En plus des contraintes matérielles sur la taille mémoire ou concernant la vitesse d'exécution à prendre en compte, il faut veiller à ce que la latence intrinsèque créée par un codec vidéo ne soit pas trop importante.

Après avoir défini précisément les notions de délais et de latence dans la section 4.3.1, nous justifions en section 4.3.2 pourquoi les différentes transformées temporelles de Haar, 5/3 et assimilées ne peuvent être utilisées dans des schémas de codage $t + 2D$ pour des applications de visioconférence ou de vidéosurveillance en temps réel, du fait de la latence trop importante qu'elles engendrent. Afin de pallier à ce problème, nous présentons en section 4.3.3 une méthode flexible et générique pour réduire la latence créée par une transformée temporelle. Des résultats expérimentaux décrits en section 4.3.4 et menés sur la transformée temporelle 5/3 montrent alors l'existence d'un compromis intéressant entre latence et efficacité de codage, dépendant des besoins de l'application visée.

Ces travaux ont conduit à la publication d'un premier article de conférence [104] où seul le délai d'encodage était considéré lors de la construction d'une transformée à délai

réduit. Un deuxième article [107] a alors complété ces travaux en envisageant tous les délais et en améliorant nettement l'efficacité de la transformée précédente.

4.3.1 Introduction, latence et délais

Dans une application de type visioconférence, le délai représente simplement le temps écoulé entre la capture d'une image côté émetteur et son affichage côté récepteur. Aussi appelée latence ou retard, le délai est dépendant de nombreux facteurs et se décompose en délai de transmission réseau, durée de traitement par le processeur, délai de paquetisation, de switching routeur... Certains de ces facteurs sont dépendants de l'architecture réseau et matérielle et peuvent être réduits (délai de transmission par le réseau, durée de traitement de calcul) tandis que d'autres sont intrinsèques à l'application et restent incompressibles.

Dans le cas de notre schéma de codage $t+2D$, les modules d'estimation de mouvement, de transformation spatiale et de codage entropique créent un délai qui n'est fonction que de la puissance de calcul du processeur. De nombreuses techniques d'optimisation logicielle, matérielle et algorithmique existent pour accélérer ces calculs mais sortent largement du cadre de ce chapitre.

Par contre, les transformées temporelles utilisées dans le schéma nécessitent généralement des images situées dans le futur ; elles sont donc non-causales et introduisent alors une certaine latence dans le codec. Cette dernière n'est pas compressible et ne dépend que de la fréquence de la vidéo et du nombre d'images situées dans le futur nécessaires à la transformation d'une image. Dans le cas des filtres temporels de type Haar ou 5/3, nous verrons que cette latence est suffisamment importante pour interdire l'utilisation d'un codec $t + 2D$ dans des applications de type visioconférence.

La problématique de la latence introduite par les filtres temporels 5/3 et 9/7 a été constatée en tout premier par Parisot [93, 94]. Dans ces articles, les auteurs insistent de plus sur l'optimisation de l'occupation mémoire nécessaire à l'implémentation matérielle de tels filtres. Ils dressent ainsi les tables relatives à la latence induite et à la mémoire requise par les filtres 5/3 et 9/7 mais ne proposent pas d'alternatives pour les réduire.

4.3.2 Analyse des délais créés par différents filtres temporels

Afin de se placer dans un cadre indépendant de tout réseau matériel et de tout contexte logiciel, nous considérons désormais que les délais de transmission et de temps de calcul sont *instantanés*. Sur la base de ces hypothèses, le délai D lié à une transformée temporelle n'est alors fonction que de la fréquence vidéo f exprimée en Hertz et d'un nombre N d'images et est noté $D = N \times f$.

Délai d'encodage

Le délai d'encodage $D_e = N_e \times f$ est la durée maximale nécessaire à la transformation d'une image courante en sa sous-bande correspondante. C'est donc le nombre maximal N_e d'images situées dans le futur nécessaire à la transformation de l'image courante. Nous avons représenté sur la Fig. 4.16 une analyse temporelle de type 5/3 sur 3 niveaux où le délai maximal N_e et les délais d'encodage de chaque image sont indiqués, exprimés en nombre d'images N. L'image d'indice 2 possède ainsi un délai d'encodage de 4, signifiant qu'elle nécessite 4 images dans le futur pour pouvoir être décomposée. Le

chemin en gras désigne le chemin de traitement lié à l'image ayant le plus grand délai d'encodage $N_e = 14$.

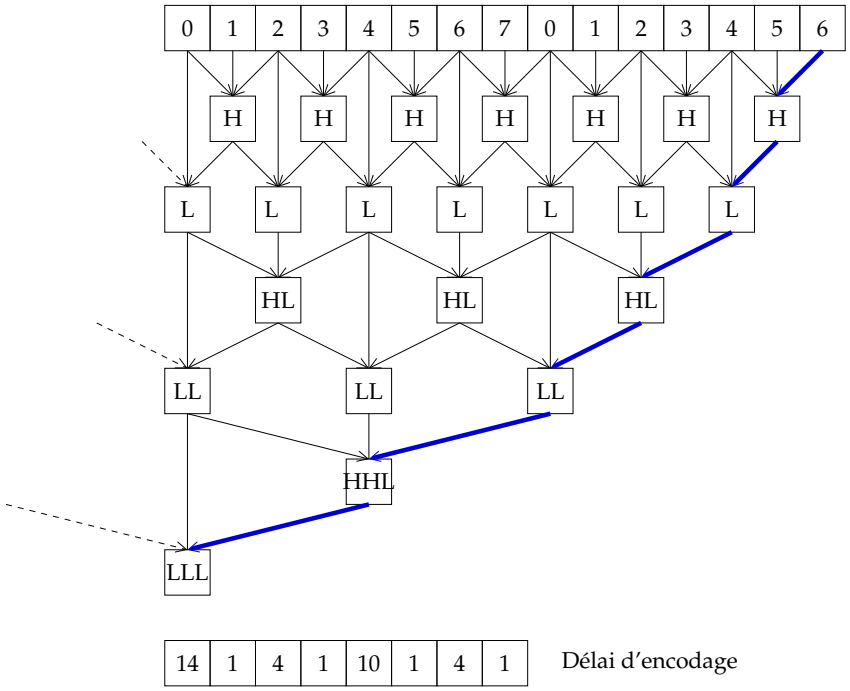


FIG. 4.16 – Délai maximal d'encodage d'une trame dans une analyse temporelle 5/3 à 3 niveaux. Le chemin en gras désigne le chemin de traitement lié à l'image ayant le plus grand délai d'encodage $N_e = 14$.

Délai de décodage et de reconstruction

Le délai de décodage $D_d = N_d \times f$ est la durée maximale nécessaire à la transformation inverse d'une sous-bande en sa trame correspondante. C'est donc le nombre maximal N_d de sous-bandes situées dans le futur nécessaire à la reconstruction de l'image courante.

Le délai de reconstruction $D_r = N_r \times f$ est la durée maximale nécessaire à la transformation d'une image courante en sa sous-bande correspondante. C'est donc le nombre maximal N_r d'images situées dans le futur nécessaire à la transformation *et* à la reconstruction de l'image courante. Le délai de reconstruction n'est pas égal à la somme du délai d'encodage D_e et du délai de décodage D_d car l'image possédant le délai maximal d'encodage n'est pas nécessairement celle qui possède le délai maximal de décodage. Il est aussi appelé délai point à point (*End-to-end delay*) dans la littérature. Ce délai a une signification précise dans le cadre d'une application de vidéoconférence en temps réel car il caractérise précisément le temps nécessaire à une image capturée du côté émetteur pour pouvoir être reconstruite par le récepteur.

Nous avons représenté sur la Fig. 4.17 une analyse temporelle de type 5/3 sur 3 niveaux où les délais maximaux N_d et N_r sont représentés. Les délais de décodage et de reconstruction de chaque image sont aussi indiqués, exprimés en nombre d'images N. L'image d'indice 0 nécessite ainsi 4 sous-bandes pour être reconstruite et donc 14 images futures vues du côté encodeur pour pouvoir être reconstruite. Le chemin en gras désigne

le chemin de traitement lié à l'image ayant le plus grand délai de décodage $N_d = 11$ et de reconstruction $N_r = 21$.

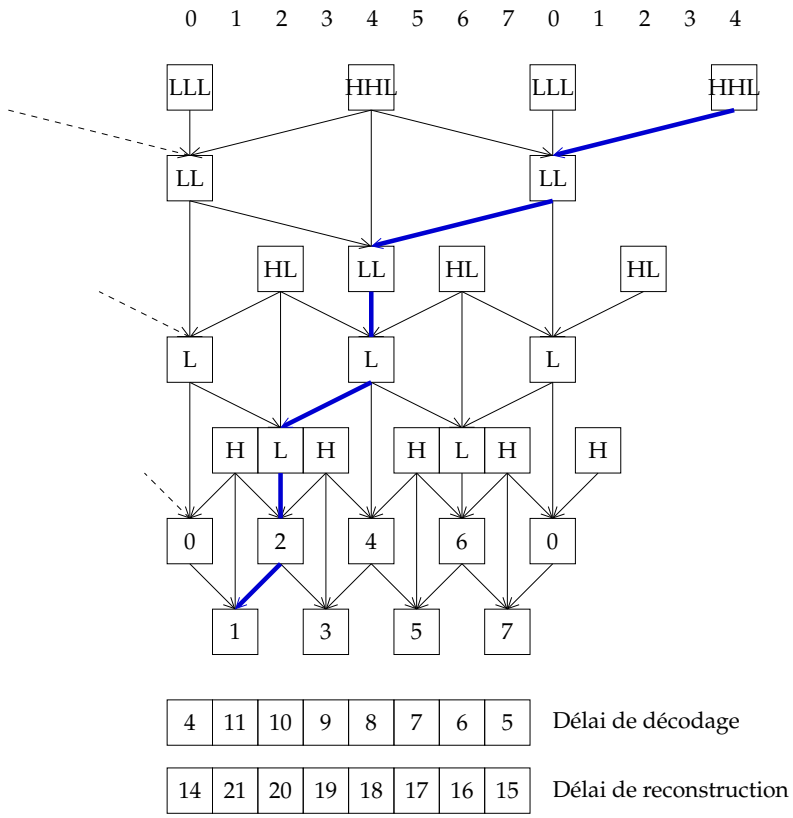


FIG. 4.17 – Délais maximaux de décodage et de reconstruction d'une trame dans une synthèse temporelle 5/3 à 3 niveaux. Le chemin en gras désigne le chemin de traitement lié à l'image ayant le plus grand délai de décodage $N_d = 11$ et de reconstruction $N_r = 21$.

En considérant une analyse temporelle sur N niveaux, il est possible de calculer les délais d'encodage, de décodage et de reconstruction introduits par divers filtres temporels. Nous nous intéressons dans les sections suivantes aux calculs de ces délais créés par les transformées temporelles les plus couramment utilisées.

Filtre temporel de Haar

Le filtre temporel de Haar est décrit par les équations suivantes :

$$h_t = x_{2t+1} - \mathcal{C}(x_{2t}, \boldsymbol{v}^+_{2t+1})$$
$$l_t = x_{2t} + \frac{1}{2}\mathcal{C}^{-1}(h_t, \boldsymbol{v}^+_{2t+1})$$

On peut montrer qu'une analyse temporelle de Haar sur N niveaux engendre alors les délais :

$$\begin{cases} N_e = 2^N - 1 \\ N_d = 2^{N-1} \\ N_r = 2^N - 1 \end{cases}$$

La suppression de l'étape de mise à jour du filtre de Haar conduit à un filtre purement causal, n'introduisant aucun délai $N_e = N_d = N_r = 0$. Cependant, comme vu précédemment dans la section. 3.2 du chapitre précédent, l'efficacité du filtre de Haar n'est pas satisfaisante. Nous verrons par la suite qu'il existe des alternatives plus intéressantes pour obtenir des filtres à délai faible ou même nul.

Filtre temporel 5/3

Nous rappelons les équations de filtrage temporel 5/3 :

$$h_t = x_{2t+1} - \frac{1}{2}\big(\mathcal{C}(x_{2t}, \boldsymbol{v}^+_{2t+1}) + \mathcal{C}(x_{2t+2}, \boldsymbol{v}^-_{2t+1})\big)$$
$$l_t = x_{2t} + \frac{1}{4}\big(\mathcal{C}^{-1}(h_{t-1}, \boldsymbol{v}^-_{2t-1}) + \mathcal{C}^{-1}(h_t, \boldsymbol{v}^+_{2t+1})\big)$$

Il est possible de montrer qu'une décomposition temporelle sur N niveaux, selon le filtre 5/3 classique ou selon sa variante 5/3 uniforme décrit dans la section 4.2, introduit les délais suivants :

$$\begin{cases} N_e = 2^{N+1} - 2 \\ N_d = 3 \times 2^{N-1} - 1 \\ N_r = 3 \times (2^N - 1) \end{cases}$$

Ainsi, l'utilisation d'une transformée 5/3 sur 4 niveaux temporels dans une application de visioconférence induirait un délai de reconstruction d'au minimum $D_r = N_r \times f = 1.5$ s ! Le seuil de confort visuel étant d'environ 300 ms, ce délai est ainsi bien trop important.

On remarque que la seule marge d'action pour réduire le délai dans une décomposition 5/3 consiste à diminuer le nombre de niveaux N de l'analyse temporelle, entraînant par conséquence une forte diminution des performances du codeur vidéo. Ceci motive ainsi la nécessité de construire un filtre avec une latence moindre.

De plus, on notera que la transformée temporelle 5/3 sans mise à jour utilisée par André [13] possède des délais plus faibles que la transformée 5/3 classique :

$$\begin{cases} N_e = 2^{N-1} \\ N_d = 2^N - 1 \\ N_r = 2^N - 1 \end{cases}$$

Comparé au filtre 5/3 classique, on observe une réduction d'un facteur 4 sur N_e, d'un facteur 3/2 sur N_d et d'un facteur 3 sur N_r. Bien qu'importante, cette diminution du délai reste toutefois insuffisante et ne permet pas d'assurer une latence compatible avec une application de type visioconférence.

En étendant le cas des filtres temporels dyadiques présentés ci-dessus, il est possible de calculer les délais introduits par les filtres temporels 3-bandes de Tillier [143, 144], dont un des intérêts réside dans leur aptitude à fournir des facteurs de scalabilité d'ordre 3.

Filtres temporels 3-bandes

Le filtre temporel 3-bandes le plus simple [143] est l'équivalent 3-bandes du filtre de Haar. Il est monodirectionnel, créé deux sous-bandes de détail et s'exprime sous la forme

lifting suivante :

$$\begin{aligned} h_t^+ &= x_{3t+1} - \mathcal{C}(x_{3t}, \boldsymbol{v}_{3t+1}^+) \\ h_t^- &= x_{3t-1} - \mathcal{C}(x_{3t}, \boldsymbol{v}_{3t-1}^+) \\ l_t &= x_{3t} + \frac{1}{4}\big(\mathcal{C}^{-1}(h_t^+, \boldsymbol{v}_{3t+1}^+) + \mathcal{C}^{-1}(h_t^-, \boldsymbol{v}_{3t-1}^+)\big) \end{aligned}$$

L'analyse temporelle sur N niveaux par un filtre Haar 3-bandes introduit les délais :

$$\begin{cases} N_e = (3^N - 1)/2 \\ N_d = (3^N - 1)/2 \\ N_r = (3^N - 1)/2 \end{cases}$$

L'introduction d'une prédiction bidirectionnelle permet alors d'obtenir un équivalent 3-bandes du filtre temporel 5/3 [144], s'exprimant par :

$$\begin{aligned} h_t^+ &= x_{3t+1} - \frac{1}{2}\big(\mathcal{C}(x_{3t}, \boldsymbol{v}_{3t+1}^+) + \mathcal{C}(x_{3t+2}, \boldsymbol{v}_{3t+1}^-)\big) \\ h_t^- &= x_{3t-1} - \frac{1}{2}\big(\mathcal{C}(x_{3t}, \boldsymbol{v}_{3t-1}^+) + \mathcal{C}(x_{3t-2}, \boldsymbol{v}_{3t-1}^-)\big) \\ l_t &= x_{3t} + \frac{1}{4}\big(\mathcal{C}^{-1}(h_t^-, \boldsymbol{v}_{3t+1}^-) + \mathcal{C}^{-1}(h_t^+, \boldsymbol{v}_{3t-1}^+)\big) \end{aligned}$$

Ce filtre engendre les délais suivants :

$$\begin{cases} N_e = 3^N - 1 \\ N_d = (3^N - 1)/2 \\ N_r = 3^N - 1 \end{cases}$$

Loin de réduire les délais de la transformée temporelle 5/3, ces filtres 3-bandes introduisent des retards nettement supérieurs à leurs homologues dyadiques pour un même nombre de niveaux temporels. Ils ne peuvent donc satisfaire nos contraintes de latence.

4.3.3 Construction d'un filtre temporel flexible à délai contraint

Plusieurs solutions comme la suppression drastique de l'étape de mise à jour [157] ou la combinaison étagée de plusieurs filtres temporels (5/3 et Haar) ont été proposées [53] mais se révèlent insuffisantes. En effet, elles permettent de réduire le délai mais ne sont pas suffisamment flexibles pour le diminuer à dessein voire l'annuler complètement.

Nous avons proposé un compromis simple [104] pour modérer ce retard à l'encodage, conduisant à un compromis entre efficacité de codage et contrainte de délai. Il consiste à supprimer la partie "en avant" des opérateurs de prédiction et de mise à jour des niveaux temporels les plus élevés et peut s'appliquer à n'importe quelle transformation temporelle. Cependant, ces travaux ne concernaient que la réduction du délai d'encodage et l'on observait une chute de performance importante en présence de contraintes de latence trop élevées. Afin de résoudre ces problèmes, nous avons poursuivi nos travaux et avons présenté [107] une transformée temporelle étagée composée de transformées élémentaires, afin d'obtenir un compromis souple entre latence et efficacité de codage.

Analyse temporelle 5/3 à délai contraint

Considérons les trois transformées élémentaires $T1$, $T2$ et $T3$ suivantes. La transformée $T1$ est la transformée $5/3$ classique ; elle possède la meilleure efficacité de codage mais introduit le plus grand retard. Elle est définie par :

$$\begin{aligned} h_t &= x_{2t+1} - \frac{1}{2}\big(\mathcal{C}(x_{2t}, \boldsymbol{v}^+_{2t+1}) + \mathcal{C}(x_{2t+2}, \boldsymbol{v}^-_{2t+1})\big) \\ l_t &= x_{2t} + \frac{1}{4}\big(\mathcal{C}^{-1}(h_{t-1}, \boldsymbol{v}^-_{2t-1}) + \mathcal{C}^{-1}(h_t, \boldsymbol{v}^+_{2t+1})\big) \end{aligned} \tag{T1}$$

Afin de réduire le retard introduit par la transformée élémentaire $T1$ et tout en gardant de bonnes propriétés de décorrélation, nous considérons une transformée $5/3$ dégénérée sans mise à jour en avant. En effet, nous avons mis en évidence dans la section 4.2.4 que la mise à jour n'a qu'une influence limitée sur l'efficacité de codage. Notons cette transformée $T2$:

$$\begin{aligned} h_t &= x_{2t+1} - \frac{1}{2}\big(\mathcal{C}(x_{2t}, \boldsymbol{v}^+_{2t+1}) + \mathcal{C}(x_{2t+2}, \boldsymbol{v}^-_{2t+1})\big) \\ l_t &= x_{2t} + \frac{1}{2}\mathcal{C}^{-1}(h_{t-1}, \boldsymbol{v}^-_{2t-1}) \end{aligned} \tag{T2}$$

La transformée $T2$ possède un retard deux fois moindre que la transformée $T1$ mais ne peut engendrer un retard nul. Pour atteindre ce but, nous devons continuer notre dégradation de la transformée $5/3$ et introduire alors une transformée sans prédiction en avant. De plus, comme le délai introduit par la prédiction en avant est plus important que celui induit par la mise à jour avant, nous devons supprimer la mise à jour avant. Le champ $\boldsymbol{v}^-_{2t+1}$ n'est alors utilisé que pour réaliser la mise à jour arrière, que nous décidons de supprimer pour économiser le coût de codage du champ et améliorer ainsi l'efficacité de codage. On obtient au final la transformée $T3$ suivante, qui est une forme dégénérée de la transformée de Haar sans mise à jour :

$$\begin{aligned} h_t &= x_{2t+1} - \mathcal{C}(x_{2t}, \boldsymbol{v}^+_{2t+1}) \\ l_t &= x_{2t} \end{aligned} \tag{T3}$$

Nous considérons alors une transformée temporelle étagée paramétrée par (P, Q) qui consiste à appliquer la transformée $T1$ sur les P premiers niveaux temporels, la transformée $T2$ sur les Q suivants et la transformée $T3$ sur les niveaux restants. Cette analyse temporelle que nous nommerons transformée (P, Q), introduit alors les délais suivants :

$$\text{Si } Q = 0 \begin{cases} N_e = 2^{P+1} - 2 \\ N_d = \lfloor 3 \times 2^{P-1} - 1 \rfloor \\ N_r = 3 \times (2^P - 1) \end{cases}$$

$$\text{Si } Q > 0 \begin{cases} N_e = 2^{P+1} + 2^{P+Q-1} - 2 \\ N_d = 2^{P+Q} - 1 \\ N_r = 2^{P+1} + 2^{P+Q} - 3 \end{cases}$$

Munis de ces relations, nous pouvons agir sur les paramètres P et Q de la transformée pour satisfaire un délai donné. Sachant qu'il peut y avoir plusieurs couples solutions satisfaisant un délai donné, nous choisirons les paramètres qui maximisent le nombre d'étapes de prédictions bidirectionnelles, c'est à dire ceux qui maximisent $P + Q$. On

pourra remarquer que la transformée $(P, Q) = (N, 0)$ correspond à une transformée 5/3 classique sur N niveaux et que la transformée $(P, Q) = (0, 0)$ est une analyse temporelle de Haar sans mise à jour, à délai nul. On notera de plus que les délais engendrés par la transformée (P, Q) sont indépendants du nombre de niveaux temporels N. Nous pouvons alors établir le Tab. 4.15 donnant les paramètres optimaux de la transformée (P, Q) permettant de satisfaire une contrainte de délai de reconstruction donnée.

Délai maximal de reconstruction	N_r	(P, Q) optimal
1500 ms	45	(4,0)
500 ms	15	(0,4)
300 ms	9	(1,2)
167 ms	5	(1,1)
100 ms	3	(0,2)
34 ms	1	(0,1)
0 ms	0	(0,0)

TAB. 4.15 – Paramètres optimaux (P, Q) pour satisfaire un délai de reconstruction pour une décomposition temporelle de 4 niveaux à la fréquence $f = 30$ Hz.

Ce tableau se lit comme suit : par exemple, pour obtenir une transformée (P, Q) ayant un délai de reconstruction maximal de 167 ms, valeur couramment utilisée dans les applications de visioconférence, il faut utiliser le couple de paramètres $(P, Q) = (1, 1)$. Ce choix correspond à une transformation étagée utilisant la transformée élémentaire $T1$ pour le premier niveau, la transformée $T2$ pour le second et enfin la transformée $T3$ pour les niveaux restants. Les structures correspondantes à la décomposition temporelle et à la reconstruction sur 3 niveaux sont illustrées par les Figs. 4.18 et 4.19 et sont à comparer avec les Figs.4.16 et 4.17, obtenues avec la transformée temporelle 5/3 classique. Par construction et comme attendu, on remarque que le délai de reconstruction n'est jamais supérieur à 5 trames.

4.3.4 Résultats expérimentaux

Nous avons construit dans la section précédente la transformée paramétrable (P, Q) à délai contraint. Cette réduction de délai a été rendue possible par l'utilisation de transformées élémentaires 5/3 dégénérées sans mise à jour avant ou mono-directionnelles, d'une efficacité moindre que la transformée 5/3. Il y a donc un compromis clair entre efficacité de codage et délai.

Afin d'apprécier les performances de la transformée temporelle (P, Q) en fonction d'une contrainte de délai, nous avons effectué des simulations de codage sur les séquences *Football* et *Tempête*. Les tableaux Tab. 4.16 et 4.17 présentent ces résultats exprimés en Y-PSNR, obtenus pour un ensemble de contraintes de délais et de débits.

Conformément aux attentes exprimées dans la section précédente, on observe que les meilleures performances sont atteintes dans le cas de la transformée 5/3 non-contrainte, à délai supérieur à 1500 ms. On note ensuite une dégradation légère en fonction du délai imposé, avec une perte de seulement 0.6 dB en présence d'un délai maximal de 150 ms. Cette dégradation est plus importante pour *Tempête* où la souplesse du mouvement pâtit de la perte des opérateurs de prédiction bidirectionnels dans les bas délais.

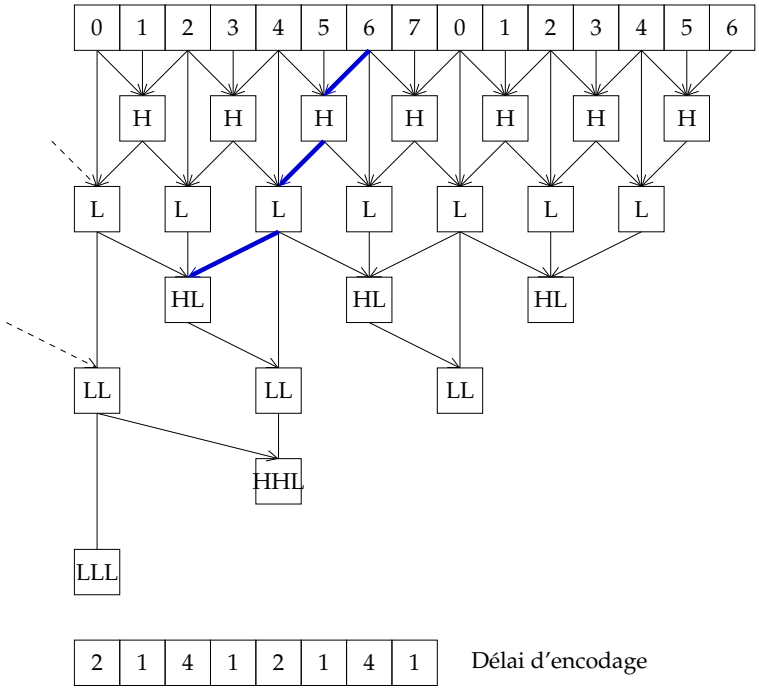


FIG. 4.18 – Délai maximal d'encodage d'une trame dans l'analyse temporelle à 3 niveaux par la transformée $(P, Q) = (1, 1)$.

Délai maximal	384 kbs	512 kbs	768 kbs	1024 kbs
1500 ms	27.15	28.49	30.32	31.79
300 ms	26.83	28.11	29.92	31.43
150 ms	26.51	27.68	29.44	30.90
35 ms	26.23	27.39	29.15	30.61
0 ms	25.94	27.08	28.79	30.30

TAB. 4.16 – Mesures de débit-distorsion obtenues avec la transformée temporelle (P, Q) pour plusieurs contraintes de délai de reconstruction et pour différents débits sur la séquence *Football* CIF 30 Hz.

Délai maximal	384 kbs	512 kbs	768 kbs	1024 kbs
1500 ms	29.72	31.00	32.77	33.95
300 ms	28.65	29.97	31.65	32.98
150 ms	27.90	29.23	30.84	32.20
35 ms	27.35	28.65	30.23	31.61
0 ms	26.91	28.11	29.64	31.02

TAB. 4.17 – Mesures de débit-distorsion obtenues avec la transformée temporelle (P, Q) pour plusieurs contraintes de délai de reconstruction et pour différents débits sur la séquence *Tempête* CIF 30 Hz.

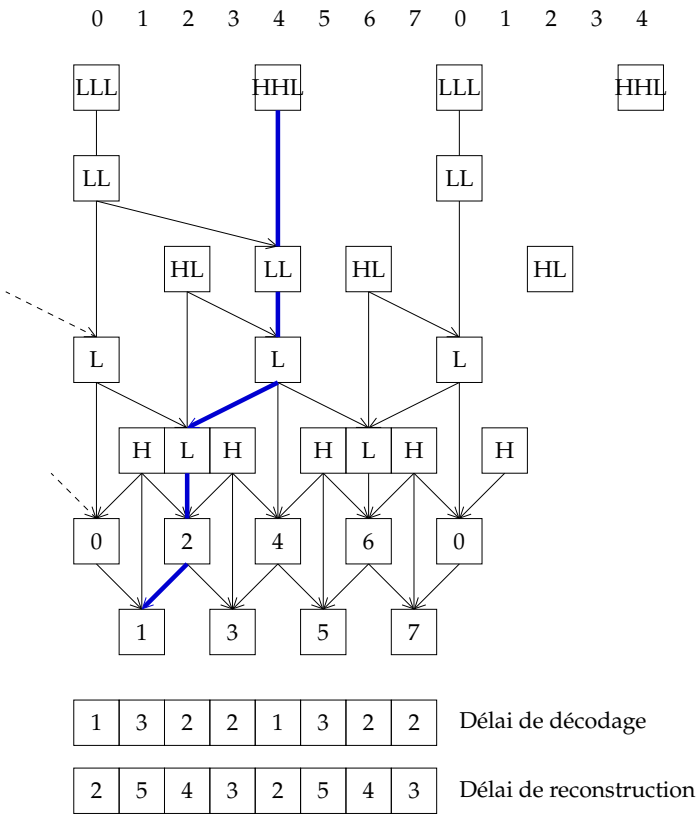


FIG. 4.19 – Délais maximaux de décodage et de reconstruction d'une trame dans la synthèse à 3 niveaux par la transformée $(P, Q) = (1, 1)$.

La transformée temporelle (P, Q) à délai contraint a de plus été implémentée conjointement par Viéron [158] au sein du codec SVC du groupe de normalisation MPEG et a donné des résultats similaires tout à fait satisfaisants. D'autres travaux ultérieurs sur la réduction du retard introduite par la transformée temporelle dans le codec SVC ont aussi été menés dans [124]. Les auteurs proposent de partitionner la décomposition d'un GOP en sous-ensembles indépendants, en contraignant chaque image de ces ensembles à ne jamais nécessiter plus de N images en avant pour être décomposée. Tout comme notre technique, cette proposition revient à couper les dépendances en avant lors de la décomposition temporelle. Cependant, elle possède l'inconvénient de créer des structures de prédiction irrégulières où, sur un même niveau temporel, les mêmes trames peuvent être prédites par une ou deux images et être mises à jour ou pas.

4.3.5 Conclusion

Nous avons présenté une transformée temporelle flexible, dont le délai est paramétrable en fonction des besoins d'une application. Les résultats expérimentaux ont montré l'existence d'un compromis intéressant entre délai et efficacité de codage. Ils ont de plus mis en évidence la dégradation modérée de l'efficacité de codage en fonction du délai, en partant du cas non-contraint au cas de délai nul. Entre ces deux extrêmes, un large éventail de compromis entre l'efficacité souhaitée et le délai maximal admissible existe, laissant le choix du filtre en fonction des besoins de l'application. Il est à noter que les mé-

thodes présentées ont été utilisées afin de réduire la latence des filtres temporels mis en jeu dans le schéma de codage $t+2D$. Cependant, des perspectives envisageables consisteraient à utiliser ces mêmes techniques pour abaisser le nombre total d'images nécessaires à la décomposition temporelle, afin de réduire la taille des mémoires tampons du codec. On pourrait par exemple considérer le nombres d'images dans le futur et *dans le passé*, nécessaires pour transformer une image courante, en raisonnant sur la taille du nombre d'images du tampon cyclique utilisé dans l'implémentation au fil de l'eau de la transformée temporelle.

4.4 Transformée Daubechies-4 compensée en mouvement

Nous avons vu dans la section 2.2.3 que les premiers codeurs vidéos scalables $t + 2D$ utilisaient une transformée temporelle de Haar par souci de simplicité. L'introduction du lifting temporel par Pesquet-Popescu [108] a alors permis l'utilisation de n'importe quelle décomposition temporelle même non-linéaire, tout en garantissant son inversibilité. De nombreuses transformées basées sur le filtre temporel $5/3$ compensé en mouvement ont pu alors être mises en œuvres et ont montré une efficacité de codage supérieure à la transformée de Haar.

Cependant, mis à part les schémas purement prédictifs UMCTF de Turaga [151], il n'existe pas dans la littérature de décomposition temporelle compensée en mouvement basée sur un filtre autre que celui de Haar ou que le filtre $5/3$. Le fait est singulier car il a été montré en codage d'image la nette supériorité de la transformée $9/7$ sur la transformée $5/3$. Est-il raisonnable de penser que l'augmentation de la taille du support d'un filtre temporel puisse améliorer son efficacité de codage ? C'est pour tenter de répondre à cette question que nous nous proposons dans cette section de mettre en œuvre une transformée temporelle compensée en mouvement basée sur l'ondelette Daubechies-4 .

4.4.1 Description et mise en œuvre

Nous souhaitons construire une transformée temporelle de support plus long que l'ondelette $5/3$ mais il n'est pas aisé de construire explicitement une structure lifting *ad-hoc* à trois étage. Nous nous proposons d'utiliser alors des structures déjà existantes, correspondant à des familles d'ondelettes connues.

Il existe au moins deux transformées en ondelettes dont la structure en lifting possède 3 étages : l'ondelette Daubechies-4 et l'ondelette biorthogonale $7/5$. La transformée en ondelettes Daubechies-4 est orthogonale, possède 2 moments nuls et un support de 4 points alors que la transformée CDF $7/5$ est biorthogonale, symétrique et possède 3 moments nuls. Poursuivant notre optique d'étudier le comportement d'une transformée temporelle de support un peu plus large que l'ondelette $5/3$, nous avons opté pour l'ondelette Daubechies-4. En effet, l'ondelette $7/5$ possède un support peut-être un peu trop grand pour une première approche et ceci peut être nuisible à la qualité de la prédiction temporelle en présence d'un mouvement trop rapide dans une séquence vidéo.

La transformation en ondelettes Daubechies-4 d'un signal mono-dimensionnel x_t peut s'exprimer sous forme lifting au moyen de trois opérateurs : un opérateur de prédiction P1, une mise à jour U et un autre opérateur de prédiction P2. Une mise à l'échelle des sous-bandes est alors effectuée par les étapes S1 et S2. La transformée de Daubechies-4

est illustrée en Fig. 4.20 et s'exprime sous forme lifting par les équations suivantes :

$$h_t^0 = x_{2t+1} - \sqrt{3}x_{2t} \tag{P1}$$

$$l_t^0 = x_{2t} + \frac{\sqrt{3}}{4}h_t^0 + \frac{\sqrt{3}-2}{4}h_{t+1}^0 \tag{U}$$

$$h_t^1 = h_t^0 + l_{t-1}^0 \tag{P2}$$

$$h_t = \frac{\sqrt{3}-1}{\sqrt{2}}h_t^1 \tag{S1}$$

$$l_t = \frac{\sqrt{3}+1}{\sqrt{2}}l_t^0 \tag{S2}$$

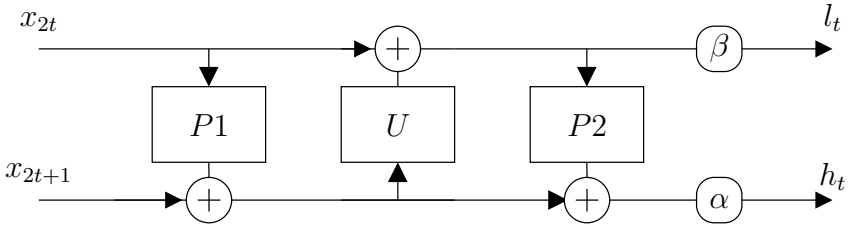


FIG. 4.20 – Structure lifting de la transformée en ondelettes de Daubechies-4

Le cadre général du lifting temporel développé par Pesquet-Popescu [108], rappelé dans la section 3.1.2, nous permet aisément de construire une transformée temporelle compensée en mouvement à partir de la formulation lifting d'une transformée monodimensionnelle. L'utilisation de l'opérateur de compensation de mouvement $\mathcal{C}$ introduit dans la section 3.1.3 nous permet alors de construire une transformée temporelle basée sur l'ondelette Daubechies-4 et s'exprimant sous la forme :

$$h_t^0 = x_{2t+1} - \sqrt{3}\,\mathcal{C}(x_{2t}, \boldsymbol{v}_0) \tag{P1}$$

$$l_t^0 = x_{2t} + \frac{\sqrt{3}}{4}\mathcal{C}(h_t^0, \boldsymbol{v}_1) + \frac{\sqrt{3}-2}{4}\mathcal{C}(h_{t+1}^0, \boldsymbol{v}_2) \tag{U}$$

$$h_t^1 = h_t^0 + \mathcal{C}(l_{t-1}^0, \boldsymbol{v}_3) \tag{P2}$$

$$h_t = \frac{\sqrt{3}-1}{\sqrt{2}}h_t^1 \tag{S1}$$

$$l_t = \frac{\sqrt{3}+1}{\sqrt{2}}l_t^0 \tag{S2}$$

La Fig. 4.21 illustre la décomposition d'un extrait de séquence vidéo sur un niveau en utilisant cette transformée temporelle. Comme on peut l'apercevoir, elle met en jeu quatre champs de vecteurs mouvement $\boldsymbol{v}_0$, $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ et $\boldsymbol{v}_3$. Tout comme dans le cas des filtres de Haar ou 5/3 et comme abordé dans la section 3.1.2, il n'est pas souhaitable de les conserver tous. En effet, ils ont un coût de codage non-négligeable et il est préférable d'en estimer certains et en déduire les autres. La première étape de prédiction $P1$ nécessite un champ de mouvement avant $\boldsymbol{v}_0 = v_{2t+1}^+$, prédisant l'image x_{2t+1} par rapport à l'image x_{2t}. La mise à jour U met en jeu les deux champs $\boldsymbol{v}_1$ et $\boldsymbol{v}_2$. Le champ de mouvement $\boldsymbol{v}_1$ est l'opposé du champ $\boldsymbol{v}_0$ et peut se calculer par inversion en utilisant l'opérateur de compensation inverse $\mathcal{C}^{-1}$, comme dans le cas des filtres de Haar ou 5/3. Le champ de

mouvement arrière $\boldsymbol{v}_2$ prédit l'image x_{2t+3} par rapport à l'image x_{2t}, sur une distance de trois images. Ce champ est aussi l'opposé du champ $\boldsymbol{v}_3$, mis en jeu dans la prédiction $P2$: le champ $\boldsymbol{v}_3$ peut donc être calculé par inversion de $\boldsymbol{v}_2$.

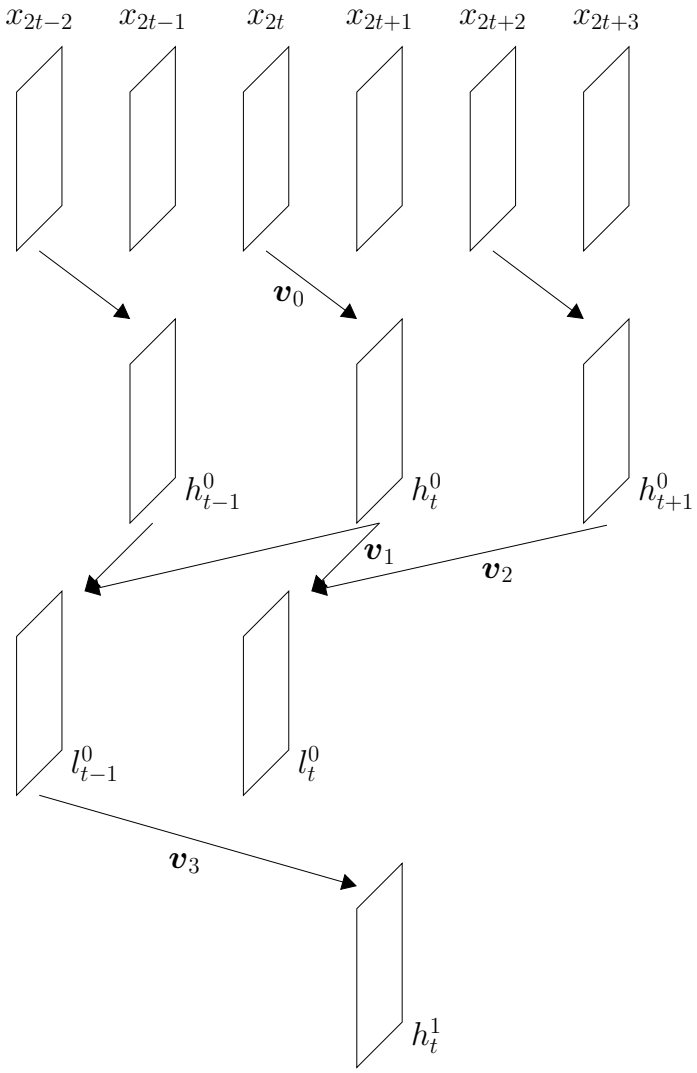


FIG. 4.21 – Décomposition temporelle en ondelettes Daubechies-4

Plusieurs stratégies sont donc possibles sur le choix des champs de mouvement mis en jeu dans la transformée de Daubechies-4. Afin de réduire la redondance de ces champs et de ne pas augmenter la complexité de notre prototype qui ne gère qu'au maximum deux champs de mouvement, nous choisissons de ne considérer que les champs avant $\boldsymbol{v}_0 = \boldsymbol{v}^1_{2t+1}$ et $\boldsymbol{v}_3 = \boldsymbol{v}^2_{2t+1}$, mis en jeu dans les prédictions $P1$ et $P2$. Seuls ces champs seront alors estimés et encodés dans le bitstream vidéo compressé. L'opérateur de compensation inverse $\mathcal{C}^{-1}$ permet d'obtenir les autres champs de mouvement $\boldsymbol{v}_1$ et $\boldsymbol{v}_2$ par inversion des champs $\boldsymbol{v}_0$ et $\boldsymbol{v}_3$, respectivement. L'opérateur de mise à jour de la transformée temporelle Daubechies-4 se réécrit alors :

$$l^0_t = x_{2t} + \frac{\sqrt{3}}{4}\mathcal{C}^{-1}(h^0_t, \boldsymbol{v}_0) + \frac{\sqrt{3}-2}{4}\mathcal{C}^{-1}(h^0_{t+1}, \boldsymbol{v}_3) \qquad \text{(U)}$$

La mise en œuvre efficace de la transformée temporelle Daubechies-4 nécessite une implémentation au fil de l'eau comme abordé dans la section 3.1.4 où les images sont décomposées et transformées à la volée. Cette implémentation repose sur un module où un buffer cyclique consomme deux nouvelles images et produit deux sous-bandes temporelles. Le module effectue alors un cycle en accomplissant les étapes décrites dans la Fig. 4.22, de façon similaire au module réalisant la transformée temporelle 5/3 de la Fig. 3.6.

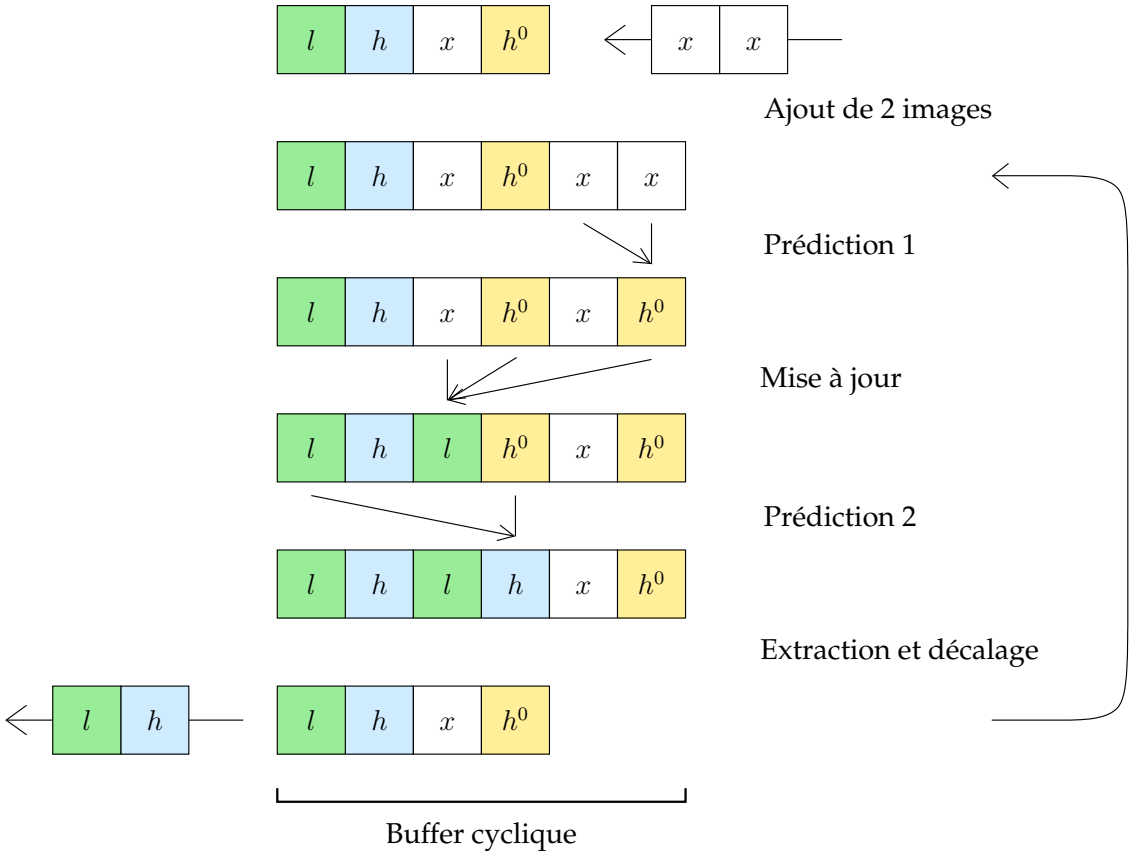


FIG. 4.22 – Schéma de fonctionnement du module de traitement au fil de l'eau de la transformée temporelle Daubechies-4. On y observe l'évolution du buffer cyclique de filtrage.

4.4.2 Résultats expérimentaux

Afin d'évaluer son efficacité de codage vidéo, nous avons intégré la transformée temporelle Daubechies-4 (D4) au sein de notre prototype sous la forme modulaire au fil de l'eau décrite précédemment. Les simulations ont été effectuées sur les séquences *Mobile* et *Foreman* CIF 30 Hz, en utilisant 3 niveaux de décompositions temporelles. Les résultats de simulation sont présentés dans les Tabs. 4.18 et 4.19, où ils sont comparés à des expérimentations de codage en conditions similaires effectuées avec les filtres temporels de Haar et 5/3.

On observe la performance médiocre réalisée par le filtre temporel D4. En effet, les résultats de codage observés restent inférieurs à ceux obtenus avec le filtre de Haar ou

YSNR (en dB)	384 kbs	512 kbs	768 kbs	1024 kbs	2048 kbs
5/3	25.30	27.32	**29.85**	**31.45**	**35.46**
Haar	**25.73**	**27.38**	29.66	31.15	35.06
D4	18.33	19.14	21.15	22.53	27.08

TAB. 4.18 – Mesures de distorsion obtenues en utilisant différents filtres temporels à différents débits sur la séquence *Mobile* CIF 30 Hz.

YSNR (en dB)	384 kbs	512 kbs	768 kbs	1024 kbs	2048 kbs
5/3	**32.76**	**33.85**	**35.27**	**36.50**	**39.61**
Haar	32.21	33.23	34.65	35.85	38.93
D4	22.09	23.02	24.19	25.45	28.74

TAB. 4.19 – Mesures de distorsion obtenues en utilisant différents filtres temporels à différents débits sur la séquence *Foreman* CIF 30 Hz.

le filtre 5/3. Il est possible que ce manque d'efficacité soit expliqué par les nombreux champs de mouvement nécessités par la transformation D4. Deux inversions de champs sont ainsi nécessaires pour effectuer l'opération de mise à jour, qui n'a pas la même signification que sa consœur dans le filtre 5/3. En effet et contrairement au filtre 5/3, l'opérateur U ajoute ici à la sous-bande l_t^0 les sous-bandes h_t^0 qui sont loin d'être des images peu énergétiques. Les erreurs de mouvement créées par la compensation inverse lors de la mise à jour ont alors une influence importante en créant de larges coefficients sur les sous-bandes de détail. Le surcoût nécessaire au codage de ces coefficients dégrade ainsi le rapport signal à bruit et diminue l'efficacité de la transformée temporelle D4. Enfin, la forte dissymétrie de l'ondelette Daubechies-4 est peut-être en cause dans le manque d'efficacité de la transformée temporelle D4. D'autres configurations de mouvement, l'utilisation d'un nombre supérieur de champs ou la mise en œuvre de la transformée 7/5 pourraient être envisagés afin d'augmenter l'efficacité de codage.

4.5 Conclusion

Nous avons présenté dans ce chapitre plusieurs stratégies d'optimisation de la transformée temporelle mise en jeu dans le schéma de codage $t + 2D$. En se basant sur sa structure lifting, nous avons poursuivi plusieurs axes de recherche visant à améliorer son efficacité de décorrélation temporelle.

Nous avons tout d'abord proposé un algorithme quasi-optimal d'estimation bidirectionnelle conjointe des champs de mouvement mis en jeu dans la transformée temporelle 5/3 compensée en mouvement. Cet algorithme est itératif, converge rapidement et permet la minimisation de la distorsion des sous-bandes temporelles de détail. Sa mise en place au sein du codec MC-EZBC conduit à un gain moyen en PSNR de plus de 1 dB par rapport à une estimation indépendante des champs de mouvement, pour une complexité équivalente.

Un inconvénient majeur de la transformée temporelle 5/3 est sa propension à créer des artefacts fantômes dans les séquences décodées à bas débits. Ces artefacts sont visuellement désagréables, complexifient le codage des images et sont liés à la présence de zones non-connectées durant l'étape de mise à jour temporelle. Afin de supprimer ces

artefacts, nous avons proposé une transformée 5/3 uniforme où les champs de mouvement sont orientés dans le même sens, empêchant la création de zones non-connectées. Comme précédemment, il est alors possible de concevoir un algorithme d'estimation bidirectionnelle conjoint des champs de mouvement mis en jeu dans la transformée 5/3 uniforme. Sa mise en œuvre expérimentale permet une réduction visible des artefacts présents dans les sous-bandes temporelles d'approximation et offre une efficacité de codage supérieure à la transformée 5/3 optimisée, surpassant même dans certains cas le codec H.264, pourtant non-scalable.

Les filtres temporels classiques introduisent un retard important dans le schéma de codage vidéo $t+2D$, prohibant leur utilisation pour des applications de visioconférence en temps réel. Après avoir étudié les causes de cette latence, nous avons présenté une transformée temporelle flexible basée sur le filtre 5/3, capable de respecter une contrainte de délai imposée. Elle consiste en la construction d'une analyse temporelle utilisant trois types de filtres temporels élémentaires. Les résultats expérimentaux montrent une faible dégradation du PSNR en fonction du délai imposé et concluent sur l'existence d'un compromis entre le délai imposé et l'efficacité de codage obtenue. Cette transformée flexible offre ainsi une large plage de possibilités, s'étalant du cas non-contraint au cas de délai nul, en fonction des besoins de l'application.

Enfin, conscients du gain important en efficacité de codage apporté par la transformée temporelle 5/3 comparée à la transformée de Haar, nous avons souhaité expérimenter un filtre compensé en mouvement plus long, basé sur l'ondelette Daubechies-4. Sa mise en œuvre au sein d'un schéma de codage $t+2D$ est facilitée par l'utilisation de la structure lifting et une implémentation au fil de l'eau. La transformée temporelle Daubechies-4 montre cependant une efficacité de codage inférieure aux filtres de Haar et 5/3, due probablement à une gestion complexe des champs de mouvement.

Chapitre 5

Bancs de filtres M-bandes et filtrage spatial

Les transformées en bancs de filtres M-bandes sont obtenues par extension du nombre de bandes mises en jeu dans un banc de filtres dyadique. De part leurs performances de décorrélation, leur flexibilité et leur grande sélectivité fréquentielle, certaines d'entres elles comme la DCT ou les transformées orthogonales à recouvrement sont très utilisées en compression d'image fixe. Nous nous intéressons dans ce chapitre à l'étude de leurs propriétés, à leur mise en œuvre dans le cadre de notre schéma de codage vidéo et nous présenterons alors un moyen d'étendre leurs propriétés de scalabilité.

Après avoir rappelé la définition des transformées M-bandes et énoncé quelques unes de leurs propriétés dans la section 5.1, nous décrivons les transformées de cette famille les plus communément utilisées en codage d'image fixe et en codage vidéo. Nous verrons alors comment leurs propriétés de sélectivité fréquentielle et leur généricité peuvent être mises à profit dans le cadre de notre schéma de codage vidéo.

Tout au long du chapitre précédent, nous avons étudié comment les images d'une séquence vidéo sont transformées par le filtre temporel afin de tirer parti de leur redondance temporelle. Les sous-bandes temporelles résultantes sont alors décomposées spatialement pour exploiter la redondance spatiale présente dans ces images. Nous nous intéressons tout d'abord dans la section 5.2 à l'étude des caractéristiques spatiales et fréquentielles de ces sous-bandes temporelles. L'utilisation d'un schéma générique de construction de transformées M-bandes nous permet alors de spécifier une transformée 4-bandes adaptée à la décorrélation des sous-bandes temporelles de détail. Nous mettons ensuite en œuvre cette transformée au sein des codecs MC-EZBC et Vidwav et observons l'efficacité de codage qu'elle offre. D'autres transformées M-bandes sont aussi utilisées et comparées à la transformée 9/7 dyadique.

En dépit de leurs avantages, les transformées M-bandes semblent cependant ne pas posséder des propriétés de scalabilité suffisamment fines pour permettre d'offrir des changements de résolutions de rapports variés. En effet, dans un banc de filtres M-bandes classique, la sous-bande résultant du filtrage passe-bas peut servir d'approximation de l'image originale à une résolution réduite d'un facteur M, offrant ainsi seulement une scalabilité d'ordre M. Afin de pallier ce problème, nous présentons enfin dans la section 5.3 comment le banc de filtres de synthèse peut être modifié de façon à donner aux transformée M-bandes des propriétés de scalabilité étendues.

5.1 Bancs de filtres M-bandes ; rappels

Nous rappelons dans cette section la définition du banc de filtres M-bandes, étudié en détail par Vaidyanathan [155] et énonçons plusieurs de ses propriétés. Des exemples de transformées M-bandes communément utilisées en codage d'image sont alors présentés

et nous décrivons les relations qu'entretiennent les bancs de filtres M-bandes avec les bases d'ondelettes M-bandes.

5.1.1 Définition

La transformée en banc de filtres M-bandes est obtenue par extension du nombre de bandes mises en jeu dans un banc de filtres dyadique. De part son nombre accru de sous-bandes, elle est alors plus flexible que ce dernier et possède une meilleure sélectivité fréquentielle. Tout comme le banc d'analyse et de synthèse dyadique, il est possible d'accoler un banc d'analyse et de synthèse M-bandes de manière à créer un schéma de compression où le banc d'analyse sera utilisé lors de l'encodage. Un tel banc d'analyse-synthèse M-bandes est illustré par la Fig. 5.1.

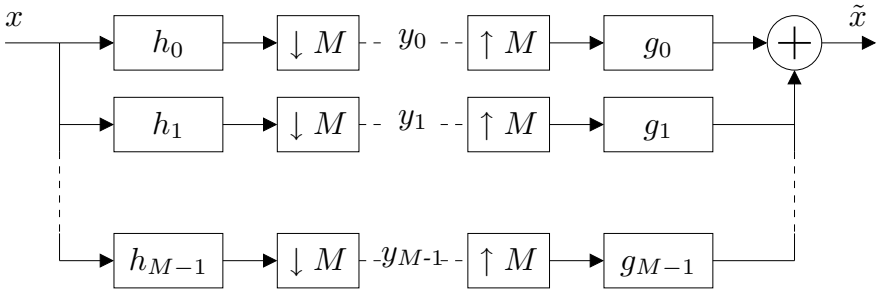


FIG. 5.1 – Banc de filtres d'analyse-synthèse M-bandes.

Dans le banc d'analyse, le signal d'entrée x est tout d'abord filtré par un jeu de M filtres d'analyse de réponses impulsionnelles $\{h_k\}_{0\leq k<M}$. Chaque sous-bande résultante est alors sous-échantillonnée d'un facteur M par l'opérateur $[\downarrow M]$. Cet opérateur est défini dans le cas général par :

$$u = [\downarrow M]v \quad \Leftrightarrow \quad \forall n \in \mathbb{Z},\ u[n] = v[nM] \tag{5.1}$$

On obtient alors les sous-bandes $\{y_k\}_{0\leq k<M}$, résultant de la décomposition du signal x par le banc d'analyse. Tout comme dans le cas du banc de filtres dyadique, ces sous-bandes sont destinées à être quantifiées et codées par un codeur entropique.

La reconstruction du signal est effectuée par le banc de synthèse. Les sous-bandes $\{y_k\}_{0\leq k<M}$ sont tout d'abord suréchantillonnées d'un facteur M par l'opérateur $[\uparrow M]$. Ce dernier est défini dans le cas général par :

$$u = [\uparrow M]v \quad \Leftrightarrow \quad \forall n \in \mathbb{Z},\ u[n] = \begin{cases} v[n/M] & \text{si } n \text{ est divisible par } M \\ 0 & \text{sinon} \end{cases} \tag{5.2}$$

Les sous-bandes suréchantillonnées sont alors filtrées par le jeu de filtres de synthèse de réponses impulsionnelles $\{g_k\}_{0\leq k<M}$. La sommation des signaux résultants du filtrage permet alors de reconstruire le signal $\widetilde{x}$.

Les conditions de reconstruction parfaites permettant d'assurer $\widetilde{x} = x$ ont été énoncées par Vaidyanathan [155]. Elles sont équivalentes à une condition de biorthogonalité sur les familles $\left\{h_k[n - Mt]\right\}_{t\in\mathbb{Z}}$ et $\left\{g_k[n - Mt]\right\}_{t\in\mathbb{Z}}$, et s'écrivent dans le domaine temporel sous la forme :

$$\forall\, k,\ \forall\, i, j \quad \sum_{n\in\mathbb{Z}} h_i(n) g_j(n - Mk) = \delta_k \delta_{i-j} \tag{5.3}$$

où δ est l'opérateur de Kronecker. Ces propriétés peuvent aussi s'énoncer dans le domaine fréquentiel par les relations :

$$\forall f,\ 0 \leq i,j < M, \qquad \sum_{k=0}^{M-1} \widehat{h}_i\Big(\frac{f+k}{M}\Big)\widehat{g}_j^{\,*}\Big(\frac{f+k}{M}\Big) = M\delta_{i-j} \tag{5.4}$$

Du fait du nombre accru de sous-bandes mises en jeu dans le banc de filtres, les équations de reconstruction parfaite laissent davantage de liberté pour le choix des filtres $\{g_k\}_{0\leq k<M}$ et $\{h_k\}_{0\leq k<M}$, comparé au cas dyadique. C'est la raison de la flexibilité accrue des transformées M-bandes.

On dira que le banc de filtres M-bandes est orthogonal si pour tout $0 \leq k < M$, $g_k[n] = h_k[-n]$. En effet, pour tout $n \in \mathbb{Z}$, les familles $\{h_k[n-jM]\}_{j\in\mathbb{Z}}$ et $\{g_k[n-jM]\}_{j\in\mathbb{Z}}$ forment alors une base orthogonale et ceci implique l'orthogonalité de la transformée M-bandes. C'est donc une isométrie qui préserve la norme ℓ_2 et donc l'énergie d'un signal, assurant ainsi $\sum_n x(n)^2 = \sum_j \sum_n y_j(n)^2$.

On remarquera de plus que la décomposition en paquets d'ondelettes dyadiques en utilisant une base de décomposition uniforme, où toutes les sous-bandes sont de même taille, est une transformée M-bandes où M est une puissance de deux.

Ondelettes M-bandes

Nous avons vu dans la section 1.2.2 les liens étroits qu'entretiennent les bancs de filtres dyadiques et l'analyse multirésolution par ondelettes. Il est en effet possible de montrer que toute transformée en ondelettes discrètes de support compact peut être mise sous forme de banc de filtres.

Tout comme les bases d'ondelettes dyadiques rappelées en section 1.2.1, il est possible de construire des bases d'ondelettes M-bandes [135] en utilisant *plusieurs* espaces de détail $\{\boldsymbol{W}_j^p\}_{1\leq p<M}$ au niveau j, pour représenter l'information de détail perdue entre deux niveaux de résolutions. Un espace de détail $\boldsymbol{W}_j^p$ est alors engendré par les translatées de la p-ième ondelette mère ψ^p dilatée au niveau j et est donc défini par :

$$\boldsymbol{W}_j^p = \Big\{ t \mapsto \frac{1}{M^{j/2}}\psi^p\big(\frac{t}{M^j} - k\big)\Big\}_{k\,\in\,\mathbb{Z}} \tag{5.5}$$

Lors de l'analyse multirésolution d'un signal, la différence entre l'approximation sur $\boldsymbol{V}_j$ et celle sur $\boldsymbol{V}_{j+1}$ peut alors être représentée par un ensemble de fonctions appartenant aux espaces $\{\boldsymbol{W}_j^p\}_{1\leq p<M}$. Il est alors possible de définir l'espace $\boldsymbol{V}_j$ par sommation directe d'espaces vectoriels selon la relation :

$$\boldsymbol{V}_j = \boldsymbol{V}_{j+1} \oplus \Big[\bigoplus_{p=1}^{M-1} \boldsymbol{W}_{j+1}^p\Big] \tag{5.6}$$

Cependant, nous ne nous attarderons pas dans la suite du document sur les ondelettes M-bandes et nos constructions de transformées M-bandes reposeront uniquement sur la formulation en banc de filtres présentée ci-dessus.

5.1.2 Transformées en blocs

Les transformées en blocs peuvent être vues comme des transformées M-bandes dont les filtres $\{h_k\}_{0\leq k<M}$ et $\{g_k\}_{0\leq k<M}$ ont une réponse impulsionnelle de taille $L = M$

échantillons. Nous rappelons les propriétés de quelques transformées en blocs très utilisées en compression d'image et en codage vidéo.

Transformée en cosinus discrète - DCT

La transformée en cosinus discrète ou DCT (*Discrete Cosine Transform*) a été popularisée par Rao [11] et est couramment utilisée en compression d'image et en codage vidéo : elle est en effet à la base des codecs JPEG et MPEG. C'est une transformée orthogonale qui correspond à des filtres à phase linéaire et qui peut s'interpréter comme la transformée de Fourier discrète d'un signal fini prolongé symétriquement sur ses bords. La DCT est aussi une transformée M-bandes dont les filtres $\{h_k\}_{0\leq k<M}$ de longueur $L = M$ échantillons sont donnés par :

$$\forall\, 0 \leq n < M, \quad h_k(n) = \lambda_k \sqrt{\frac{2}{M}} \cos\left[\frac{k\pi}{M}\left(n+\frac{1}{2}\right)\right] \quad \text{avec } \lambda_k = \begin{cases} 1/\sqrt{2} & \text{si } k{=}0 \\ 1 & \text{sinon} \end{cases}$$

La formulation en bancs de filtres n'est certainement pas la façon la plus efficace pour implémenter le calcul de la DCT et il existe de nombreux algorithmes rapides [36] visant à alléger la complexité de sa mise en œuvre. La forme 2D séparable de la DCT à 8-bandes est très utilisée en codage spatial pour décomposer les images car on peut montrer que son efficacité de décorrélation est proche de celle de la KLT. C'est pourquoi la DCT est utilisée dans les normes JPEG, MPEG-1, MPEG-2, MPEG-4 Partie 2 et dans la série des codecs H.261 et H.263 pour décomposer les images en blocs avant quantification et codage.

En codage d'images, un des inconvénients majeurs de la DCT réside dans sa propension à créer des artefacts de type blocs lorsque elle est utilisée à bas débit. En effet, la DCT est une transformée locale qui n'opère que sur le bloc courant de taille $M \times M$ pixels, indépendamment des blocs voisins. Lors de la perte d'informations par quantification, chaque bloc est alors décodé indépendamment des autres et des discontinuités peuvent apparaître à leurs frontières. Les transformées orthogonales à recouvrement ont été introduites pour pallier à cet inconvénient et sont abordées dans la suite du document.

Transformée DCT entière et transformée d'Hadamard

La DCT est une transformation dont les coefficients des filtres d'analyse et de synthèse sont des nombres réels. Cependant, une implémentation logicielle ou matérielle de la DCT utilise nécessairement une approximation à précision finie de ces coefficients. Lors du déploiement des normes JPEG et MPEG, on a observé que ces approximations étaient la cause de dérives importantes entre différentes implémentations de codecs d'images ou vidéo utilisant la DCT. Ces dérives sont ainsi dues à des architectures logicielles et matérielles différentes qui utilisent des opérateurs de précision inégale.

Afin d'éviter ces problèmes de dérives, la norme H.264 a opté pour l'utilisation de transformées spatiales *entières* afin d'obtenir des résultats déterministes, indépendants d'une architecture matérielle ou d'une factorisation particulière de la transformée. La norme H.264 [118] utilise ainsi deux transformées 4-bandes spatiales : une transformée DCT entière et une transformée de Hadamard.

La transformée DCT entière utilisée résulte simplement de l'approximation entière de la DCT classique sur 4 bandes. Elle est utilisée sous forme séparable 2D pour décorréler spatialement les blocs issues des résidus de la prédiction temporelle du codec

H.264. Cette transformée s'écrit classiquement sous la forme matricielle séparable $y = H_{\text{dcte}} x H_{\text{dcte}}^T$, où x est un bloc de 4×4 pixels et y les coefficients transformés. Vu comme une transformée 4-bandes, les coefficients de ses filtres d'analyse sont donnés par les lignes de la matrice H_{dcte} suivante :

$$\forall\, 0 \leq n < 4, \quad h_k(n) = H_{\text{dcte}}(n,k) \quad \text{avec} \quad H_{\text{dcte}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

La transformée de Hadamard est une autre transformée orthogonale utilisée par la norme H.264 pour transformer à nouveaux les coefficients de la composante basse fréquence issue de la transformée DCT entière. Elle s'écrit en utilisant la forme concise de cette dernière au moyen de la matrice H_{had} :

$$\forall\, 0 \leq n < 4, \quad h_k(n) = H_{\text{had}}(n,k) \quad \text{avec} \quad H_{\text{had}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$$

On notera l'existence de récentes alternatives à l'approximation de la DCT entière utilisée dans H.264, autorisant cependant une implémentation n'ayant pas recours à une arithmétique à virgule flottante. Ces travaux concernent la BinDCT de Tran [147], basée sur une factorisation directe en treillis de la DCT et d'autres travaux, dont la DCT entière de Chen [37]. Le sujet est cependant vaste et nous ne n'y attarderons pas.

5.1.3 Transformées à recouvrement

En dépit de leur efficacité de décorrélation et de leur simplicité d'utilisation, les transformées en blocs ont l'inconvénient de générer des artefacts aux bords des blocs très visibles lors de la quantification à bas débit. Afin d'amoindrir ces effets, plusieurs auteurs dont Malvar [82, 85] et Meyer [79] ont les premiers préconisé l'utilisation de transformées M-bandes *à recouvrement*, dont les supports des filtres ont une taille L supérieure au nombre de bandes M.

LOT, LBT et GenLOT

La transformée orthogonale à recouvrement, dénommée LOT (*Lapped Orthogonal Transform*), a été introduite et popularisée par Malvar [85]. C'est une modification de la DCT qui permet, contrairement à cette dernière, de décomposer un signal en blocs en utilisant le voisinage du bloc courant. Elle permet ainsi d'amoindrir nettement les problèmes d'artefacts de blocs très visibles lors de l'utilisation de la DCT avec une quantification à bas débit. La LOT est une transformée orthogonale, à phase linéaire dont la formulation classique est basée sur une modification du graphe de traitement de la DCT et est illustrée dans [85]. Elle peut cependant être représentée par un banc de filtres M-bandes dont les coefficients des filtres d'analyse $\{h_k\}_{0 \leq k < M}$ de support $L = 2M$ sont donnés dans [79].

La relaxation de la condition d'orthogonalité $g_k[n] = h_k[-n]$ a permis à Malvar [83] de construire une version biorthogonale de la LOT, la LBT (*Lapped Biorthogonal Transform*) ou transformée biorthogonale à recouvrement. La LBT est classiquement définie par une modification du graphe de traitement de la LOT, où les coefficients de la bande y_1 sont

multipliés par $\sqrt{2}$ à l'analyse et par $1/\sqrt{2}$ à la synthèse. Cette subtile modification permet à la LBT de projeter un signal sur une base dont les fonctions décroissent vers zéro en leurs bords. Cette propriété constitue une amélioration notable sur la LOT, dont les fonctions de base décroissent vers des valeurs *proches* de zéro. La LBT possède ainsi un meilleur gain de codage que la LOT ou la DCT et conduit expérimentalement à un gain significatif de l'efficacité de codage par rapport à ces dernières.

Des travaux ultérieurs dus à Nguyen, de Queiroz et Rao [132] ont permis la généralisation de la LOT et de la LBT en montrant que ces transformées sont des cas spéciaux de la GenLOT [45] (*GENeralized LOT*). Cette dernière étend la largeur de chevauchement des transformées à recouvrement en mettant en jeu des filtres d'analyse de taille KM.

MLT et MBLT

En parallèle des transformées LOT et LBT basées sur la DCT, Malvar a introduit une autre transformée à recouvrement : la transformée modulée à recouvrement [82], aussi nommée MLT (*Modulated Lapped Transform*). C'est une transformée orthogonale, basée sur la modulation d'un filtre prototype cosinusoïdal à réponse finie, dont les filtres d'analyse $\{h_k\}_{0\leq k<M}$ de support $L = 2M$ sont définis par :

$$\forall\, 0 \leq n < 2M, \quad h_k(n) = w(n)\sqrt{\frac{2}{M}}\cos\left[\left(n+\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}\right]$$
$$\text{avec} \quad w(n) = -\sin\left[\left(n+\frac{1}{2}\right)\frac{\pi}{2M}\right]$$

La MLT n'est pas une transformée à phase linéaire. Elle possède cependant une bonne sélectivité fréquentielle où les lobes secondaires sont peu importants, comme illustré sur la Fig. 5.2, où sont tracées les réponses fréquentielles des filtres $\{h_k\}_{0\leq k<4}$ intervenant dans la MLT à $M = 4$ bandes.

La relaxation de la contrainte portant sur la fenêtre w permet d'introduire une version biorthogonale de la MLT, la MLBT [83] (*Modulated Lapped Biorthogonal Transform*) ou transformée modulée biorthogonale à recouvrement. En partant du même filtre prototype de la MLT, mais modulé par des fenêtres d'analyse w_a et de synthèse w_s différentes, on définit la MLBT par :

$$\forall\, 0 \leq n < 2M, \quad h_k(n) = w_a(n)\sqrt{\frac{2}{M}}\cos\left[\left(n+\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}\right]$$
$$\forall\, 0 \leq n < 2M, \quad g_k(n) = w_s(n)\sqrt{\frac{2}{M}}\cos\left[\left(n+\frac{M+1}{2}\right)\left(k+\frac{1}{2}\right)\frac{\pi}{M}\right]$$

où les fenêtres symétriques d'analyse $w_a(n) = w_a(2M-n-1)$ et de synthèse $w_s(n) = w_s(2M-n-1)$ sont définies par :

$$\forall\, 0 \leq n < M, \quad w_s(n) = \frac{1-\cos\left[\left(\frac{n+1}{2M}\right)^{\alpha}\pi\right]+\beta}{2+\beta}$$
$$\forall\, 0 \leq n < M, \quad w_a(n) = \frac{w_s(n)}{w_s^2(n)^2 + w_s^2(n+M)}$$

Les paramètres α et β permettent de modifier les fenêtres w_a et w_s de la MLBT et d'agir sur ses propriétés. La MLBT possède une meilleure sélectivité fréquentielle que la MLT. Ceci s'observe sur la Fig. 5.2 où sont tracées les réponses fréquentielles des filtres 4-bandes d'analyse de la MLT et de la MLBT, paramétrées avec $\alpha = 0.85$ et $\beta = 0.0$

La MLT et la MLBT sont des transformées principalement utilisées en codage audio car leur non-linéarité de phase est gênante en codage d'image. Cependant, leur grande sélectivité fréquentielle en fait des transformées candidates intéressantes pour le codage des sous-bandes issues de la décomposition temporelle d'une séquence vidéo.

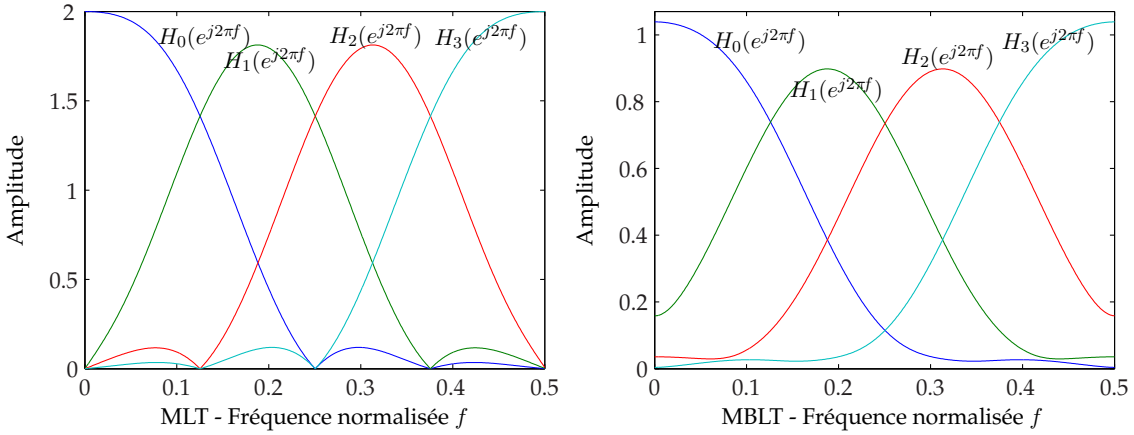


FIG. 5.2 – Réponses fréquentielles des filtres d'analyse $\{h_k\}_{0\leq k<4}$ de la MLT 4-bandes MLT (gauche) et de la MLBT 4-bandes (droite).

5.2 Codage spatial par bancs de filtres M-bandes

Dans le cadre de notre schéma de codage vidéo, nous nous intéressons dans un premier temps en section 5.2.1 à l'étude des différents types de sous-bandes issues du filtrage temporel. L'utilisation d'un cadre générique de construction de transformées M-bandes nous permet alors dans la section 5.2.2 de spécifier une transformée 4-bandes adaptée à la décorrélation des sous-bandes temporelles de détail. Enfin, des simulations expérimentales nous permettent de montrer le gain objectif en efficacité de codage apporté par cette transformée. Les résultats de ces travaux ont conduit à la publication d'un article de conférence [101], contribuant à la discussion sur un thème peu abordé dans la littérature.

Nous présentons alors en section 5.2.3 des résultats plus complets sur l'utilisation des transformées M-bandes comme alternative au filtrage 9/7 pour la décomposition des sous-bandes temporelles de détail. La transformée 4-bandes FB1 construite précédemment et les transformées 8-bandes LOT, LBT, MLT et MBLT sont alors mises en œuvre au sein du codec Vidwav afin de conduire les simulations. Les conclusions de cette étude publiée dans [100] sont cependant mitigées et nous tenterons d'en expliquer les raisons.

5.2.1 Caractéristiques des sous-bandes temporelles

Dans le cadre de notre schéma de codage vidéo, les sous-bandes issues de la décomposition temporelle d'une séquence vidéo sont des images présentant un aspect visuel très différent en fonction de leur type. À titre d'exemple, nous avons illustré en Fig. 5.3 les

sous-bandes issues de la décomposition temporelle d'un extrait de la séquence *Foreman*. Commentons tout d'abord l'aspect visuel de ces images.

Les sous-bandes d'approximation, provenant du filtrage temporel passe-bas, ont un aspect très similaire à une image naturelle. Comme vu dans la section 4.2.1 du chapitre précédent, elles peuvent toutefois comporter quelques artefacts fantômes et des discontinuités spatiales, provenant de la pseudo-inversion des champs de mouvement effectuée durant l'étape de mise à jour temporelle.

Au contraire, les sous-bandes de détail, provenant du filtrage temporel passe-haut, ne présentent pas du tout un comportement d'image naturelle : ces images sont de moyenne nulle, possèdent de nombreux "contours" et de larges zones de texture à hautes fréquences, baignant dans un bruit assez perceptible. Ces trames ressemblent beaucoup à des images de gradient ou à une image de résidu de prédiction temporelle de type DFD, intervenant dans les schémas de codage vidéo hybride décrits en section 2.1.

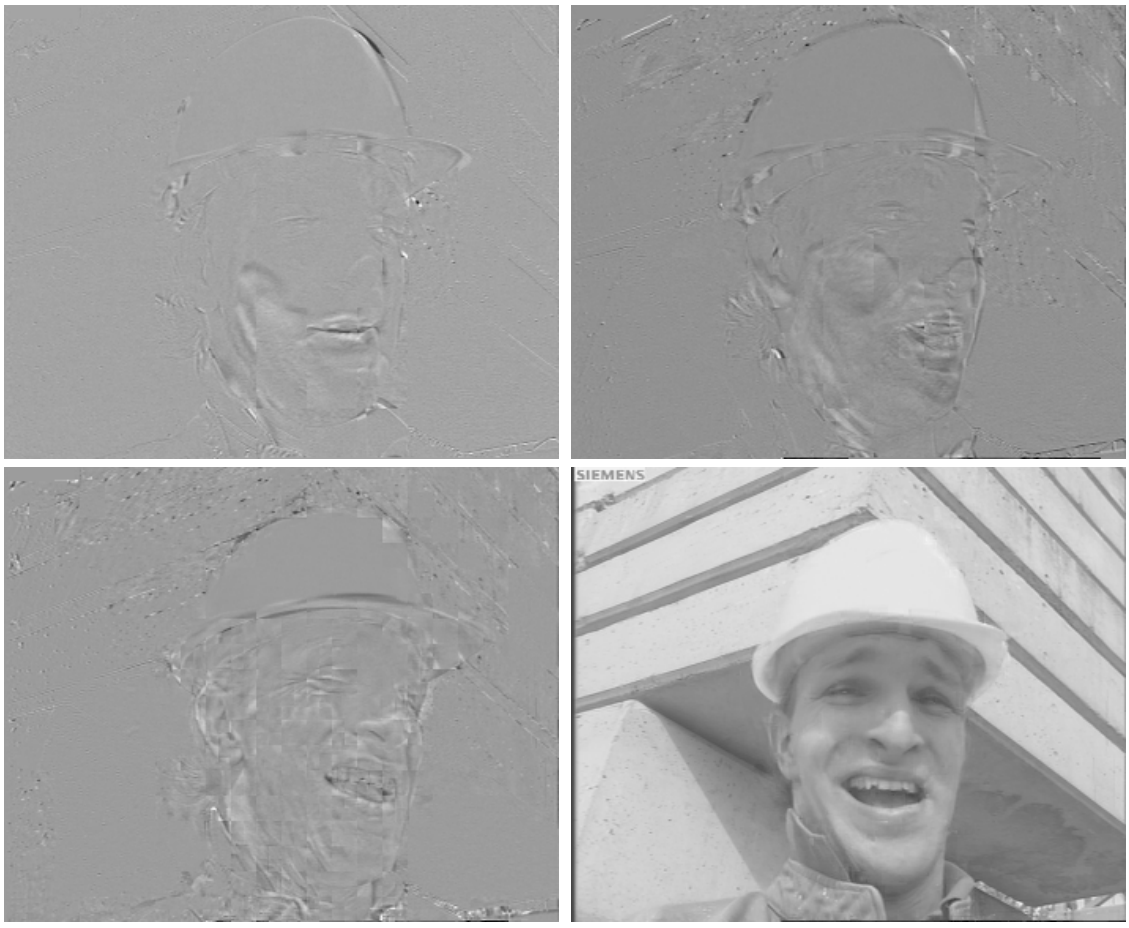


FIG. 5.3 – Sous-bandes issues de la décomposition temporelle 5/3 sur 4 niveaux de la séquence *Foreman* CIF. De gauche à droite et de haut en bas, on observe successivement une sous-bande de détail du premier, deuxième et troisième niveau, suivie d'une sous-bande d'approximation du quatrième niveau temporel.

L'étude des caractéristiques spatiales des sous-bandes temporelles peut être précisée dans le domaine fréquentiel. À cette fin, nous avons présenté dans les Tabs. 5.1 et 5.2 la répartition de la densité spectrale de puissance des sous-bandes d'approximation et de

détail, issues de la décomposition temporelle 5/3 des séquences *Mobile* et *Foreman*.

f_h/f_v	$[0, \frac{1}{8})$	$[\frac{1}{8}, \frac{1}{4})$	$[\frac{1}{4}, \frac{3}{8})$	$[\frac{3}{8}, \frac{1}{2}]$
$[0, \frac{1}{8})$	97.99%	0.40%	0.19%	0.05%
$[\frac{1}{8}, \frac{1}{4})$	0.66%	0.23%	0.12%	0.02%
$[\frac{1}{4}, \frac{3}{8})$	0.18%	0.07%	0.04%	0.01%
$[\frac{3}{8}, \frac{1}{2}]$	0.03%	0.01%	0.0%	0.0%

f_h/f_v	$[0, \frac{1}{8})$	$[\frac{1}{8}, \frac{1}{4})$	$[\frac{1}{4}, \frac{3}{8})$	$[\frac{3}{8}, \frac{1}{2}]$
$[0, \frac{1}{8})$	99.58%	0.07%	0.02%	0.01%
$[\frac{1}{8}, \frac{1}{4})$	0.18%	0.01%	0.0%	0.0%
$[\frac{1}{4}, \frac{3}{8})$	0.07%	0.01%	0.0%	0.0%
$[\frac{3}{8}, \frac{1}{2}]$	0.04%	0.01%	0.0%	0.0%

TAB. 5.1 – Répartition moyenne de la densité spectrale d'énergie observée sur les sous-bandes d'approximation issues du $4^{ème}$ niveau de la décomposition temporelle des séquences *Mobile* (gauche) et *Foreman* (droite), selon les bandes fréquentielles normalisées horizontale f_h et verticale f_v.

f_h/f_v	$[0, \frac{1}{8})$	$[\frac{1}{8}, \frac{1}{4})$	$[\frac{1}{4}, \frac{3}{8})$	$[\frac{3}{8}, \frac{1}{2}]$
$[0, \frac{1}{8})$	18.80%	10%	7.13%	3.27%
$[\frac{1}{8}, \frac{1}{4})$	14.91%	9.20%	6.65%	2.50%
$[\frac{1}{4}, \frac{3}{8})$	9.71%	6.09%	4.12%	1.21%
$[\frac{3}{8}, \frac{1}{2}]$	3.30%	1.75%	1.05%	0.29%

f_h/f_v	$[0, \frac{1}{8})$	$[\frac{1}{8}, \frac{1}{4})$	$[\frac{1}{4}, \frac{3}{8})$	$[\frac{3}{8}, \frac{1}{2}]$
$[0, \frac{1}{8})$	39.55%	10.29%	4.27%	1.96%
$[\frac{1}{8}, \frac{1}{4})$	13.89%	4.06%	1.74%	0.52%
$[\frac{1}{4}, \frac{3}{8})$	7.88%	2.73%	1.18%	0.3%
$[\frac{3}{8}, \frac{1}{2}]$	7.33%	2.88%	1.17%	0.25%

TAB. 5.2 – Répartition moyenne de la densité spectrale d'énergie observée sur les sous-bandes de détail issues du $4^{ème}$ niveau de la décomposition temporelle des séquences *Mobile* (gauche) et *Foreman* (droite), selon les bandes fréquentielles normalisées horizontale f_h et verticale f_v.

On aperçoit clairement la forte dissimilarité des caractéristiques spectrales des deux types de sous-bandes temporelles. Dans les images d'approximation, plus de 98 % de l'énergie de l'image est contenu dans la bande 2D de fréquence normalisée passe-bas inférieure à $1/8$. Il n'y a donc quasiment pas de moyennes et hautes fréquences dans les sous-bandes d'approximation. Le cas des sous-bandes temporelles de détail est très différent. La portion des basses fréquences inférieures à $1/8$ représente seulement 18.80 % de l'énergie totale sur *Mobile* et 40 % dans le cas de *Foreman*. Le spectre 2D est ainsi nettement plus riche en moyennes et hautes fréquences, horizontales comme verticales. Cependant, il est loin d'être uniformément réparti et les hautes fréquences diagonales sont par exemple rares. Nous ne sommes donc pas en présence d'un bruit blanc et pouvons ainsi espérer trouver une transformée adaptée à la décorrélation de ce type d'image.

Bien que ces deux types de sous-bandes temporelles ne partagent pas du tout les mêmes caractéristiques visuelles et fréquentielles, la majorité des schémas de codage vidéo $t + 2D$ [16, 26, 127, 164] utilisent la transformation en ondelettes biorthogonale 9/7 pour décomposer indifféremment les sous-bandes d'approximation et de détail. Ce choix se relève certainement justifié dans le cas des images d'approximation, tant elles ressemblent à des image naturelles et du fait que la transformation 9/7 soit adaptée [18, 159] à la décomposition de ce type d'images. Cependant, son utilisation pour le filtrage spatial des sous-bandes temporelles de détail n'est motivée par aucun argument.

Ces remarques motivent ainsi notre approche décrite dans la section suivante, consistant en la construction d'une transformée 4-bandes dont les caractéristiques de régularité et de sélectivité fréquentielle sont adaptées à la décomposition des sous-bandes de détail.

5.2.2 Construction d'un banc de filtres 4-bandes adapté

Nous souhaitons construire un banc de filtres M-bandes adapté à la décomposition spatiale des sous-bandes temporelles de détail. Cependant et avant d'imposer des conditions sur le banc de filtres, comment construit-on une transformée M-bandes ayant des propriétés précises ? La tâche n'est pas aisée car les équations de reconstruction parfaite (5.3) sont difficiles à satisfaire. Afin d'y parvenir simplement, nous nous proposons d'utiliser un algorithme de construction de transformées M-bandes offrant suffisamment de degrés de liberté pour nous permettre par la suite d'imposer des conditions supplémentaires de régularité.

Algorithme d'Alkin et Caglar

Alkin et Caglar [12] ont proposé un algorithme général de construction de bancs de filtres M-bandes orthogonaux, à phase linéaire et à reconstruction parfaite. Cet algorithme offre de plus suffisamment de degrés de liberté pour pouvoir ajouter des conditions supplémentaires sur le banc de filtres, de façon à l'adapter à la décorrélation d'un signal spécifique. Le cadre de construction impose que M soit une puissance de 2 et que les filtres $\{h_k\}$ soient tous d'une même taille L, multiple de $2M$; on notera ainsi $L = 2KM$.

Le principe est le suivant : étant donné un filtre prototype passe-bas h_0 symétrique par rapport à $-1/2$ et dont les M-translatées forment une famille orthogonale, comme spécifié par la relation (5.7), on obtient les coefficients des $M-1$ autres filtres par permutation des coefficients du filtre h_0.

$$\forall\, k \in \mathbb{Z} \quad \sum_n h_0(n) h_0(n - Mk) = \delta_k \tag{5.7}$$

Le filtre prototype passe-bas h_0 caractérise donc entièrement le banc de filtres. Les autres filtres $\{h_k\}_{1 \le k < M}$ sont obtenus à partir de h_0 et de la relation $h_k = \boldsymbol{B}_k h_0$ où $\boldsymbol{B}_k$ est une matrice de permutation calculée grâce aux relations suivantes et au moyen du produit tensoriel $\otimes$:

$$\boldsymbol{B}_k = \prod_{i=1}^{\log_2 M} \boldsymbol{A}_i^{r_{ki}}$$

$$\boldsymbol{A}_i = \boldsymbol{P}_{N/2^i} \otimes \boldsymbol{J}_{2^i}$$

$$\boldsymbol{P}_K = \{p_{jk}\}_{K\times K} \text{ avec } p_{jk} = (-1)^j \delta_{j-k} \quad \text{(matrice identité alternée)}$$

$$\boldsymbol{J}_K = \{a_{jk}\}_{K\times K} \text{ avec } a_{jk} = \delta_{K+1-j-k} \quad \text{(matrice contre-identité)}$$

où r_{ki} prend ses valeurs dans $\{0, 1\}$ et représente le i-ème bit de poids faible de la représentation binaire de l'entier k.

L'algorithme d'Alkin et Caglar permet ainsi de construire une transformée M-bandes à reconstruction parfaite à partir du seul filtre h_0. Celui-ci doit cependant être symétrique et vérifier les conditions d'orthogonalité (5.7). Ainsi, pour une transformée M-bandes dont les filtres ont une taille $L = 2KM$, on dispose de $d = L/2 - L/M = K(M-2)$ degrés de liberté pour choisir le filtre prototype passe-bas h_0.

Construction du filtre prototype h_0 dans le cas 4-bandes

Considérons dans un premier temps le cas $M = 4$ bandes car c'est le cas le plus simple non-dyadique autorisé par le schéma de construction d'Alkin et Caglar. De même, nous

choisissons une longueur de filtre $L = 8$ échantillons, correspondant au cas le plus simple $K = 1$. Cette longueur est proche de celle de l'ondelette 9/7 d'analyse et paraît raisonnable pour assurer une décorrélation efficace des sous-bandes temporelles de détail.

Dans le cas $M = 4$ bandes, les coefficients des filtres h_1, h_2 et h_3 sont alors obtenus à partir du filtre prototype passe-bas symétrique h_0 et des permutations suivantes :

$$\forall\ -4 \leq n \leq 3 \quad \left[\begin{array}{ll} h_1(n) & = (-1)^{\lfloor n/2 \rfloor} h_0(n + (-1)^n) \\ h_2(n) & = (-1)^n h_1(n) \\ h_3(n) & = (-1)^n h_0(n) \end{array} \right. \tag{5.8}$$

Conformément à la section précédente, nous disposons donc de $d = 2$ degrés de liberté pour choisir le filtre h_0. Nous décidons de les utiliser pour construire un banc de filtres régulier avec suffisamment de moments nuls, lui permettant ainsi d'assurer une bonne approximation polynomiale. À cette fin, nous imposons des conditions pour maximiser le nombre de moments nuls des ondelettes ψ_1, ψ_2 et ψ_3 sous-jacentes au banc de filtres et respectivement associées aux filtres h_1, h_2 et h_3.

Tout d'abord, le filtre prototype h_0 de taille $L = 8$ doit être symétrique et satisfaire les conditions d'orthogonalité exprimées par les relations (5.7). Ces hypothèses peuvent être traduites par les équations suivantes :

$$\left\{ \begin{array}{rl} h_0(0)^2 + h_0(1)^2 + h_0(2)^2 + h_0(3)^2 & = 1/2 \\ h_0(0)h_0(3) + h_0(1)h_0(2) & = 0 \end{array} \right. \tag{5.9}$$

Imposons désormais les moments nuls, grâce à une propriété liant une ondelette à son filtre miroir conjugué associé. Une ondelette ψ possède ainsi p moments nuls si et seulement si la transformée en Z de son filtre miroir $H(z)$ possède un zéro de multiplicité p en $z = 1$. L'antisymétrie des filtres h_1 et h_3 par rapport à $-1/2$ implique que $H_1(1) = H_3(1) = 0$ et confère donc à leur ondelettes respectives ψ_1 et ψ_3 au moins un moment nul.

Nous décidons d'utiliser un degré de liberté pour imposer un moment nul sur ψ_2. Cette condition revient donc à poser une contrainte d'égalité sur h_2 afin de vérifier $H_2(1) = 0$. Grâce aux relations de permutations (5.8), cette condition s'exprime alors sous forme de contrainte sur h_0 par :

$$h_0(0) + h_0(3) = h_0(1) + h_0(2) \tag{5.10}$$

Afin de minimiser les coefficients d'ondelettes situés dans la dernière bande et de permettre ainsi au codeur à arbre de zéros sous-jacent de mieux traiter les coefficients des bandes précédentes, nous imposons à ψ_3 de posséder au moins deux moments nuls. Ceci équivaut à exiger que la dérivée de la transformée en Z de h_3 vérifie $H_3'(1) = 0$, conduisant à $\sum_k k h_3(k) = 0$. De plus, due à l'antisymétrie de h_3, cette condition se simplifie par $\sum_{k \geq 0} (2k + 1) h_3(k) = 0$, imposant à son tour une nouvelle contrainte sur h_0 :

$$-7h_0(3) + 5h_0(2) - 3h_0(1) + h_0(0) = 0 \tag{5.11}$$

Il est utile de rappeler que dû à la symétrie de h_2 par rapport à $-1/2$, on peut montrer que $H_2(1) = 0$ implique $H_2'(1) = 0$ et qu'ainsi ψ_2 possède deux moments nuls. De même et à cause de l'antisymétrie de h_3, $H_3'(1) = 0$ implique que $H_3''(1) = 0$, donnant alors trois moments nuls à ψ_3. Au final, les ondelettes ψ_1, ψ_2 et ψ_3 possèdent ainsi respectivement 1, 2 et 3 moments nuls.

La réunion des conditions (5.9), (5.10) et (5.11) amène alors au système d'équations suivant, qui lie les coefficients de h_0 :

$$\begin{cases} h_0(0)^2 + h_0(1)^2 + h_0(2)^2 + h_0(3)^2 &= 1/2 \\ h_0(0)h_0(3) + h_0(1)h_0(2) &= 0 \\ h_0(0) + h_0(3) - h_0(1) - h_0(2) &= 0 \\ -7h_0(3) + 5h_0(2) - 3h_0(1) + h_0(0) &= 0 \end{cases}$$

Ce système non-linéaire est résoluble et possède une unique solution :

$$\begin{cases} h_0(0) = (5+\sqrt{15})/16 \\ h_0(1) = (3+\sqrt{15})/16 \\ h_0(2) = (5-\sqrt{15})/16 \\ h_0(3) = (3-\sqrt{15})/16. \end{cases} \tag{5.12}$$

Le filtre prototype symétrique h_0 dont les coefficients sont donnés par la relation (5.12) permet ainsi de construire un banc de filtres 4-bandes au moyen des équations de permutation (5.8). Par construction, ce banc à reconstruction parfaite est orthogonal, à phase linéaire et ses ondelettes ψ_1, ψ_2 et ψ_3 sous-jacentes possèdent respectivement 1, 2 et 3 moments nuls. Il sera nommé FB1 dans la suite du document. On remarquera que dans le cas dyadique, il n'est pas possible de construire un banc de filtres simultanément orthogonal et à phase linéaire (mis à part le cas trivial de Haar) : ceci justifie ainsi la flexibilité accrue des transformées M-bandes.

Il est bien sûr possible de fixer d'autres conditions sur le banc de filtres. À titre expérimental, nous pouvons ainsi imposer à ψ_1, ψ_2 et ψ_3 de posséder respectivement 3, 2 et 1 moments nuls. En suivant le même raisonnement que précédemment, on aboutit à un banc de filtres 4-bandes que nous nommerons FB2 et dont les coefficients du filtre prototype passe-bas h_0 sont donnés par la relation (5.13) suivante :

$$\begin{cases} h_0(0) = (3+\sqrt{3})/8 \\ h_0(1) = (3-\sqrt{3})/8 \\ h_0(2) = (1+\sqrt{3})/8 \\ h_0(3) = (1-\sqrt{3})/8. \end{cases} \tag{5.13}$$

On remarquera que les exemples de bancs de filtres donnés par Alkin [12] pour illustrer son algorithme n'ont pas été obtenus en imposant ces conditions formelles. Pour calculer les coefficients du filtre prototype h_0, l'auteur utilise au contraire une stratégie d'optimisation numérique ayant pour but de maximiser le gain de codage du banc de filtres. Il obtient alors des résultats numériques dont les valeurs approchées ne satisfont l'équation (5.10) qu'à 10^{-3} près. Il en résulte une transformée qui, en présence d'un signal constant, donne des coefficients non strictement égaux à 0, mais plus proches de 10^{-3}. Cela est probablement dû au fait que le calcul des coefficients par la stratégie d'optimisation numérique ait été mené avec une précision machine de 10^{-7}. Dans le contexte d'un codeur emboîté à arbre de zéros, ces coefficients non strictement nuls sont susceptibles de dégrader l'efficacité du codage.

Nous présentons dans le Tab. 5.3 les valeurs numériques des filtres prototypes h_0 obtenues pour les bancs de filtres FB1 et FB2 et les comparons avec celles calculées par Alkin pour le banc de filtres 4-bandes à 8 échantillons. On observe la forte similitude des coefficients de FB1 avec ceux d'Alkin. De plus, afin d'étudier quelques propriétés des bancs FB1 et FB2, nous avons représenté sur la Fig. 5.4 les fonctions de transfert de leurs filtres.

Les fonctions de transfert du banc d'Alkin, très proches de celles de FB1, n'ont pas été représentées.

n	$h_0(n)$ FB1	$h_0(n)$ FB2	$h_0(n)$ Alkin
0	0.554561459	0.591506350	0.567030813
1	0.429561459	0.158493649	0.406151488
2	0.070438540	0.341506350	0.094517754
3	-0.054561459	-0.091506350	-0.067700953

TAB. 5.3 – Comparaison des valeurs numériques des coefficients de h_0.

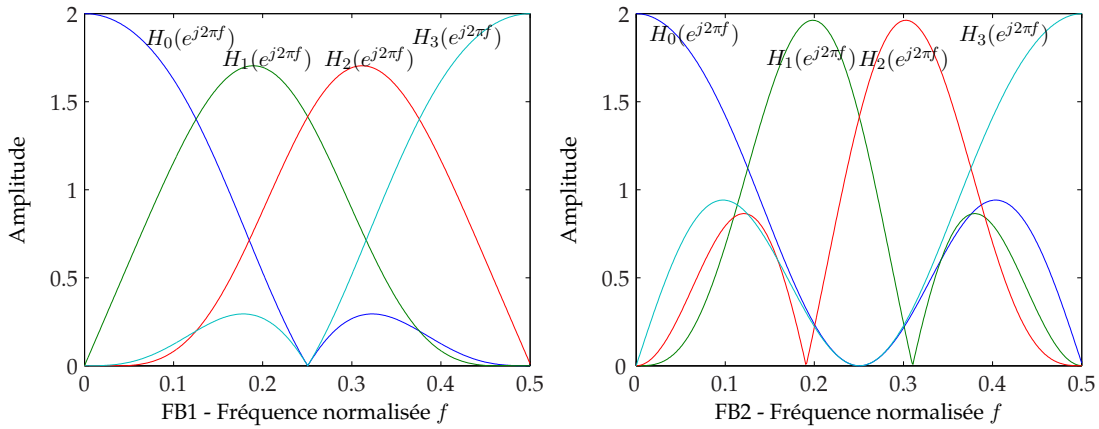


FIG. 5.4 – Réponses fréquentielles des filtres d'analyse $\{h_k\}_{0\leq k<4}$ des bancs de filtres 4-bandes FB1 (gauche) et FB2 (droite).

On observe la bonne sélectivité fréquentielle offerte par le banc de filtres FB1. Au contraire, les réponses fréquentielles des filtres du banc FB2 semblent posséder des lobes secondaires importants, pouvant nuire à l'efficacité de la décorrélation. Enfin, pour illustrer le lien existant entre le banc de filtres 4-bandes FB1 et ses ondelettes sous-jacentes, nous avons tracé sur la Fig. 5.5 la fonction d'échelle ϕ et les ondelettes ψ_1, ψ_2 et ψ_3 associées respectivement aux filtres h_0, h_1 h_2 et h_3 du banc FB1.

Résultats expérimentaux

Afin d'évaluer le gain objectif en efficacité de codage apporté par les transformées FB1 et FB2 utilisées pour la décomposition spatiale des sous-bandes temporelles de détail, nous avons réalisé des simulations de codage vidéo en utilisant notre prototype basé sur le codec MC-EZBC. Lors de nos expérimentations, nous avons considéré les séquences CIF *Mobile* et *City* à 30 Hz, choisies pour la grande diversité de mouvement et de textures qu'elles offrent. Ces dernières ont alors été décomposées sur 5 niveaux temporels par la transformée 5/3 uniforme, décrite dans la section 4.2. Les sous-bandes temporelles d'approximation résultantes ont été décomposées spatialement sur 5 niveaux avec la transformée 9/7. Enfin, les sous-bandes temporelles de détail et d'approximation ont été encodées avec l'algorithme EZBC qui a du être adapté pour fonctionner avec des pyramides spatiales 4-bandes. La modification a simplement consisté en la réorganisation de la pyramide afin qu'elle ressemble à une pyramide spatiale dyadique classique.

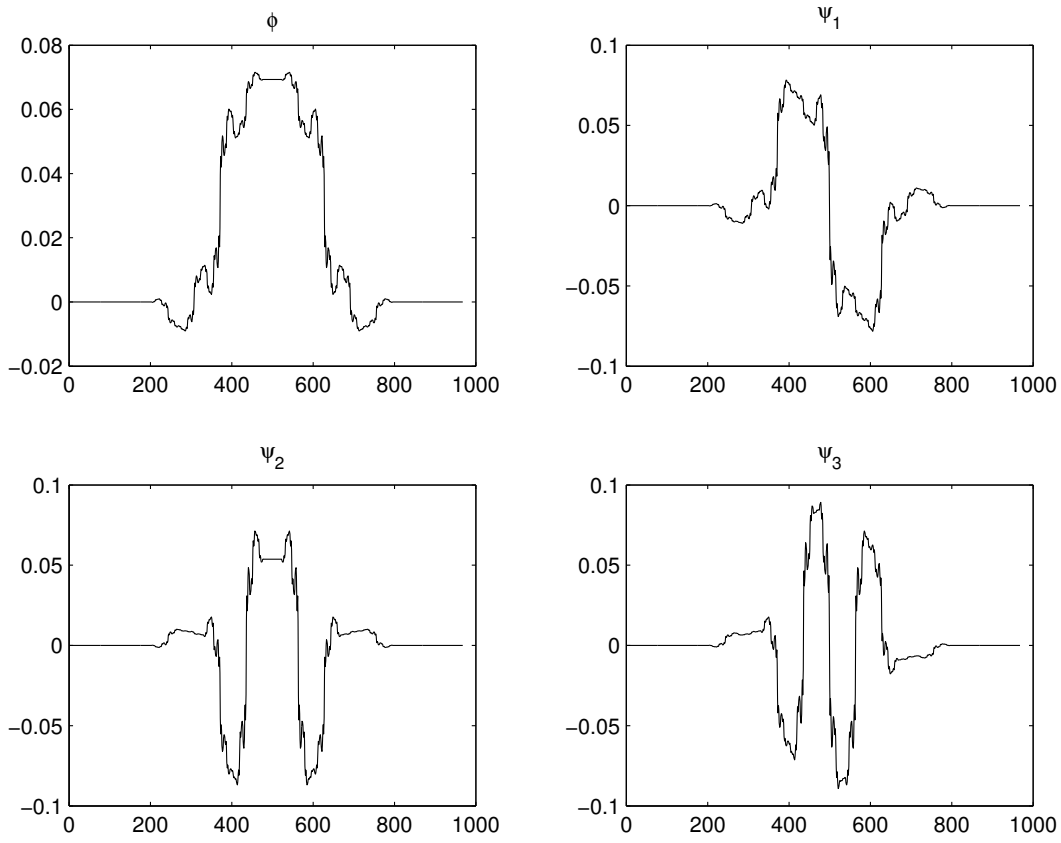


FIG. 5.5 – Fonction d'échelle et ondelettes associées au banc de filtres 4-bandes FB1.

Nous présentons dans les Tabs. 5.4 et 5.5 l'efficacité de codage obtenue en utilisant les transformées 4-bandes FB1 et FB2 pour décomposer les sous-bandes temporelles de détail sur un niveau spatial. Afin de comparer ces résultats, nous avons de plus effectué des simulations de codage en utilisant les transformées dyadiques 9/7 et 5/3 pour décomposer ces mêmes sous-bandes temporelles sur deux niveaux spatiaux. L'ensemble des résultats est exprimé sous forme de Y-PSNR moyen calculé sur l'ensemble des images décodées.

YSNR (en dB)	384 kbs	512 kbs	768 kbs	1024 kbs
4-band FB1	27.93	29.95	32.16	33.65
4-band FB2	27.80	29.70	31.70	33.05
9/7	27.71	29.76	32.00	33.47
5/3	27.40	29.52	31.60	33.05

TAB. 5.4 – Comparaison débit-distorsion de plusieurs transformées spatiales sur la séquence *Mobile* CIF à 30 Hz.

Nous observons que l'utilisation du banc de filtres FB1 pour décomposer les sous-bandes temporelles de détail apporte un gain d'environ 0.2 dB à tous les débits, comparé à la transformée 9/7 et un gain d'environ 0.5 dB lors de la comparaison avec la transformée 5/3. Nous remarquons aussi les performances médiocres obtenues avec le banc

YSNR (en dB)	384 kbs	512 kbs	768 kbs	1024 kbs
4-band FB1	33.61	35.45	37.65	39.26
4-band FB2	33.44	35.10	37.08	38.59
9/7	33.40	35.23	37.47	39.09
5/3	33.16	34.92	37.06	38.73

TAB. 5.5 – Comparaison débit-distorsion de plusieurs transformées spatiales sur la séquence *City* CIF à 30 Hz.

de filtres FB2, qui montre une efficacité inférieure à celle de la transformée 9/7. Cette contre-performance n'est pas étonnante et est certainement due à la mauvaise sélectivité fréquentielle du banc de filtres FB2.

Les gains en efficacité de codage observés lors de l'utilisation du banc de filtres FB1 pour filtrer spatialement les sous-bandes de détail, bien que modérés, sont cependant encourageants. Ils justifient en effet la démarche suivie, à savoir la construction d'une transformée M-bandes en imposant des conditions de régularité sur ses filtres sous-jacents. Il serait alors intéressant de continuer ces travaux en utilisant des bancs de filtres plus longs et sur un nombre supérieur de bandes. Cependant, l'architecture du codec spatial EZBC ne s'y prête pas car ce dernier ne gère que les pyramides spatiales dyadiques. Nous nous proposons dans la section suivante d'utiliser le codec Vidwav pour continuer notre étude sur les transformées M-bandes utilisées pour la décomposition spatiale des sous-bandes temporelles de détail.

5.2.3 Étude de différents bancs de filtres

Le codec vidéo MPEG Vidwav, issu des travaux de Song, Wu, Xiong et Xu [133, 164, 165] sur l'algorithme 3D-ESCOT, est un schéma de codage vidéo générique et efficace décrit dans la section 2.2.5. Il offre des performances équivalentes au schéma de codage MC-EZBC en résolution nominale mais donne de meilleurs résultats en scalabilité spatiale. De plus, il est nettement plus flexible que le codec MC-EZBC et gère correctement les décompositions spatiales en paquets d'ondelettes. Dans le contexte du processus de normalisation MPEG sur le codage vidéo scalable SVC, le codec Vidwav a été mis tardivement à la disposition des membres MPEG en Mai 2005. De part son efficacité de codage et sa souplesse d'utilisation, il nous a alors semblé intéressant d'utiliser ce codec afin de poursuivre nos investigations sur les décompositions M-bandes des sous-bandes temporelles de détail.

On remarquera les travaux similaires sur les paquets d'ondelettes proposés par Cheng [162] pour décomposer les résidus de prédiction temporelle DFD dans le cadre d'un codeur vidéo hybride MPEG-2. Bien qu'intéressants, les résultats ne sont cependant pas entièrement satisfaisants et nécessitent l'utilisation d'algorithmes complexes de recherche de la meilleure base. Dans la continuation de ces travaux, Trocan [149] a présenté dans le contexte du codec Vidwav, des résultats nettement plus encourageants sur l'utilisation des décompositions en paquets d'ondelettes des sous-bandes temporelles de détail.

Nous avons ainsi conduit plusieurs simulations de codage vidéo en utilisant le codec Vidwav. Les séquences ont été encodées dans le mode $t+2D$, sans utilisation du module *Inband* $2D+t+2D$ ni du module *Base layer*. L'estimation de mouvement est basée sur des blocs de tailles variant de 4×4 à 16×16 pixels en utilisant des champs estimés au $1/4$ de pixel près. La décomposition temporelle des séquences a été réalisée au moyen

du filtre 5/3 sur 4 niveaux où chaque bloc est filtré selon son mode de prédiction le plus favorable, tel que décrit dans la section 2.2.5. Les sous-bandes temporelles sont codées au moyen de l'algorithme 3D-ESCOT.

L'algorithme 3D-ESCOT est capable de gérer les décompositions spatiales dyadiques en paquets d'ondelettes. Or, il est clair que la structure de décomposition d'une transformée M-bandes peut être exprimée sous forme d'une base dyadique en paquets d'ondelettes si M est une puissance de deux. Il suffit ainsi de décomposer uniformément toutes les sous-bandes d'approximation et de détail de manière à obtenir une base de décomposition où toutes les sous-bandes sont de même taille. Le codec Vidwav peut donc coder les coefficients issus d'une transformée M-bandes si M est une puissance de 2.

Dans nos expérimentations, les sous-bandes temporelles d'approximation sont décomposées spatialement sur 5 niveaux par la transformée dyadique 9/7. Afin d'évaluer l'efficacité des transformées M-bandes, les sous-bandes de détail sont décomposées avec la transformée dyadique 9/7 sur deux niveaux spatiaux et sur un seul niveau avec les transformée 4-bandes MLT, MBLT et FB1 et les transformées 8-bandes LOT et LBT. L'expérimentation supplémentaire MLT2 a été conduite avec la transformée MLT sur deux niveaux spatiaux. L'implémentation de la LBT et de la LOT est due à Malvar [81, 84].

Nous comparons dans les Tabs. 5.6, 5.7 et 5.8 l'efficacité de codage observée à plusieurs débits avec les différentes transformées spatiales proposées pour la décomposition des sous-bandes temporelles de détail. L'efficacité de codage est exprimée en terme de PNSR moyen calculé sur la composante de luminance Y de l'ensemble des images décodées. De plus, nous avons inséré à des fins de comparaison les résultats de codage obtenus avec le codec MPEG/ITU SVC, en cours de normalisation dans sa version JSVM-1.

YSNR (en dB)	192 kbs	256 kbs	384 kbs
2-band 9/7	26.04	27.47	29.20
4-band FB1	26.06	27.46	29.17
4-band MLT	**26.12**	**27.51**	**29.20**
4-band MLT2	25.77	27.15	28.75
4-band MLBT	25.92	27.17	28.66
8-band LOT	26.02	27.36	29.03
8-band LBT	25.83	27.18	28.82
JSVM-1	26.11	27.25	29.16

TAB. 5.6 – Comparaison débit-distorsion de différentes transformées spatiales utilisées pour la décomposition des sous-bandes de détail de la séquence *Mobile* CIF à 30 Hz.

Nous observons tout d'abord que la MLT et la transformée dyadique 9/7 donnent les meilleurs résultats globaux dans ces expérimentations et pour tous les débits. La transformée MLT est la plus efficace sur les séquences *Mobile* et *Bus* tandis que la performance du filtre 9/7 s'illustre sur la séquence *Foreman*. Ceci peut être expliqué par les mouvements complexes rotatoires observés sur cette séquence ; ceux-ci gênent la prédiction temporelle assurée par le filtre 5/3 et induisent la création de zones mal-prédites au sein des sous-bandes temporelles de détail. Ces zones possèdent des caractéristiques proches d'une image naturelle, comme illustré par la Fig. 5.3 et justifié dans le Tab. 5.2 par la présence de basses fréquences. Dû à son nombre de moments nuls élevés, la transformée 9/7 possède alors de meilleures propriétés d'approximation sur ces zones, expliquant ainsi le

YSNR (en dB)	128 kbs	160 kbs	192 kbs	256 kbs
2-band 9/7	**31.47**	**32.46**	**33.22**	**34.33**
4-band FB1	31.23	32.22	32.97	34.07
4-band MLT	31.24	32.23	32.99	34.12
4-band MLT2	31.06	31.99	32.73	33.79
4-band MBLT	30.97	31.89	32.68	33.85
8-band LOT	31.43	32.38	33.13	34.22
8-band LBT	31.31	32.03	33.03	34.14
JSVM-1	30.92	31.96	32.70	34.20

TAB. 5.7 – Comparaison débit-distorsion de différentes transformées spatiales utilisées pour la décomposition des sous-bandes de détail de la séquence *Foreman* CIF à 30 Hz.

YSNR (en dB)	256 kbs	320 kbs	384 kbs	512 kbs
2-band 9/7	27.41	28.45	29.21	30.50
4-band FB1	27.41	28.44	29.22	30.53
4-band MLT	**27.43**	**28.47**	**29.26**	**30.57**
4-band MLT2	27.20	28.15	28.81	30.10
4-band MLBT	27.14	28.12	28.76	30.02
8-band LOT	27.40	28.42	29.18	30.45
8-band LBT	27.24	28.26	29.06	30.28
JSVM-1	27.11	27.84	28.90	30.09

TAB. 5.8 – Comparaison débit-distorsion de différentes transformées spatiales utilisées pour la décomposition des sous-bandes de détail de la séquence *Bus* CIF à 30 Hz.

gain observé par rapport à la transformée MLT.

On notera aussi les bonnes performances globales obtenues par la transformée à recouvrement LOT. Sachant que les algorithmes rapides implémentant cette transformée sont basés sur la DCT et que ces derniers se retrouvent aisément dans des équipement matériels destinés au traitement vidéo, la LOT constitue une bonne alternative entre la performance qu'elle offre et la complexité qu'elle nécessite. Dans le cadre de son codec PTC [84], Malvar a de plus montré que le calcul de la LOT possède une complexité inférieure à celle de la transformée 9/7 d'environ 15 %.

Durant ces expérimentations, la transformée FB1 donne cependant des résultats inférieurs à ceux observés avec la MLT ou la transformée 9/7. Ces observations ne confirment donc pas les conclusions tirées dans la section précédente où, dans le cadre du codec vidéo MC-EZBC, la transformée FB1 surpassait la transformée 9/7 d'environ 0.3 dB. Ceci peut s'expliquer par la prédiction temporelle avec sélection de blocs opérée par le codec Vidwav. Ce type de prédiction diminue en effet le coût des blocs mais au prix d'une augmentation du nombre d'artefacts dans les sous-bandes temporelles de détail, qui ne sont pas traités efficacement par la transformée FB1.

Les transformées biorthogonales MLBT et LBT ne se sont pas distinguées par de bons résultats dans ces simulations. En dépit de sa meilleure sélectivité fréquentielle, la MLBT possède en effet un gain de codage inférieur à la MLT. Il semble en fait que la biortho-

gonalité de ces transformées nuit à leur efficacité de décorrélation, laissant penser que l'algorithme d'allocation débit-distorsion intégré dans le codec 3D-ESCOT nécessite la présence d'une transformée orthogonale.

Enfin, on observe les mauvais résultats obtenus avec l'expérimentation MLT2, basée sur la décomposition des sous-bandes temporelles de détail par la transformée MLT sur deux niveaux spatiaux. Comparé à la transformée MLT sur un seul niveau spatial, on observe ainsi que l'ajout de ce niveau fait chuter l'efficacité de codage. La raison pourrait consister en la présence des 31 sous-bandes spatiales mises en jeu dans l'expérimentation MLT2, ajoutant une surcharge importante au bitstream qui n'est pas négligeable dans la gamme de bas débits sur lesquels les simulations sont faites.

5.3 Scalabilité fractionnaire

Malgré leur flexibilité et leur grande sélectivité fréquentielle, les transformées M-bandes semblent cependant ne pas posséder de propriétés de scalabilité aussi fines que les transformées dyadiques. En effet, dans le banc de filtres M-bandes classique, seule la sous-bande résultant du filtrage passe-bas peut servir d'approximation de l'image originale, conduisant ainsi à un facteur de scalabilité d'ordre M.

Avant d'aborder ce problème, nous détaillons tout d'abord dans la section 5.3.1 la motivation et les raisons sous-jacentes à l'utilité de la scalabilité fine, notamment dans le domaine spatial. Nous aborderons de plus les stratégies actuellement utilisées pour pallier à l'absence d'une telle scalabilité. Nous rappellerons enfin dans cette section les liens étroits qu'entretient la notion de scalabilité avec les méthodes classiques utilisées pour changer la résolution d'une image.

Nous présentons et démontrons alors dans la section 5.3.2 une propriété étonnante, capable de donner à un banc de filtres M-bandes la faculté de pouvoir reconstruire l'image de départ réduite d'un facteur rationnel quelconque M/P. Cette propriété, nommée scalabilité fractionnaire ou scalabilité rationnelle, est obtenue par une simple modification du banc de synthèse, sans changement du banc d'analyse. Elle permet ainsi la construction de schémas de décodage par sous-bandes offrant des facteurs de scalabilité rationnels. Elle étend ainsi grandement la gamme de facteurs de scalabilité que l'on peut obtenir en codage par sous-bandes, comparée aux seuls facteurs dyadiques offerts par les transformées en ondelettes classiques.

Par rapport à une stratégie simple qui consisterait à reconstruire entièrement une image puis à la redimensionner à une résolution réduite M/P, notre technique permet de reconstruire *directement* l'image à la résolution réduite. Nous calculons dans la section 5.3.3 la complexité théorique des deux stratégies et montrerons alors la supériorité de notre approche.

Enfin, nous montrons en section 5.3.4 plusieurs résultats expérimentaux pour confirmer notre méthode. Nous présentons tout d'abord quelques images obtenues en modifiant un banc de synthèse M-bandes et évaluons la vitesse de reconstruction. Nous expérimentons alors divers filtres de rééchantillonnage pour évaluer leur influence sur la qualité de l'image reconstruite et donnons enfin de nombreux résultats d'efficacité de codage en utilisant la propriété de scalabilité rationnelle.

Ces travaux ont fait l'objet de la publication d'un article de revue [103] qui dans un premier temps propose et démontre la technique proposée, puis décrit la complexité théorique de notre approche. Les résultats expérimentaux ont quant à eux fait l'objet de la

soumission d'un article de conférence [102].

5.3.1 Motivation

Le codage d'image par transformée en ondelettes n'offre qu'une scalabilité de type dyadique où la sous-bande d'approximation peut être vue comme la version réduite d'un facteur 2 de l'image originale. Les décompositions successives de la sous-bande d'approximation permettent alors d'obtenir d'autres facteurs mais tous sont des puissances de deux et aucun autre facteur ne peut être obtenu directement. Enfin, les transformées M-bandes étudiées dans ce chapitre offrent une scalabilité encore plus grossière où tous les facteurs sont des puissances de M.

Divers articles [50, 86] soulignent pourtant la nécessite de pouvoir gérer une large gamme de facteurs de scalabilité et, si possible, de pouvoir tenir compte des facteurs rationnels. Prenons l'exemple des nouveaux formats HDTV 720p et 1080p, dont les résolutions sont liées par un facteur $2/3$. En utilisant une transformée en ondelettes classique, il n'est pas possible de construire un flux scalable capable d'être reconstruit à ces deux résolutions. Il n'est pas non plus possible de créer un flux capable de représenter deux formats n'ayant pas le même facteur de forme (*aspect ratio*) : c'est le cas des formats $16/9$ et $4/3$. La même situation se présente si on désire diffuser un contenu visuel scalable destiné à des téléphones mobiles qui ne possèdent pas les mêmes tailles d'écran.

Comment obtenir alors un schéma de codage capable d'offrir des rapports rationnels et non-dyadiques ? Bien qu'il n'existe pas de transformées capables d'offrir naturellement de tels rapports, il est possible d'utiliser des stratégies de type Simulcast ou des schémas de codage prédictif en couches pour contourner cette limitation. C'est en suivant cette approche que Marquant [86] a ainsi proposé d'utiliser des facteurs de scalabilité rationnels lors de la construction des différentes couches spatiales utilisées dans le codec SVC.

Une autre classe de stratégies consiste en la reconstruction totale de l'image, suivie d'une étape de redimensionnement afin que la taille de l'image décodée soit adaptée à la résolution du terminal récepteur. C'est cependant une approche coûteuse en terme de complexité car l'image complète doit être décodée à pleine résolution puis rééchantillonnée. On remarquera que d'autres techniques existent comme dans [96] où les auteurs proposent une stratégie de redimensionnement d'image dans le domaine transformé DCT en utilisant un module externe de transcodage. Cette approche ne nécessite pas la modification du parc de terminaux récepteurs mais est très coûteuse en termes de calculs car elle nécessite le décodage et le réencodage des images.

Notre approche

Nous proposons dans la section 5.3.2 une solution générique permettant de pallier à ces problèmes. Dans le contexte d'un banc de filtres M-bandes, cette solution consiste à modifier le banc de synthèse, de façon à ce que dernier puisse reconstruire une image d'une résolution réduite d'un facteur rationnel quelconque M/P. Nous montrons de plus que l'image reconstruite n'est pas quelconque ; elle est identique à celle qu'on aurait obtenue par redimensionnement de l'image originale au moyen d'un opérateur donné. Nous concluons enfin en montrant que cette modification permet de donner au banc de filtres M-bandes la faculté de pouvoir gérer les facteurs de scalabilité rationnels.

Les travaux de Gopinah [55] et Kovačević [70] traitent de bancs de filtres à reconstruction parfaite et à facteurs d'échantillonnage rationnels. Les auteurs abordent ici le

problème de la construction de bancs de filtres capable d'analyser un signal sur une division non-uniforme du spectre, où chaque branche du banc de filtres d'analyse possède un facteur d'échantillonnage rationnel. Sur la base de ces travaux, Vaidyanathan [35] a alors proposé un cadre unificateur, amenant à la construction de banc de filtres multidimensionnels rationnels. D'autres travaux [19, 22] ont fait le lien avec l'analyse multidimensionnelle rationnelle. Cependant, tous ces travaux abordent la construction de bancs de filtres où le banc d'analyse *et* le banc de synthèse sont simultanément spécifiés de façon à fournir *un seul* facteur de scalabilité rationnel. Notre approche est différente dans la mesure où nous proposons à partir d'un banc d'analyse M-bandes quelconque, de modifier son banc de synthèse associé de façon à fournir un facteur de scalabilité rationnel *quelconque* M/P, où M et P sont des entiers avec $P \leq M$.

On remarquera enfin les articles [90, 91] décrivant une méthode de codage d'image permettant l'obtention de facteurs rationnels de la forme $N/2^D$. Elle repose sur l'utilisation d'une décomposition en paquets d'ondelettes sur une base uniforme et consiste en une reconstruction suivie d'une étape de redimensionnement. Notre approche permet cependant l'obtention d'une gamme de facteurs plus étendue et est moins complexe car elle ne nécessite pas d'étape supplémentaire de redimensionnement. Enfin, une solution basée sur une DCT 8-bandes est proposée dans [68] : elle consiste à n'utiliser que P sous-bandes sur les 8 disponibles lors de la reconstruction et à modifier le facteur de suréchantillonnage par P. Cette solution est cependant restreinte à une transformation de type DCT et est susceptible de créer des artefacts de Gibbs lors de la reconstruction. Nous verrons dans la suite que nos travaux sont une généralisation de cette solution.

5.3.2 Modification du banc de synthèse

Dans le contexte d'un banc de filtres M-bandes, nous décrivons dans cette section comment modifier le banc de synthèse, afin que ce dernier puisse reconstruire une image d'une résolution réduite d'un facteur rationnel quelconque M/P. Cependant, la discussion sur le redimensionnement d'images par un facteur rationnel nécessite préalablement que l'on définisse clairement cette notion.

Redimensionnement et filtre de rééchantillonnage

Comment redimensionner une image d'un facteur M/P ? La question peut paraître anodine mais le sujet est pourtant vaste : il existe ainsi une littérature abondante sur le sujet [116, 160]. Cependant, la technique habituelle est linéaire et consiste à rééchantillonner l'image sur ses deux dimensions. Dans la suite, nous noterons alors indifféremment $[\downarrow \frac{M}{P}]$ l'opérateur de rééchantillonnage d'un facteur M/P et l'opérateur de redimensionnement de facteur M/P.

L'opérateur de rééchantillonnage $[\downarrow \frac{M}{P}]$ est défini comme la mise en cascade d'un opérateur de sur-échantillonnage de facteur P, d'un filtre de rééchantillonnage w et d'un opérateur de sous-échantillonnage de facteur M. Il permet de réduire la résolution d'un signal ou d'une image d'un facteur M/P et est illustré en Fig. 5.6.

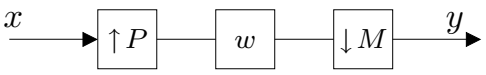


FIG. 5.6 – Opérateur de rééchantillonnage $[\downarrow \frac{M}{P}]$.

L'utilisation d'un filtre de rééchantillonnage w est rendue nécessaire pour éviter les effets de crénelage (*aliasing*) créés par le sous-échantillonnage $[\downarrow M]$. Son choix résulte d'un compromis entre la taille de son support, sa ressemblance avec un filtre passe-bas idéal de fréquence normalisée de coupure P/M et son aptitude à ne pas générer d'artefacts de type *ringing* en cas de quantification à bas débit.

Les filtres w classiquement utilisés en redimensionnement d'image sont le filtre échelon, qui permet une interpolation de type plus proche voisin, le filtre triangle, qui permet une interpolation de type linéaire, le filtre bicubique [65] et le filtre de Lanczos. On remarquera les travaux récents de Seidner [128] sur la construction de filtres de rééchantillonnage optimaux dans le domaine polyphase. Cependant, le choix et la construction d'un filtre w optimal dépasse largement les prétentions de ce document.

Nous nous limiterons dans ce document à l'utilisation du filtre échelon w_e et du filtre triangle w_t. Le filtre échelon w_e de taille P, permet une interpolation de type plus proche voisin et possède la réponse impulsionnelle suivante :

$$w_e[k] = \begin{cases} 1, & \text{si } 0 \leq k < P \\ 0, & \text{sinon} \end{cases} \tag{5.14}$$

Enfin, le filtre triangle w_t de taille $2P-1$, permet une interpolation de type linéaire et sa réponse impulsionnelle est définie par :

$$w_t[k] = \begin{cases} 1 - |k/P|\,, & \text{si } -P < k < P \\ 0, & \text{sinon} \end{cases} \tag{5.15}$$

Modification proposée

Au moyen de l'opérateur de redimensionnement $[\downarrow \frac{M}{P}]$ précédemment défini, nous pouvons alors énoncer la proposition suivante :

Proposition 1 Soit un banc de filtres M-bandes à reconstruction parfaite. Dans le banc de synthèse, il suffit de remplacer les sur-échantillonneurs de facteur M par des sur-échantilloneurs de facteur P et d'utiliser des filtres de synthèse rééchantillonnés par un facteur M/P $\{\widetilde{g}_k = [\downarrow \frac{M}{P}]g_k\}$ à la place des $\{g_k\}$ pour pouvoir reconstruire un signal de sortie qui est exactement la version rééchantillonnée d'un facteur M/P du signal d'entrée : $\tilde{y} = [\downarrow \frac{M}{P}]x$. Cette proposition est illustrée par la Fig. 5.7.

Donnons une preuve de cette proposition. Dans un premier temps, nous calculons la transformée de Fourier du signal $z = [\downarrow \frac{M}{P}]x$ et dans un second temps, celle du signal $\tilde{y}$. Nous montrons alors l'égalité des deux expressions.

Preuve. Calculons tout d'abord la transformée de Fourier de $z = [\downarrow \frac{M}{P}]x$. Au vu de la définition de l'opérateur de rééchantillonnage M/P, elle s'exprime par :

$$\widehat{z}(f) = \frac{1}{M} \sum_{i=0}^{M-1} \widehat{w}\Big(\frac{f+i}{M}\Big)\widehat{x}\Big(\frac{Pf+Pi}{M}\Big)$$

De plus, sachant qu'il existe un entier unique k strictement inférieur à M tel que $Pi = Mq + k$ et à cause de la 1-périodicité de $\widehat{x}$ qui implique que $\widehat{x}(f+q) = \widehat{x}(f)$, nous pouvons alors réécrire $\widehat{z}(f)$ par :

$$\widehat{z}(f) = \frac{1}{M} \sum_{k=0}^{M-1} \alpha_k(f)\widehat{x}\Big(\frac{Pf+k}{M}\Big) \quad \text{avec} \quad \alpha_k(f) = \sum_{\substack{0 \leq i < M \\ Pi \equiv k[M]}} \widehat{w}\Big(\frac{f+i}{M}\Big) \tag{5.16}$$

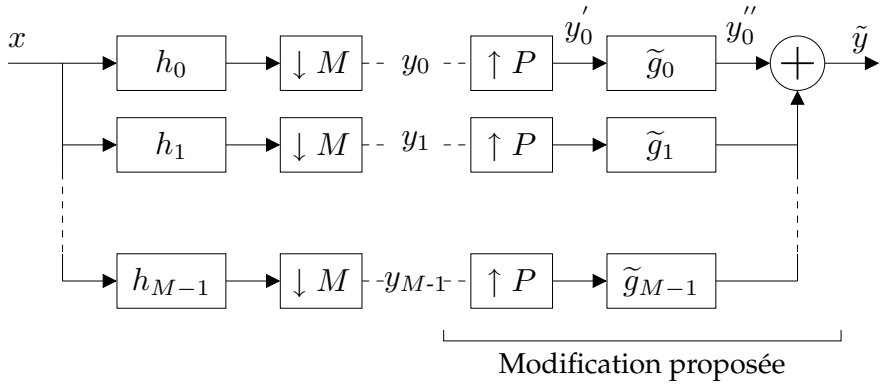


FIG. 5.7 – Banc de filtres M-bandes d'analyse et de synthèse. Ce dernier est en mesure de fournir une image réduite d'un facteur M/P.

Calculons désormais la transformée de Fourier du signal $\widetilde{y}$. Considérons une sous-bande y_i issue du banc de filtres d'analyse. Sa transformée de Fourier $\widehat{y}_i(f)$ peut s'exprimer par :

$$\widehat{y}_i(f) = \frac{1}{M}\sum_{k=0}^{M-1} \widehat{x}\Big(\frac{f+k}{M}\Big)\widehat{h}_i\Big(\frac{f+k}{M}\Big)$$

Après sur-échantillonnage d'un facteur P, nous obtenons alors :

$$\widehat{y}_i^{'}(f) = \frac{1}{M}\sum_{k=0}^{M-1} \widehat{x}\Big(\frac{Pf+k}{M}\Big)\widehat{h}_i\Big(\frac{Pf+k}{M}\Big)$$

Suite au filtrage par les filtres de synthèses modifiés $\widetilde{g}_i = [\downarrow \frac{M}{P}]g_i$, on a :

$$\widehat{y}_i^{''}(f) = \frac{1}{M^2}\Big[\sum_{k=0}^{M-1} \widehat{x}\Big(\frac{Pf+k}{M}\Big)\widehat{h}_i\Big(\frac{Pf+k}{M}\Big)\Big]\Big[\sum_{j=0}^{M-1} \alpha_j(f)\widehat{g}_i\Big(\frac{Pf+j}{M}\Big)\Big]$$

Après permutation des sommes, la transformée de Fourier du signal $\tilde{y}$ vaut donc :

$$\widehat{\tilde{y}}(f) = \sum_{i=0}^{M-1}\widehat{y}_i^{''}(f) = \frac{1}{M^2}\sum_{k=0}^{M-1}\widehat{x}\Big(\frac{Pf+k}{M}\Big)\sum_{j=0}^{M-1}\Big[\alpha_j(f)\sum_{i=0}^{M-1}\widehat{h}_i\Big(\frac{Pf+k}{M}\Big)\widehat{g}_i\Big(\frac{Pf+j}{M}\Big)\Big] \tag{5.17}$$

Or, la condition de reconstruction parfaite (5.4) des filtres M-bandes s'exprime par :

$$\forall f,\ 0 \le i,j < M, \qquad \sum_{k=0}^{M-1}\widehat{h}_i\Big(\frac{f+k}{M}\Big)\widehat{g}_j^{\,*}\Big(\frac{f+k}{M}\Big) = M\delta_{i-j}$$

En permutant les termes de sommation, cette condition peut se réécrire :

$$\forall f,\ 0 \le l,k < M, \qquad \sum_{i=0}^{M-1}\widehat{h}_i\Big(\frac{f+k}{M}\Big)\widehat{g}_i^{\,*}\Big(\frac{f+l}{M}\Big) = M\delta_{l-k} \tag{5.18}$$

L'injection de la condition (5.18) dans l'équation (5.17) nous permet alors d'obtenir l'expression finale de $\widehat{\tilde{y}}(f)$:

$$\widehat{\tilde{y}}(f) = \frac{1}{M}\sum_{k=0}^{M-1}\alpha_k(f)\widehat{x}\Big(\frac{Pf+k}{M}\Big) \tag{5.19}$$

Les expressions (5.16) et (5.19) sont identiques. Nous avons donc $\tilde{y} = [\downarrow \frac{M}{P}]x$.

■

L'image reconstruite par le banc de synthèse modifié est donc l'image originale mais redimensionnée d'un facteur M/P par l'opérateur $z = [\downarrow \frac{M}{P}]x$. Il est ainsi clair que cette modification permet de donner aux bancs de filtres M-bandes une propriété de scalabilité rationnelle d'un facteur variable M/P, pour tout entier P inférieur à M et sans modification du banc d'analyse.

On notera de plus qu'il est possible de reconstruire un signal de taille réduite en utilisant seulement les Q premières sous-bandes $\{y_i\}_{0 \leq i < Q}$ fournies par le banc d'analyse. Dans ce cas, le signal reconstruit P_{V_Q} et l'erreur de reconstruction s'écrivent :

$$P_{V_Q}(x) = \sum_{j=0}^{Q-1} \sum_{k} g_j(n - Mk) y_j(k)$$

$$x - P_{V_Q}(x) = \sum_{j=Q}^{M-1} \sum_{k} g_j(n - Mk) y_j(k)$$

P_{V_Q} est donc un projecteur *oblique* du signal d'entrée x sur l'espace vectoriel $A = \text{Vect}\{(g_j(n - Mk))_{n \in \mathbb{Z}},\ j \in \{0, \ldots, Q-1\},\ k \in \mathbb{Z}\}$. Plus précisément, $x - P_{V_Q}(x)$ est orthogonal à $B = \text{Vect}\{(h_j(n - Mk))_{n \in \mathbb{Z}},\ j \in \{0, \ldots, Q-1\},\ k \in \mathbb{Z}\}$ puisque $< x - P_{V_Q}(x), h_j(n - Mk) >= 0, \forall j \in \{0, \ldots, Q-1\}$. En développant des arguments similaires à ceux utilisés pour montrer l'équation (5.19), nous pouvons déduire que le signal reconstruit à partir des Q premières sous-bandes et en utilisant le banc de synthèse modifié s'écrit alors :

$$\widetilde{z}_Q = [\downarrow \tfrac{M}{P}] P_{V_Q}(x).$$

5.3.3 Complexité théorique

Nous nous proposons dans cette section de calculer la complexité théorique de notre structure avec modification du banc de synthèse et de la comparer à celle d'une structure naïve qui consisterait à reconstruire entièrement le signal puis à le redimensionner. Une telle structure est illustrée par la Fig. 5.8.

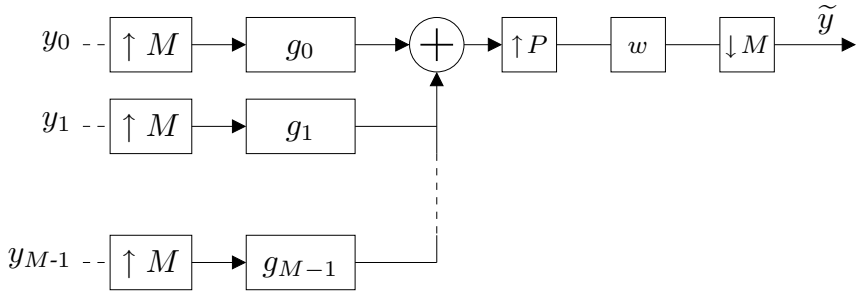


FIG. 5.8 – Reconstruction à pleine résolution par un banc de synthèse M-bandes avec une étape de redimensionnement.

La représentation polyphase [155] permet une implémentation efficace des structures présentées. Tout d'abord, un algorithme rapide pour réaliser l'opération de rééchantillonnage décrite par l'opérateur $[\downarrow \frac{M}{P}]$ en section 5.3.2 possède une complexité de nW/M

multiplications, où n est la taille du signal d'entrée et W la longueur de la réponse impulsionnelle du filtre w. De plus, les opérateurs d'interpolation intervenant dans le banc de synthèse M-bandes consistent au sur-échantillonnage d'un signal d'un facteur M suivi d'une convolution par un filtre d'une réponse impulsionnelle de longueur N. Une implémentation efficace d'un tel opérateur nécessite alors nN multiplications. Si Q sous-bandes sont considérées pour la reconstruction, le banc de synthèse M-bandes possède ainsi une complexité de nQN multiplications.

Nous sommes alors en mesure de calculer les complexités théoriques des deux structures de reconstruction, exprimées en nombre de multiplications pour Q sous-bandes de taille n. Les résultats sont présentés sur le Tab. 5.9. On observe ainsi la réduction théorique de la complexité apportée par notre structure.

Structure	Complexité (en multiplications)
Proposée	$\frac{n}{M}\left(NQ\frac{P}{M}+W\frac{Q}{M}\right)$
Avec post-redimensionnement	$\frac{n}{M}\left(NQ+W\right)$

TAB. 5.9 – Comparaison de la complexité des deux structures.

5.3.4 Résultats expérimentaux

Afin d'illustrer l'intérêt de notre proposition, nous avons conduit plusieurs expérimentations en utilisant des bancs de filtres M-bandes variés, des paramètres P et Q différents et plusieurs filtres de rééchantillonnage w. Dans un premier temps, nous visualisons l'influence des paramètres P, Q et du filtre de rééchantillonnage w. Nous mesurons alors la durée de reconstruction réelle accomplie par notre structure et la comparons à une structure avec post-redimensionnement afin de justifier nos résultats de complexité théorique. Nous présentons enfin des résultats concernant l'efficacité de codage en utilisant plusieurs bancs de filtres et en faisant varier les paramètres P et Q.

Scalabilité rationnelle

Nous nous proposons dans cette section d'utiliser le banc de filtres 8-bandes à 16 échantillons proposé par Alkin [12] qui est bien adapté à la compression d'image et au codage vidéo. En appliquant la modification du banc de synthèse telle que décrite en section 5.3.2 et en utilisant le filtre de rééchantillonnage linéaire w_t, nous pouvons reconstruire une image d'une résolution réduite d'un facteur $8/P$ à partir de Q sous-bandes. Aucun codage n'est effectué dans cette expérimentation. La Fig. 5.9 illustre les résultats obtenus sur *Lena* en faisant varier le paramètre P de 1 à 8 et en utilisant un nombre de sous-bandes Q égal à P. On observe ainsi la finesse de la scalabilité offerte par notre méthode.

Dans une expérimentation sans codage, le nombre de sous-bandes Q permet d'influer sur la qualité de la reconstruction. La Fig. 5.10 montre les résultats obtenus sur *Barbara* en utilisant un paramètre $P = 5$ fixe et un nombre de sous-bandes Q variant de 1 à 8. On observe l'augmentation très rapide de la qualité de l'image en fonction de Q et ceci est très lié à la sélectivité fréquentielle du banc de filtres. Dans la sous-section suivante consacrée au codage, nous verrons l'utilité qu'offre la possibilité de reconstruire l'image en utilisant un nombre réduit de sous-bandes.

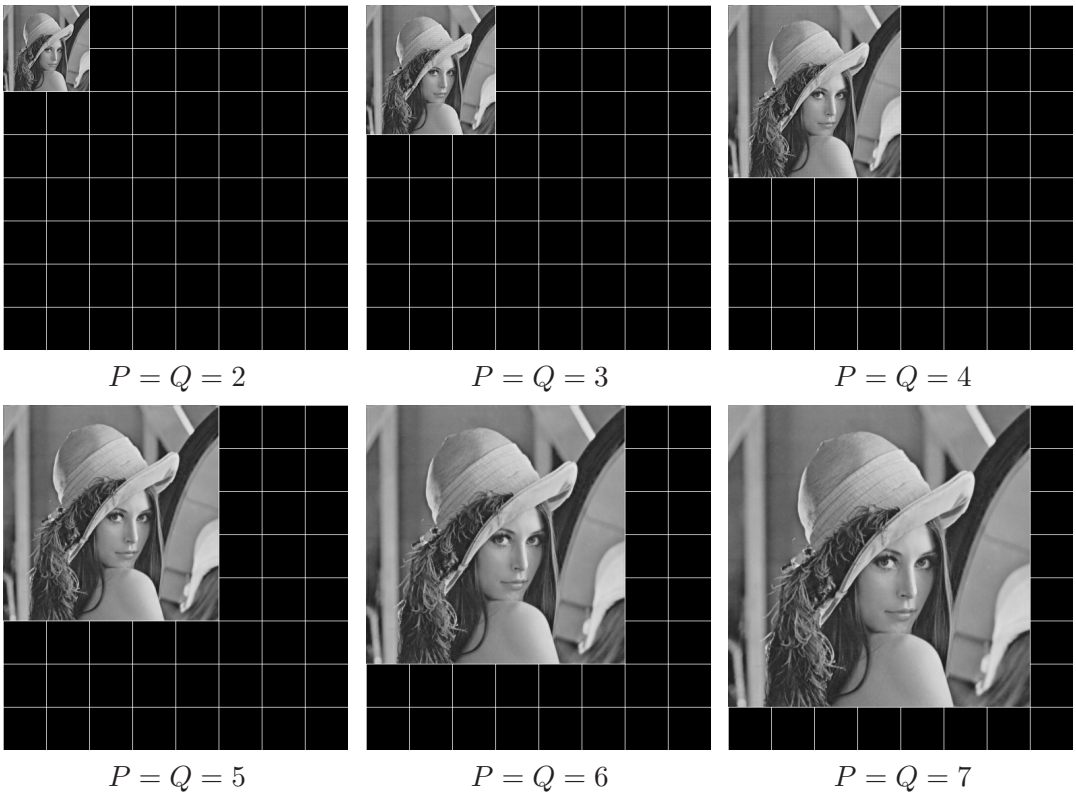


FIG. 5.9 – Illustration de différents facteurs de scalabilité M/P obtenus par la méthode proposée sur l'image *Lena* avec $M = 8$, $P = \{2, 3, 4, 5, 6, 7\}$ et $Q = P$.

Influence du filtre de rééchantillonnage

Quelle est l'influence du filtre de rééchantillonnage w sur les images reconstruites ? Afin de répondre à cette question, nous avons conduit sur *Barbara* des expérimentations avec $P = 5$, un nombre variable de sous-bandes Q et en comparant les filtres de rééchantillonnage échelon w_e et triangle w_t. Les résultats sont présentés en Fig. 5.11. On observe la meilleure qualité visuelle des images obtenues avec le filtre triangle w_t, dû à ses propriétés d'interpolation linéaire. Son utilisation conduit ainsi à l'obtention d'images lisses et dépourvues d'artefacts de crénelage, très visibles au contraire dans le cas $Q = 8,\ w_e$.

Durée de reconstruction

Afin d'illustrer les résultats de complexité théorique, nous nous plaçons encore dans un scénario sans codage, muni de la transformée 8-bandes d'Alkin. Afin de reconstruire une image réduite d'un facteur $8/P$ à partir de Q sous-bandes, nous considérons tout d'abord notre structure avec modification du banc de filtres de synthèse. Nous considérons alors une deuxième structure qui consiste naïvement en la reconstruction du signal à pleine résolution, suivie d'une étape de rééchantillonnage. En réutilisant les paramètres mis en œuvre dans les Fig. 5.9 et 5.10 et le filtre de rééchantillonnage w_t, nous présentons dans les Tabs. 5.10 et 5.11 les durées de reconstruction obtenues avec les deux structures sur un Pentium IV cadencé à 2.8 GHz. On observe clairement la supériorité de notre approche, montrant des durées de reconstruction jusqu'à 3 fois inférieures à la structure naïve.

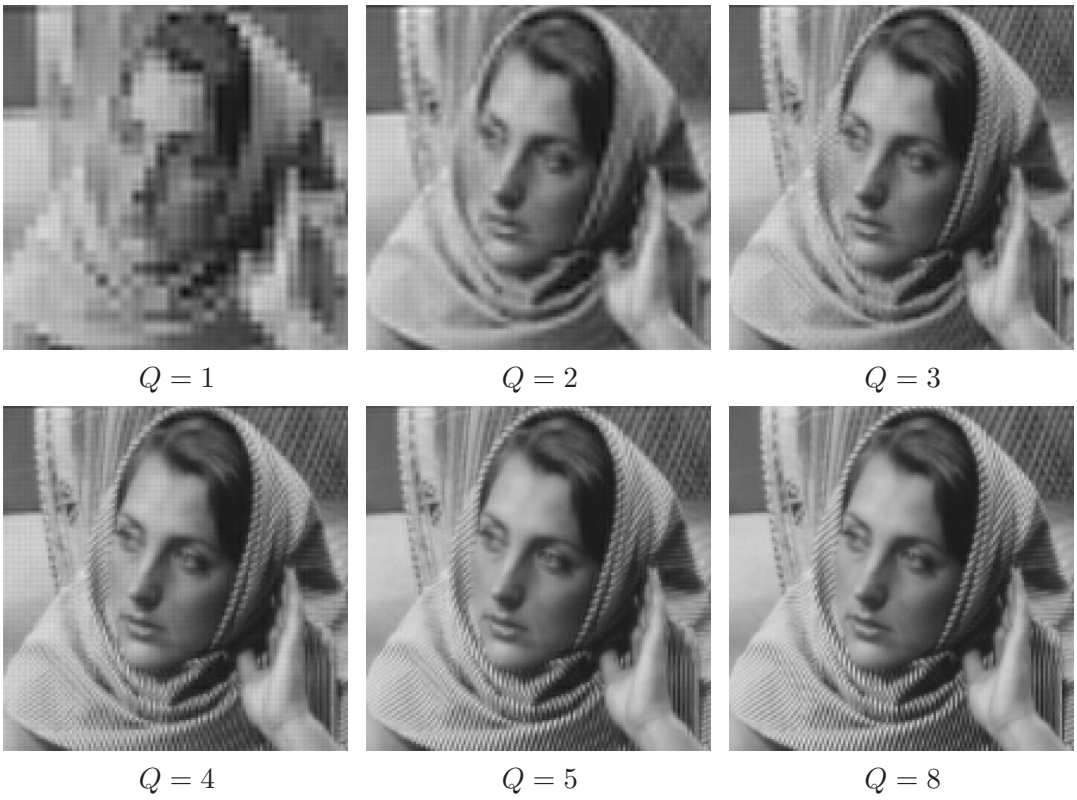


FIG. 5.10 – Illustration de différentes reconstructions en utilisant un nombre variable de sous-bandes Q sur l'image *Barbara* avec $M = 8$, $P = 5$ et $Q = \{1, 2, 3, 4, 5, 8\}$.

Durée (en ms)	$P = 2$	$P = 3$	$P = 4$	$P = 5$	$P = 6$	$P = 7$	$P = 8$
Structure proposée	62.3	77.8	102.3	126.8	162.3	207.4	263.5
Avec rééchantillonnage	168.1	199.8	247.8	283.8	320.5	373.5	262.9

TAB. 5.10 – Durées mesurées pour reconstruire *Lena* au moyen des deux structures présentées et avec les paramètres utilisés dans la Fig. 5.9, $M = 8$ et $Q = P$.

Durée (en ms)	$Q = 1$	$Q = 2$	$Q = 3$	$Q = 4$	$Q = 5$	$Q = 6$	$Q = 8$
Structure proposée	60.5	78.7	95.8	108.1	126.0	143.7	180.5
Avec rééchantillonnage	186.4	214.7	241.3	258.2	284.6	311.9	369.4

TAB. 5.11 – Durées mesurées pour reconstruire *Barbara* au moyen des deux structures présentées et avec les paramètres utilisés dans la Fig. 5.10, $M = 8$, $P = 5$ et plusieurs valeurs de Q.

Efficacité de codage avec reconstruction totale

Notre méthode permet de donner la propriété de scalabilité rationnelle à un banc de filtres M-bandes. Cette propriété peut s'avérer très utile en codage d'image où l'on souhaite créer un train binaire, scalable, unique et capable d'être décodé dans de nombreuses résolutions. Cependant, il n'est pas souhaitable que la propriété de scalabilité rationnelle

$Q = 3,\ w_e$ $Q = 4,\ w_e$ $Q = 8,\ w_e$

$Q = 3,\ w_t$ $Q = 4,\ w_t$ $Q = 8,\ w_t$

FIG. 5.11 – Influence du filtre de rééchantillonnage w_e et w_t, illustrée par différentes reconstructions en utilisant un nombre de variable de sous-bandes Q sur l'image *Barbara* avec $M = 8$, $P = 5$ et $Q = \{3, 4, 8\}$.

nuise à l'efficacité de codage. Afin de répondre à cette question, nous avons souhaité tester dans cette section l'efficacité de codage de plusieurs transformées M-bandes en les comparant à la transformée dyadique 9/7.

Nous avons considéré dans nos expérimentations les transformées 8-bandes suivantes : transformée d'Alkin [12] à 16 échantillons, DCT, LOT, LBT et une transformée en paquets d'ondelettes 9/7 avec une base uniforme. Nous avons de plus testé la transformée dyadique 9/7 dont l'efficacité a été prouvée par le succès du codec JPEG-2000. Les images sont décomposées sur un niveau spatial avec les transformées 8-bandes et sur trois niveaux avec la transformée dyadique, de façon à obtenir une sous-bande spatiale d'approximation de même taille. Les structures de décomposition utilisées pour les transformées 8-bandes et pour la transformée dyadique sont illustrées en Fig. 5.12, où Q et Q' dénotent respectivement le nombre de sous-bandes utilisées pour la reconstruction dans la cas 8-bandes et dans le cas dyadique. Les expérimentations ont été conduites sur le codec JPEG-2000 VM 8.0, en utilisant l'implémentation logicielle de Malvar [81] de la DCT, la LOT et de la LBT. Le rééchantillonnage des filtres est fait au moyen du filtre triangle w_t, jouant le rôle d'interpolateur linéaire.

Nous souhaitons dans un premier temps évaluer l'efficacité de codage offerte par les transformées M-bandes avec reconstruction totale, sans modification du banc de synthèse ($P = 8$) et en utilisant toutes les sous-bandes lors de la reconstruction ($Q = 8$). Les Tabs. 5.12 et 5.13 présentent les résultats obtenus en comparant plusieurs transformées 8-bandes et la transformée dyadique 9/7.

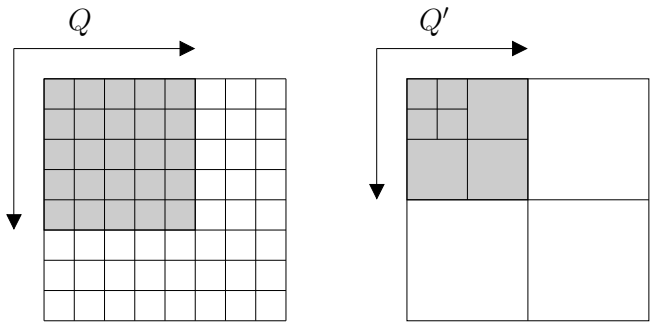


FIG. 5.12 – Bases de décomposition utilisées pour les transformées 8-bandes (gauche) et pour la transformée dyadique 9/7 (droite). Q et Q' dénotent le nombre de sous-bandes utilisées pour la reconstruction.

Débit (bpp)	0.1	0.2	0.4	1.0
9/7-dyadique	24.47	27.02	30.61	37.02
Alkin	24.29	26.63	29.79	35.25
LOT	25.00	27.84	31.34	37.23
LBT	**25.27**	**28.17**	**31.76**	**37.70**
DCT	23.92	26.44	29.78	36.00
9/7-paquets	25.27	28.00	31.38	37.43

TAB. 5.12 – Comparaison débit-PSNR en dB de la transformée dyadique 9/7 et de plusieurs transformées 8-bandes en reconstruisant *Barbara* à pleine résolution ($P = Q = 8$).

Débit (bpp)	0.1	0.2	0.4	1.0
9/7-dyadique	29.62	32.74	**36.05**	**40.36**
Alkin	28.31	31.09	34.06	38.79
LOT	29.06	32.05	35.27	39.47
LBT	29.48	32.63	35.71	39.82
DCT	28.20	31.29	34.73	39.27
9/7-paquets	**29.72**	**32.78**	35.80	39.85

TAB. 5.13 – Comparaison débit-PSNR en dB de la transformée dyadique 9/7 et de plusieurs transformées 8-bandes en reconstruisant *Lena* à pleine résolution ($P = Q = 8$).

On observe que la transformée LBT offre les meilleurs résultats sur *Barbara* tandis que la transformée dyadique 9/7 semble plus efficace sur *Lena*. Il apparaît enfin que ces deux transformées ont la meilleure efficacité globale sur l'ensemble des deux images. Ces résultats sont confirmés sur la Fig. 5.13, où l'on observe la meilleure qualité des images reconstruites avec la transformée dyadique 9/7 et la LBT. Elles surpassent notamment la DCT, qui présente de sévères artefacts de type bloc. En accord avec le codec PTC [84] issu des travaux de Malvar sur la compression d'image par codage emboîté des coefficients LBT, ces résultats montrent que la transformée LBT donne de bons résultats et reste compétitive en comparaison avec la transformée dyadique 9/7.

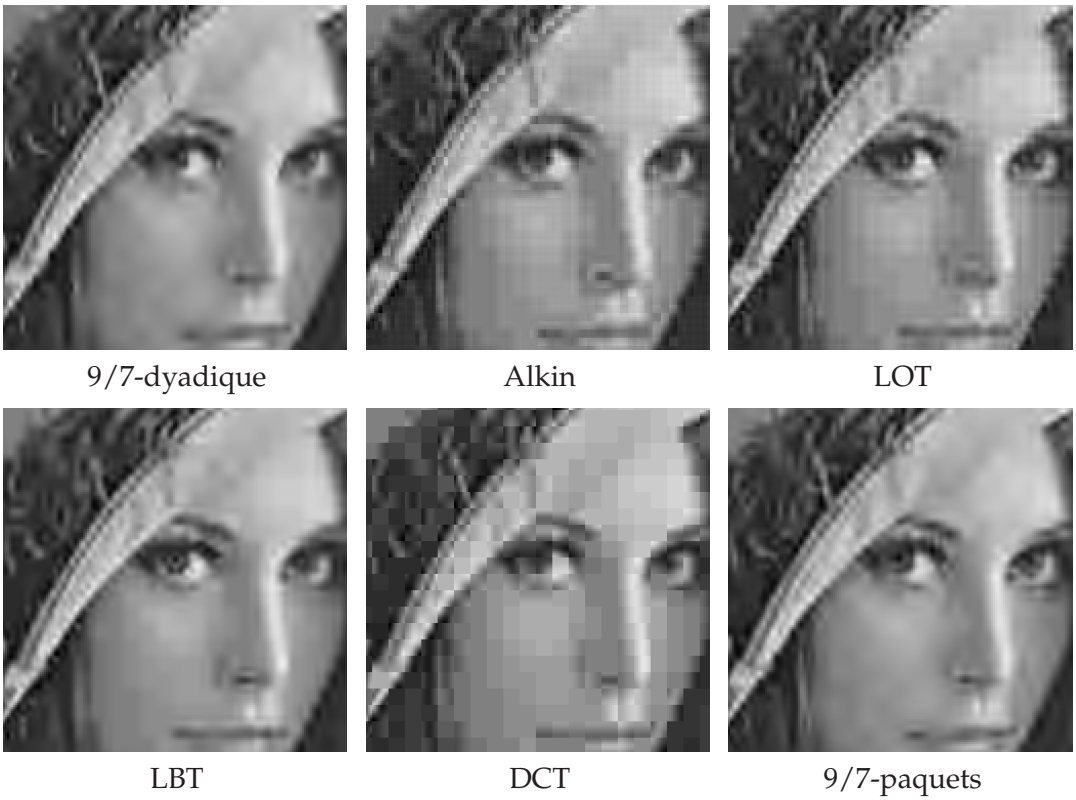


FIG. 5.13 – Zooms sur des images reconstruites à pleine résolution ($P = Q = 8$) de *Lena* avec différentes transformées, à un débit de 0.1 bpp.

Efficacité de codage avec scalabilité rationnelle

Nous souhaitons désormais évaluer l'efficacité de codage offerte par les transformées M-bandes en utilisant la scalabilité rationnelle dans le scénario suivant. Supposons un réseau où un serveur de contenu diffuse une image compressée à destination d'un parc de récepteurs possédant des écrans de tailles différentes. Certains terminaux sont des PDA capables d'afficher des images 512 pixels de largeur tandis que d'autres sont des smartphones limités à 320 ou 192 pixels. Enfin, les récepteurs de type téléphone portables possèdent des écrans de seulement 128 pixels de large. Ces contraintes exigent alors la construction d'un train binaire scalable capable de tenir compte des facteurs de scalabilité rationnels $8/8$, $8/5$, $8/3$ et $8/2$. Nous allons mettre en place deux stratégies pour répondre aux exigences de ce scénario. La première est basée sur une transformée dyadique suivie d'une étape de redimensionnement tandis que la seconde repose sur une transformée M-bandes munie de la scalabilité rationnelle.

Comme vu précédemment, un schéma de codage basé sur une transformée en ondelettes dyadique ne peut pas offrir de manière native des facteurs de scalabilité non dyadiques. De plus, la stratégie Simulcast étant trop coûteuse et le schéma de codage prédictif en couches n'étant pas assez souple, il nous faut utiliser une technique avec post-redimensionnement. Le terminal récepteur est alors capable de recevoir Q' sous-bandes compressées, de reconstruire l'image à pleine résolution, de la redimensionner d'un facteur $8/P$ et l'adapter ainsi à la taille de son écran. Notons qu'à cause des contraintes de la pyramide dyadique, Q' est nécessairement une puissance de deux.

Au contraire, un schéma de codage reposant sur une transformée M-bandes et utilisant notre propriété de scalabilité rationnelle est capable de fournir directement les facteurs de scalabilité exigés par le scénario. Il est alors aisé de créer un train binaire scalable capable d'être décodé aux résolutions citées précédemment. On notera que les décompositions M-bandes offrent une granularité plus fine sur le choix du nombre de bandes : Q peut ainsi être choisi de façon quelconque entre 1 et M.

Nous comparons tout d'abord dans les Tabs. 5.14 et 5.15 l'efficacité de codage des transformées 8-bandes et de la transformée dyadique 9/7 obtenues sur les images *Barbara* et *Lena*, en utilisant $P = Q = 5$. Cette expérimentation est équivalente au scénario proposé où le terminal récepteur ne peut recevoir que $Q = 5$ bandes du train binaire et possède une résolution réduite d'un facteur 8/5 par rapport à la taille de l'image originale. De plus, à cause des restrictions incombant aux décompositions dyadiques, Q' a du être abaissé à 4.

Débit (bpp)	0.1	0.2	0.4	1.0
9/7-dyadique	25.44	26.71	28.10	28.88
Alkin	25.50	27.42	29.08	30.47
LOT	26.18	28.51	30.72	32.43
LBT	**26.51**	**28.79**	**30.90**	**32.48**
DCT	25.10	27.24	29.77	32.18
9/7-paquets	25.62	27.19	28.28	28.93

TAB. 5.14 – Comparaison débit-PSNR en dB de la transformée dyadique 9/7 et de plusieurs transformées 8-bandes en reconstruisant *Barbara* à la résolution 8/5 ($P = Q = 5$).

Débit (bpp)	0.1	0.2	0.4	1.0
9/7-dyadique	30.89	34.23	36.76	38.08
Alkin	29.42	32.35	34.65	36.40
LOT	30.34	33.97	37.45	40.51
LBT	30.72	**34.41**	**38.04**	**41.14**
DCT	29.15	32.82	36.83	40.54
9/7-paquets	**31.01**	34.24	36.56	38.18

TAB. 5.15 – Comparaison débit-PSNR en dB de la transformée dyadique 9/7 et de plusieurs transformées 8-bandes en reconstruisant *Lena* à la résolution 8/5 ($P = Q = 5$).

Nous observons que les transformées 8-bandes surpassent nettement la transformée dyadique 9/7. Les résultats obtenus avec la LBT montrent en particulier un gain allant jusqu'à 3 dB. Ceci peut-être expliqué par le fait que la structure pyramidale de la transformée 9/7 dyadique impose à cette dernière de n'utiliser que $Q' = 4$ bandes alors que les transformées 8-bandes disposent de $Q = 5$ bandes. Le gain observé pourrait ainsi être expliqué par le manque de flexibilité de la structure de la décomposition dyadique.

Nous avons cependant voulu vérifier cette hypothèse en relançant les mêmes simulations avec $P = 5$ mais en utilisant le même nombre de bandes $Q = Q' = 4$. Les résultats obtenus sur *Lena* sont présentés dans le Tab. 5.16. On observe encore la supériorité de l'efficacité de codage des transformées LBT et en paquets d'ondelettes 9/7, comparée à

la transformée dyadique 9/7. Cependant, le gain obtenu est inférieur à celui observé lors de la simulation précédente et peut s'expliquer par la meilleure sélectivité fréquentielle des transformées 8-bandes. En effet, Li avait déjà constaté [75] la mauvaise sélectivité fréquentielle du filtre passe-haut dyadique 9/7 en observant une fuite importante d'énergie dans sa bande de coupure. Cette caractéristique peut alors conduire à une augmentation de l'amplitude des coefficients d'ondelettes, réduisant ainsi l'efficacité globale de codage.

Débit (bpp)	0.1	0.2	0.4	1.0
9/7-dyadique	30.89	34.23	36.76	38.08
Alkin	29.34	31.88	33.54	34.46
LOT	30.34	33.83	36.52	38.28
LBT	30.72	34.19	**37.03**	**38.60**
DCT	29.14	32.64	35.75	37.54
9/7-paquets	**31.01**	**34.24**	36.55	38.09

TAB. 5.16 – Comparaison débit-PSNR en dB de la transformée dyadique 9/7 et de plusieurs transformées 8-bandes en reconstruisant *Lena* à la résolution 8/5 ($P = 5, Q = 4$).

Nous comparons désormais l'efficacité de codage obtenue sur *Lena* avec un coefficient de réduction de $8/3$, avec $P = 3$ et $Q = 4$. Les résultats sont présentés dans le Tab. 5.17. On observe encore la supériorité de l'efficacité de codage des transformées 8-bandes LBT et 9/7-paquets, comparée à la transformée dyadique 9/7.

Débit (bpp)	0.1	0.2	0.4	1.0
9/7-dyadique	30.88	34.14	36.74	38.09
Alkin	29.01	31.52	33.11	33.98
LOT	30.15	33.55	36.22	37.96
LBT	30.65	34.04	**36.83**	**38.36**
DCT	28.70	32.14	35.29	37.24
9/7-paquets	**30.98**	**34.18**	36.53	38.12

TAB. 5.17 – Comparaison débit-PSNR en dB de la transformée dyadique 9/7 et de plusieurs transformées 8-bandes en reconstruisant *Lena* à la résolution 8/3 ($P = 3, Q = 4$).

La Fig. 5.14 montre les images obtenues lors de la reconstruction avec les différentes transformées en utilisant $P = Q = 3$ et un débit de 0.4 bpp. Les résultats sont comparés avec une version de référence non-codée et obtenue par redimensionnement de l'image originale d'un facteur $8/3$, en utilisant le filtre d'interpolation linéaire w_t. Nous observons clairement la meilleure qualité de l'image reconstruite avec la LBT, comparée à celle obtenue avec la transformée dyadique 9/7.

Un autre exemple illustrant la mauvaise sélectivité fréquentielle des filtres mis en jeu dans la transformée 9/7 est présenté en Fig. 5.15. On remarquera sur l'image obtenue avec le filtre 9/7 les artefacts de crénelage très visibles sur le foulard porté par Barbara. De tels artefacts ne sont pas présents en utilisant la transformée LBT.

LBT 9/7-dyadique Référence

FIG. 5.14 – Zooms sur des images reconstruites à la résolution réduite de $8/3$ ($P = Q = 3$) de *Lena* à 0.4 bpp.

LBT 9/7-dyadique Référence

FIG. 5.15 – Zooms sur des images reconstruites à la résolution réduite de $8/6$ ($P = Q = 6$) de *Barbara* à 0.4 bpp.

5.4 Conclusion

Le principe du schéma de codage vidéo $t+2D$ repose sur l'utilisation d'un filtre temporel compensé en mouvement, pour tirer bénéfice de la redondance temporelle des images, suivie d'une décomposition spatiale des sous-bandes résultantes afin d'exploiter leur redondance spatiale. L'utilisation de la transformée biorthogonale 9/7 est justifiée pour la transformation des sous-bandes temporelles d'approximation, ressemblant à des images naturelles, mais ne l'est pas dans le cas des sous-bandes de détail qui comportent de larges zones de texture à hautes fréquences.

De part leur grande sélectivité fréquentielle et leur flexibilité, les bancs de filtres M-bandes sont des candidats idéaux pour décomposer ces sous-bandes temporelles de détail. Dans un premier temps, nous avons construit un banc de filtres 4-bandes symétrique, orthogonal et disposant d'un nombre suffisant de moments nuls pour lui permettre d'assurer une bonne approximation polynomiale. Sa mise en place au sein du codec MC-EZBC a conduit à un gain en PSNR moyen d'environ 0.3 dB par rapport à la transformée 9/7. Nous avons alors procédé à une étude plus complète avec d'autres transformées M-bandes couramment utilisées en compression d'image (LOT, LBT, MLT) : leur utilisation au sein du codec Vidwav a cependant conduit à un gain en PSNR plus modeste.

En dépit de leurs qualités, les bancs de filtres M-bandes ne possèdent cependant pas de propriétés de scalabilité aussi fines que leurs homologues dyadiques et ne peuvent fournir seulement que des facteurs de scalabilité d'ordre M. Afin de pallier à cet inconvénient, nous avons montré qu'une simple modification du banc de filtres de synthèse permet d'obtenir un facteur de scalabilité rationnel quelconque P/M, où P est un entier inférieur à M. Cette propriété autorise ainsi la construction de schémas de décodage dotés de scalabilité fractionnaire et permet de disposer d'une vaste gamme de facteurs de scalabilité, pour n'importe quelle transformée M-bandes et sans nécessiter un changement du banc d'analyse. Nous avons alors montré la réduction en complexité offerte par cette approche, comparée à une stratégie qui consiste à reconstruire entièrement le signal puis à le redimensionner. Enfin, nous avons montré l'intérêt de cette propriété lors d'un scénario de diffusion de contenu à destination d'un parc de récepteurs possédant des écrans de tailles différentes, En effet, l'utilisation de la transformée LBT 8-bandes et de la scalabilité fractionnaire dans ce scénario permet de reconstruire directement l'image dans une résolution réduite, tout en offrant une meilleure efficacité de codage que l'utilisation de la transformée 9/7 couplée à un opérateur de redimensionnement.

Chapitre 6

Filtrage spatial par lifting adaptatif

Nous présentons dans ce chapitre des décompositions en ondelettes adaptatives et non-linéaires capables d'appréhender la nature géométrique et directionnelle des images. Ces transformées sont basées sur des structures lifting où l'opérateur de mise à jour est modifié à chaque échantillon, selon une décision prise en fonction d'un gradient local calculé sur le signal d'entrée. Les décisions ne sont pas transmises dans le flux compressé et nous nous intéressons tout particulièrement à la détermination des conditions nécessaires et suffisantes pour que ces décisions soient reconstruites lors de la synthèse, permettant ainsi la reconstruction parfaite du signal original.

Après avoir brièvement introduit notre problématique, nous rappelons en section 6.1 les travaux de Piella, Heijmans et Pesquet-Popescu [59, 113] sur lesquels sont basés nos décompositions. Les auteurs décrivent une structure lifting où l'opérateur de mise à jour est modifié à chaque échantillon en fonction d'une décision *binaire* prise sur le signal d'entrée. Ces décisions sont prises par seuillage d'une seminorme calculée sur le gradient du signal d'entrée, conduisant ainsi à choisir entre deux filtres de mise à jour. N'étant pas transmises dans le flux compressé, ces décisions doivent être reconstruites lors de la synthèse pour permettre la reconstruction parfaite du signal. Les auteurs montrent alors l'existence de conditions nécessaires et suffisantes liant la prise de décision et les filtres de mise à jour, et permettant cette reconstruction parfaite. Bien que très attractif, cette transformée adaptative n'est cependant pas suffisamment flexible car elle n'autorise qu'un critère de décision *binaire*. Dans un contexte géométrique 2D, il ne peut ainsi servir qu'à discriminer deux évènements géométriques comme un contour et une région homogène.

Afin de pouvoir tenir compte de la richesse et de la variété des images, il est souhaitable de pouvoir utiliser plusieurs critères, laissant ainsi un choix multiple entre plusieurs filtres de mise à jour. Nous proposons ainsi d'étendre le schéma de décomposition adaptatif de Piella à des critères de décisions multivaluées. Dans un premier temps, nous étudions en section 6.2 le cas d'une décision prise par comparaison de deux seminormes. Nous établissons alors les conditions nécessaires et suffisantes sur ce type de décision permettant d'assurer la reconstruction parfaite du signal. Nous étendons alors nos conclusions en section 6.3 sur la comparaison de N seminormes et poursuivons en section 6.4 avec un critère basé sur la comparaison de deux seminormes combinée au critère de seuil.

La multiplicité des valeurs de décisions offertes par ces nouveaux critères nous permet ainsi de construire des transformées inversibles, adaptatives et capables de discriminer plusieurs évènements géométriques 2D : contours horizontaux, verticaux, régions homogènes... Nous présentons alors en section 6.5 plusieurs expérimentations de compression d'image sans perte, basées sur des transformées adaptatives utilisant différents critères de décision. Les travaux présentés dans ce chapitre ont fait l'objet de la publication d'un article de conférence [112] et d'un article de revue [114].

6.1 Mise à jour adaptative avec critère de seuil binaire

6.1.1 Motivation

En dépit de son efficacité de décorrélation et de sa polyvalence, la transformée en ondelettes est toutefois limitée par sa linéarité. Ainsi, lors de l'analyse multirésolution d'une image, l'approximation par des opérateurs linéaires peut causer le floutage de certaines zones d'intérêt comme les contours ou certaines singularités, entraînant alors la perte d'informations importantes. De plus, la transformation en ondelettes séparable n'est pas bien adaptée à la géométrie des images. En supposant l'exemple d'une image constituée de régions lisses et séparées par des courbes régulières par morceaux, on constate que les ondelettes séparables ne "voient" pas la régularité présente le long de la courbe. Ces observations nous poussent ainsi à rechercher des représentations qui s'adaptent mieux aux données. On trouvera alors dans la littérature de nombreuses décompositions [29, 40, 42, 41, 47, 51, 58, 72, 148], fournissant chacune un degré divers d'adaptabilité.

Nous rappelons dans cette section les travaux de Piella, Heijmans et Pesquet-Popescu [59, 113] décrivant une transformée en ondelettes adaptative, inversible et basée sur une étape de mise à jour adaptative. Les décisions *binaires* utilisées dans ce schéma proviennent d'un critère de seuil (*Threshold Criterion*), noté TC et obtenu par seuillage d'une seminorme calculée sur le gradient du signal d'entrée.

6.1.2 Décomposition avec mise à jour adaptative et critère TC

Nous décrivons comment construire une transformée en ondelettes adaptative sous forme lifting, au moyen d'une étape de mise à jour adaptative suivie d'une étape de prédiction fixe. L'adaptabilité du schéma repose sur le choix entre deux filtres de mise à jour différents, dépendant de l'information locale fournie par les sous-bandes d'entrée. Décrivons tout d'abord la structure générale de la décomposition.

Structure générale

Soit un signal d'entrée $x_0 : \mathbb{Z}^d \to \mathbb{R}$ que l'on sépare en deux signaux $x,\ \boldsymbol{y}$, où $\boldsymbol{y}$ peut éventuellement comporter plus d'une sous-bande, par exemple $y_{s_1},\ y_{s_2}, \ldots, y_{s_M}$. Les sous-bandes $x,\ y_{s_1}, \ldots, y_{s_M}$, qui représentent généralement les composantes polyphases du signal à analyser x_0, sont les sous-bandes d'entrée de notre structure. Nous faisons l'hypothèse que la décomposition $x_0 \mapsto (x, \boldsymbol{y})$ est inversible et qu'il est ainsi possible de reconstruire x_0 à partir de ses composantes x et $\boldsymbol{y}$. Tout d'abord, le signal x est mis à jour afin d'obtenir le signal d'approximation x' puis les sous-bandes $y_{s_1}, \ldots, y_{s_M}$ sont alors prédites pour générer le signal de détail $\boldsymbol{y}' = \{{y'}_{s_1}, \ldots, {y'}_{s_M}\}$. Dans notre schéma lifting, seule l'étape de mise à jour est adaptative tandis que l'étape de prédiction est fixe. Ceci implique que le signal $\boldsymbol{y}$ peut être facilement reconstruit à partir de l'approximation x' et du signal de détail $\boldsymbol{y}'$. La reconstruction de x à partir de x' et $\boldsymbol{y}$ est cependant moins triviale.

L'idée de base de notre schéma adaptatif illustré en Fig. 6.1 réside dans le choix du filtre de mise à jour, qui est fait en fonction de l'information fournie localement par les signaux x et $\boldsymbol{y}$. Dans ce schéma, D est une *carte de décision* qui utilise toutes les bandes des signaux d'entrée, c'est à dire $D = D(x, \boldsymbol{y}) = D\big(x, y_{s_1}, \ldots, y_{s_P}\big)$. Elle prend ses valeurs dans un espace binaire $\{0, 1\}$ et gouverne le choix du filtre de mise à jour. Plus précisément, si $d_{\boldsymbol{n}}$

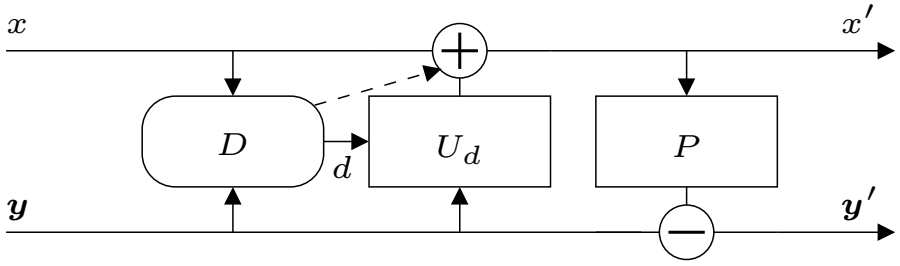


FIG. 6.1 – Structure d'analyse lifting avec mise à jour adaptative.

est la valeur de décision prise par D à la position $\boldsymbol{n} \in \mathbb{Z}^d$, alors l'échantillon $x'(\boldsymbol{n})$ est mis à jour selon la relation :

$$x'(\boldsymbol{n}) = \alpha_{d_{\boldsymbol{n}}} x(\boldsymbol{n}) + \sum_{j=1}^{J} \mu_{d_{\boldsymbol{n}},j} y_j(\boldsymbol{n}) \tag{6.1}$$

avec $y_j(\boldsymbol{n}) = y_{s_j}(\boldsymbol{n}+\boldsymbol{l}_j)\,,\ s_j \in \{s_1, \ldots, s_M\},$ et $\boldsymbol{l}_j \in L$. L'ensemble L définit ici une fenêtre de $\mathbb{Z}^d$ centrée sur l'origine. On notera que les coefficients du filtre de mise à jour $\mu_{d_{\boldsymbol{n}},j}$ dépendent de la valeur de la décision $d_{\boldsymbol{n}} \in \{0, 1\}$ de la carte D au point $\boldsymbol{n}$. Lors de la synthèse, la connaissance de d en chaque point $\boldsymbol{n}$ nous permet alors de reconstruire le signal original x. On dira alors que la *reconstruction parfaite* (PR) est possible.

Critère de seuil TC

Rappelons tout d'abord la définition d'une seminorme. On appelle *seminorme* une fonction $p : \mathbb{R}^N \mapsto \mathbb{R}^+$ vérifiant les propriétés suivantes :

$$\begin{aligned} &(i) \quad \forall\, \lambda \in \mathbb{R},\ \forall\, \boldsymbol{v} \in \mathbb{R}^N,\ p(\lambda \boldsymbol{v}) = |\lambda| p(\boldsymbol{v}) \\ &(ii) \quad \forall\, \boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathbb{R}^N\ p(\boldsymbol{v}_1 + \boldsymbol{v}_2) \leq p(\boldsymbol{v}_1) + p(\boldsymbol{v}_2) \end{aligned}$$

Une seminorme ne vérifie pas $p(\boldsymbol{v}) = 0 \Rightarrow \boldsymbol{v} = 0$ et est donc plus faible qu'une norme.

Dans notre schéma de lifting adaptatif, supposons que les décisions soient prises par seuillage d'une seminorme :

$$D\big(\boldsymbol{v}(\boldsymbol{n})\big) = [p\big(\boldsymbol{v}(\boldsymbol{n})\big) > T]$$

où $[P]$ vaut 1 si le prédicat P est vrai et 0 sinon, p est une seminorme, $T \in \mathbb{R}^+$ une valeur de seuil et $\boldsymbol{v}(\boldsymbol{n}) \in \mathbb{R}^J$ le vecteur gradient dont les composantes sont données par :

$$v_j(\boldsymbol{n}) = x(\boldsymbol{n}) - y_j(\boldsymbol{n})\,, \quad j = 1, \ldots, J$$

Durant le calcul des coefficients selon l'équation (6.1), nous supposons que :

$$\alpha_0 + \sum_{j=1}^{J} \mu_{0,j} = \alpha_1 + \sum_{j=1}^{J} \mu_{1,j} = 1$$

avec $\alpha_d \neq 0$ pour les deux valeurs de décision $d = \{0,\ 1\}$ et $\mu_{0,j} \neq \mu_{1,j}$ pour tous les $j \in \{1, \ldots, J\}$.

Il est alors simple de montrer que le vecteur gradient mis en jeu durant la synthèse $\boldsymbol{v}'(\boldsymbol{n}) \in \mathbb{R}^J$, de composantes $v'_j(\boldsymbol{n}) = x'(\boldsymbol{n}) - y_j(\boldsymbol{n})$ pour $j = 1, \ldots, J$, est lié à $\boldsymbol{v}(\boldsymbol{n})$ par la relation linéaire suivante :

$$\boldsymbol{v}'(\boldsymbol{n}) = A_d \boldsymbol{v}(\boldsymbol{n})$$

où $A_d = I - \boldsymbol{u}\boldsymbol{b}_d^T$, I désigne la matrice identité de taille $J \times J$, $\boldsymbol{u} = (1, \ldots, 1)^T$ et $\boldsymbol{b}_d = (\mu_{d,1}, \ldots, \mu_{d,J})^T$ sont des vecteurs de taille J.

Le filtre de mise à jour adaptatif est alors décrit par :

$$\begin{cases} \boldsymbol{v}' = A_d \boldsymbol{v} \\ d = [p(\boldsymbol{v}) > T] \end{cases} \tag{6.2}$$

où nous avons supprimé l'argument '$\boldsymbol{n}$' de la notation. Notons que le déterminant de la matrice A_d vaut $\det(A_d) = 1 - \boldsymbol{u}^T \boldsymbol{b}_d = 1 - \sum_{j=1}^{J} \mu_{d,j} = \alpha_d$. Notre hypothèse $\alpha_d \neq 0$ rend donc la matrice A_d inversible. Il n'est alors pas difficile de montrer que $A_d^{-1} = I - \boldsymbol{u}\boldsymbol{b}_d'^T$ où $\boldsymbol{b}'_d = -\boldsymbol{b}_d/\alpha_d$.

Considérons l'étape de mise à jour adaptative décrite par l'équation (6.2). Si $p(\boldsymbol{v}) \leq T$ à l'analyse alors la décision vaut $d = 0$ et $\boldsymbol{v}' = A_0 \boldsymbol{v}$. Au contraire, si $p(\boldsymbol{v}) > T$ alors $d = 1$ et $\boldsymbol{v}' = A_1 \boldsymbol{v}$. Pour assurer une reconstruction parfaite du signal original, il est nécessaire de reconstruire la décision d à partir du vecteur gradient $\boldsymbol{v}'$ durant la synthèse. Nous nous restreignons ici au cas où la décision d peut être reconstruite par seuillage de la seminorme $p(\boldsymbol{v}')$, c'est à dire dans le cas :

$$d = [p(\boldsymbol{v}) > T] = [p(\boldsymbol{v}') > T']$$

pour une valeur de $T' > 0$. Nous formalisons cette condition dans le critère suivant.

Critère de seuil TC. Soit un seuil $T > 0$, alors il existe un seuil (probablement différent) $T' > 0$ tel que :

$$\begin{cases} \text{Si } p(\boldsymbol{v}) \leq T \text{ alors } p(A_0 \boldsymbol{v}) \leq T' \\ \text{Si } p(\boldsymbol{v}) > T \text{ alors } p(A_1 \boldsymbol{v}) > T' \end{cases}$$

Il est clair que le critère de seuil (*Threshold Criterion*) TC garantit la propriété de reconstruction parfaite. Nous énonçons dans [59] les conditions nécessaires et suffisantes pour que le TC soit vérifié et analysons différentes seminormes, dont la seminorme quadratique et la seminorme de gradient pondéré.

Avant d'introduire de nouveaux critères de décisions, nous donnons quelques définitions et propriétés qui seront utiles dans la suite du chapitre.

Soit V un espace vectoriel muni de la seminorme p. Pour un opérateur linéaire $A : V \to V$, on définit la *seminorme opérateur* $p(A)$ et la *seminorme opérateur inverse* $p^{-1}(A)$ par :

$$p(A) = \sup\{p(A\boldsymbol{v}) \mid \boldsymbol{v} \in V \text{ et } p(\boldsymbol{v}) = 1\}$$
$$p^{-1}(A) = \sup\{p(\boldsymbol{v}) \mid \boldsymbol{v} \in V \text{ et } p(A\boldsymbol{v}) = 1\}$$

Dans la dernière expression, on utilise la convention $p^{-1}(A) = \infty$ si $p(A\boldsymbol{v}) = 0$ pour tout $\boldsymbol{v} \in V$, si p n'est pas l'opérateur nul. Dans ce ce dernier cas que nous ne considérerons pas dans la suite, les seminormes $p(A)$ et $p^{-1}(A)$ sont toutes deux nulles. Hormis ce cas, on a donc toujours $p^{-1}(A) > 0$. Notons que nous ne pouvons avoir $p(A) = 0$. En effet

si $p(A) = 0$, nous avons alors pour tout $\boldsymbol{v} \in V$, $p(A\boldsymbol{v}) = 0$. Cela signifie donc que pour tout opérateur A inversible et pour tout $\boldsymbol{v} \in V$, $p(\boldsymbol{v}) = 0$, ce qui est en contradiction avec l'hypothèse selon laquelle p n'est pas l'opérateur identiquement nul.

Proposition 2 Soit V un espace de Hilbert, une seminorme p définie dans V et $A : V \to V$ un opérateur linéaire borné. Si $p(A) < \infty$, alors

$$p(A\boldsymbol{v}) \leq p(A)p(\boldsymbol{v}) \quad \text{pour tout } \boldsymbol{v} \in V$$

Preuve. Cette propriété découle directement de la définition de la seminorme opérateur quand $p(\boldsymbol{v}) \neq 0$. Dans le cas $p(\boldsymbol{v}) = 0$, on montre que l'inégalité $p(A) < \infty$ est équivalente à l'implication $p(\boldsymbol{v}) = 0 \Rightarrow p(A\boldsymbol{v}) = 0$. ■

6.2 Comparaison de deux seminormes

Afin de pouvoir tenir compte de la richesse et de la variété des images, il est souhaitable de pouvoir utiliser des cartes de décision multivaluées, avec un critère autorisant plus de flexibilité que le critère de seuil TC. Nous étudions dans cette section un critère basé sur la comparaison de deux seminormes.

6.2.1 Résultats principaux

Nous nous attachons dans cette section à trouver une règle de décision basé sur la comparaison de deux seminormes p_0 et p_1 et assurant la propriété de reconstruction parfaite. Les conditions à l'analyse sont alors données par :

$$\begin{cases} d = 0 & \Leftrightarrow & p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) \\ d = 1 & \Leftrightarrow & p_0(\boldsymbol{v}) > p_1(\boldsymbol{v}) \end{cases} \tag{6.3}$$

où p_0 et p_1 sont deux seminormes non-nulles. Une fois la décision obtenue, l'étape de mise à jour est alors effectuée comme dans (6.1). Durant la synthèse, des conditions similaires sont utilisées en remplaçant $\boldsymbol{v}$ par le vecteur gradient modifié $\boldsymbol{v}' = A_d\boldsymbol{v}$:

$$\begin{cases} (i) & p_0(\boldsymbol{v}') \leq p_1(\boldsymbol{v}') & \Leftrightarrow & d' = 0 \\ (ii) & p_0(\boldsymbol{v}') > p_1(\boldsymbol{v}') & \Leftrightarrow & d' = 1\,. \end{cases} \tag{6.4}$$

Il est clair que la propriété de reconstruction parfaite est assurée si $d = d'$. Les conditions nécessaires et suffisantes dans ce cas sont similaires à celles du critère de seuil TC.

Proposition 3 La reconstruction parfaite PR est assurée si et seulement si les deux conditions suivantes sont satisfaites.

$$\forall \boldsymbol{v} \in V, \qquad \begin{cases} p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) & \Rightarrow & p_0(A_0\boldsymbol{v}) \leq p_1(A_0\boldsymbol{v}) \\ p_0(\boldsymbol{v}) > p_1(\boldsymbol{v}) & \Rightarrow & p_0(A_1\boldsymbol{v}) > p_1(A_1\boldsymbol{v}) \end{cases} \tag{6.5}$$

Preuve. Supposons tout d'abord que l'équation (6.5) soit vérifiée. Montrons alors que la PR est assurée seulement si $d = d'$.

Supposons $p_0(\boldsymbol{v}') \leq p_1(\boldsymbol{v}')$, c'est à dire $d' = 0$ et $d = 1$. Nous avons alors $p_0(A_1\boldsymbol{v}) \leq p_1(A_1\boldsymbol{v})$. Nous obtenons alors de la seconde équation de (6.5), $p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v})$ qui est équivalent à $d = 0$, conformément à (6.3). Cette contradiction avec l'hypothèse de départ $d = 1$ montre alors qu'il est nécessaire d'avoir $d = 0$ pour assurer la reconstruction parfaite.

Supposons désormais que $p_0(\boldsymbol{v}') > p_1(\boldsymbol{v}')$, c'est à dire $d' = 1$ et $d = 0$. Nous avons $p_0(A_0\boldsymbol{v}) > p_1(A_0\boldsymbol{v})$. De la première équation de (6.5), il s'ensuit que $p_0(\boldsymbol{v}) > p_1(\boldsymbol{v})$ et ceci prouve alors la nécessité d'avoir $d = 1$ pour assurer la PR.

Enfin, en supposant maintenant que la PR soit vérifiée, alors les équations (6.3) et (6.4) montrent clairement que les conditions (6.5) sont satisfaites. ■

Remarque 1 Les conditions (6.5) montrent que le domaine de (A_0, A_1), resp. $(\boldsymbol{b}_0, \boldsymbol{b}_1)$ et permettant d'assurer permettant d'assurer la PR est *séparable*. Cela signifie qu'il peut être scindé en un domaine admissible pour A_0, resp. $\boldsymbol{b}_0$ et un autre pour A_1, resp. $\boldsymbol{b}_1$. En outre, la seconde condition de (6.5) peut être réécrite par :

$$\forall \boldsymbol{v} \in V, \quad p_0(A_1\boldsymbol{v}) \leq p_1(A_1\boldsymbol{v}) \;\Rightarrow\; p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v})$$

Comme A_1 est inversible, l'introduction de $\boldsymbol{w} = A_1\boldsymbol{v}$ permet d'aboutir alors à :

$$\forall \boldsymbol{w} \in V, \qquad p_0(\boldsymbol{w}) \leq p_1(\boldsymbol{w}) \;\Rightarrow\; p_0(A_1^{-1}\boldsymbol{w}) \leq p_1(A_1^{-1}\boldsymbol{w})$$

La seconde condition de PR est donc similaire à la première, en remplaçant A_0 par A_1^{-1}. De plus, comme $A_0 = I - \boldsymbol{u}\boldsymbol{b}_0^T$ et $A_1 = I - \boldsymbol{u}\boldsymbol{b'}_1^T$ avec $\boldsymbol{b}'_1 = -\alpha_1^{-1}\boldsymbol{b}_1$, nous avons une symétrie entre les domaines admissibles pour $\boldsymbol{b}_0$ et $\boldsymbol{b}_1$: le second est déduit du premier par le remplacement de $\boldsymbol{b}_0$ par $-\alpha_1^{-1}\boldsymbol{b}_1$.

Introduisons désormais quelques définitions qui nous serons utiles dans la suite :

$$p_{10}(A_0) = \inf\{p_1(A_0\boldsymbol{v}) \;\mid\; \boldsymbol{v} \in V \text{ et } p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) = 1\} \tag{6.6}$$
$$p_{01}(A_1) = \inf\{p_0(A_1\boldsymbol{v}) \;\mid\; \boldsymbol{v} \in V \text{ et } p_1(\boldsymbol{v}) \leq p_0(\boldsymbol{v}) = 1\} \tag{6.7}$$

Afin de définir p_{10}, il est nécessaire que l'ensemble S_{10} défini par :

$$S_{10} = \{\boldsymbol{v} \;\mid\; p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) = 1\}$$

soit non-vide[1]. De façon analogue, l'ensemble $S_{01} = \{\boldsymbol{v} \;\mid\; p_1(\boldsymbol{v}) \leq p_0(\boldsymbol{v}) = 1\}$ doit être non-vide afin de pouvoir définir p_{01}. On remarquera enfin que si les quantités p_{01} et p_{10} existent, elles sont nécessairement finies.

Lemme 1 Les assertions suivantes sont vraies.

(*a*) Si $S_{10} \neq \emptyset$ et $p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v})$, alors $p_1(A_0\boldsymbol{v}) \geq p_{10}(A_0)p_1(\boldsymbol{v})$

(*b*) Si $S_{01} \neq \emptyset$ et $p_1(\boldsymbol{v}) \leq p_0(\boldsymbol{v})$, alors $p_0(A_1\boldsymbol{v}) \geq p_{01}(A_1)p_0(\boldsymbol{v})$

Preuve. Prouvons le cas (a). La preuve de (b) est similaire.

Tout d'abord, si $p_1(\boldsymbol{v}) \neq 0$, nous pouvons choisir $\boldsymbol{v}' = \dfrac{\boldsymbol{v}}{p_1(\boldsymbol{v})}$. Alors, $p_0(\boldsymbol{v}') \leq p_1(\boldsymbol{v}') = \dfrac{p_1(\boldsymbol{v})}{p_1(\boldsymbol{v})} = 1$ et par définition de A_0, on a $p_1(A_0\boldsymbol{v}') \geq p_{10}(A_0)$. On obtient donc $\dfrac{p_1(A_0\boldsymbol{v})}{p_1(\boldsymbol{v})} \geq p_{10}(A_0)$, ce qui implique $p_1(A_0\boldsymbol{v}) \geq p_{10}(A_0)p_1(\boldsymbol{v})$.

Enfin, si $p_1(\boldsymbol{v}) = 0$, alors la condition $p_1(A_0\boldsymbol{v}) \geq 0$ est toujours vérifiée. ■

Notons que les assertions (a) et (b) du lemme précédent impliquent respectivement :

$$p_{10}(A_0) \leq p_1(A_0) \text{ si } p_1(A_0) < \infty \quad \text{et} \quad p_{01}(A_1) \leq p_0(A_1) \text{ si } p_0(A_1) < \infty$$

Le résultat suivant donne les conditions suffisantes pour vérifier les relations (6.5).

[1] La condition $S_{10} = \emptyset$ est en fait équivalente à : $\forall \boldsymbol{v} \in V,\ p_0(\boldsymbol{v}) > p_1(\boldsymbol{v})$ ou $p_1(\boldsymbol{v}) = 0$. Ceci correspond au cas dégénéré : $\forall\ \boldsymbol{v} \in V,\ d = 1$ ou $p_0(\boldsymbol{v}) = p_1(\boldsymbol{v}) = 0$.

Proposition 4 Des *conditions suffisantes* permettant de vérifier les relations (6.5) afin d'assurer la PR sont données par :

$$S_{10} \neq \emptyset\,,\ S_{01} \neq \emptyset \tag{6.8}$$

$$p_{10}(A_0) \geq p_0(A_0) \tag{6.9}$$

$$p_{01}(A_1) \geq p_1(A_1) \tag{6.10}$$

Preuve. Ceci revient à montrer : si les conditions (6.8)-(6.10) sont vérifiées, alors la relation (6.5) doit être satisfaite. Or, du fait de l'existence de p_{10} et p_{01}, ces quantités sont bornées et on a donc $p_d(A_d) < \infty$ pour $d = 0, 1$, conformément à (6.9)-(6.10).

Afin de prouver la première relation de (6.5), nous supposons que $p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v})$. Or, comme $p_0(A_0) < \infty$, nous obtenons $p_0(A_0\boldsymbol{v}) \leq p_0(A_0)p_0(\boldsymbol{v})$ (voir la Proposition 2). Du Lemme 1(a), on a alors $p_1(A_0\boldsymbol{v}) \geq p_{10}(A_0)p_1(\boldsymbol{v})$ et comme $p_{10}(A_0) \geq p_0(A_0)$, nous obtenons $p_1(A_0\boldsymbol{v}) \geq p_0(A_0)p_1(\boldsymbol{v}) \geq p_0(A_0)p_0(\boldsymbol{v}) \geq p_0(A_0\boldsymbol{v})$. On montre ainsi la première relation.

Montrons désormais la seconde relation de (6.5). Supposons $p_0(\boldsymbol{v}) > p_1(\boldsymbol{v})$. Nous savons que $p_1(A_1\boldsymbol{v}) \leq p_1(A_1)p_1(\boldsymbol{v})$. Du Lemme 1(b), nous avons alors $p_0(A_1\boldsymbol{v}) \geq p_{01}(A_1)p_0(\boldsymbol{v})$ et comme $p_{01}(A_1) \geq p_1(A_1) \neq 0$, il s'ensuit que $p_0(A_1\boldsymbol{v}) \geq p_1(A_1)p_0(\boldsymbol{v}) > p_1(A_1)p_1(\boldsymbol{v}) \geq p_1(A_1\boldsymbol{v})$. Ceci implique alors $p_0(A_1\boldsymbol{v}) > p_1(A_1\boldsymbol{v})$ et prouve la deuxième relation de (6.5). ■

On remarquera que si (6.9)-(6.10) sont vérifiées et comme $p_d(A_d) \neq 0$, on a alors nécessairement :

$$p_{10}(A_0) \neq 0 \text{ et } \quad p_{01}(A_1) \neq 0$$

6.2.2 Un cas d'étude : la seminorme pondérée $p(\boldsymbol{v}) = |\boldsymbol{a}^T\boldsymbol{v}|$

Considérons une transformée en ondelettes adaptative en utilisant un critère de décision basé sur la comparaison de deux seminormes. Les règles de décision (6.3) sont alors utilisées pendant l'analyse et les règles (6.4) durant la synthèse. Soient p_0 et p_1 les seminormes pondérées [59] suivantes :

$$p_0(\boldsymbol{v}) = |\boldsymbol{a}_0^T\boldsymbol{v}|, \quad p_1(\boldsymbol{v}) = |\boldsymbol{a}_1^T\boldsymbol{v}|$$

où $\boldsymbol{a}_0 \neq \boldsymbol{0}$ et $\boldsymbol{a}_1 \neq \boldsymbol{0}$. Afin d'étudier les conditions de reconstruction parfaite (6.8)-(6.10) associées à ces seminormes, nous devons calculer $p_{10}(A_0)$ et $p_{01}(A_1)$. Détaillons donc le calcul de $p_{10}(A_0)$. Par définition,

$$p_{10}(A_0) = \inf\{|\boldsymbol{a}_1^T A_0\boldsymbol{v}| \ \mid\ |\boldsymbol{a}_0^T\boldsymbol{v}| \leq |\boldsymbol{a}_1^T\boldsymbol{v}| = 1\}$$

On distingue deux cas : le cas où $\boldsymbol{a}_0$ et $\boldsymbol{a}_1$ *sont* ou *ne sont pas* colinéaires.

(i) Si $\boldsymbol{a}_0$ *et* $\boldsymbol{a}_1$ *sont colinéaires.* Nous pouvons écrire $\boldsymbol{a}_0 = \gamma\boldsymbol{a}_1$ avec $\gamma \in \mathbb{R}^*$. Ce cas n'est cependant pas d'un grand intérêt pratique car il conduit à un schéma non-adaptatif[2].

(ii) Si $\boldsymbol{a}_0$ *et* $\boldsymbol{a}_1$ *ne sont pas colinéaires.* Introduisons $\boldsymbol{c} = A_0^T\boldsymbol{a}_1$, que nous pouvons alors exprimer par :

$$\boldsymbol{c} = c_0\boldsymbol{a}_0 + c_1\boldsymbol{a}_1 + \tilde{\boldsymbol{c}},$$

où $(c_0, c_1) \in \mathbb{R}^2$ et $\tilde{\boldsymbol{c}} \in \mathrm{Span}^{\perp}\{\boldsymbol{a}_0, \boldsymbol{a}_1\}$. Nous obtenons alors :

$$p_1(A_0\boldsymbol{v}) = |\boldsymbol{a}_1^T A_0\boldsymbol{v}| = |\boldsymbol{c}^T\boldsymbol{v}| = |c_0\boldsymbol{a}_0^T\boldsymbol{v} + c_1\boldsymbol{a}_1^T\boldsymbol{v} + \tilde{\boldsymbol{c}}^T\boldsymbol{v}| \geq 0.$$

Afin de calculer $p_{10}(A_0)$, nous devons minimiser l'expression précédente sous la contrainte $|\boldsymbol{a}_0^T\boldsymbol{v}| \leq |\boldsymbol{a}_1^T\boldsymbol{v}| = 1$. Nous obtenons alors le résultat suivant :

[2] En effet, si $\boldsymbol{a}_0 = \gamma\boldsymbol{a}_1$, alors d est fixe : $d = 0$ si $|\gamma| \leq 1$ et $d = 1$ dans le cas contraire. Notons cependant que $p_{10}(A_0)$ est défini seulement si $|\gamma| \leq 1$.

Lemme 2 Soient les seminormes $p_0(\boldsymbol{v}) = |\boldsymbol{a}_0^T\boldsymbol{v}|$ et $p_1(\boldsymbol{v}) = |\boldsymbol{a}_1^T\boldsymbol{v}|$ où $\boldsymbol{a}_0$ et $\boldsymbol{a}_1$ sont deux vecteurs linéairement indépendants, et $\boldsymbol{c} = A_0^T\boldsymbol{a}_1 = c_0\boldsymbol{a}_0 + c_1\boldsymbol{a}_1 + \tilde{\boldsymbol{c}}$. On a alors :

$$p_{10}(A_0) = \begin{cases} |c_1| - |c_0| & \text{if } \boldsymbol{u}^T\boldsymbol{a}_1 \neq 0 \text{ et } \boldsymbol{b}_0 = \dfrac{(1-c_1)\boldsymbol{a}_1 - c_0\boldsymbol{a}_0}{\boldsymbol{u}^T\boldsymbol{a}_1}, \text{ avec } |c_1| > |c_0| \text{ et } \tilde{\boldsymbol{c}} = \boldsymbol{0} \\ 0 & \text{sinon} \end{cases}$$

Preuve. La preuve de ce lemme est rapportée en Annexe A. ■

Un raisonnement similaire conduit au calcul de $p_{01}(A_1)$.

Nous sommes alors en mesure d'étudier les conditions suffisantes de reconstruction parfaite PR (6.8)-(6.10). Modifions tout d'abord légèrement nos notations en introduisant :

$$\boldsymbol{c}_0 = A_0^T\boldsymbol{a}_1 = c_0^0\boldsymbol{a}_0 + c_1^0\boldsymbol{a}_1 + \tilde{\boldsymbol{c}}_0\,, \quad \boldsymbol{c}_1 = A_1^T\boldsymbol{a}_0 = c_0^1\boldsymbol{a}_0 + c_1^1\boldsymbol{a}_1 + \tilde{\boldsymbol{c}}_1,$$

où $(\tilde{\boldsymbol{c}}_0, \tilde{\boldsymbol{c}}_1) \in \mathrm{Span}^{\perp}\{\boldsymbol{a}_0, \boldsymbol{a}_1\}$.

Supposons que les conditions (6.8)-(6.10) soient vérifiées. On a alors $p_{10}(A_0) \neq 0$, $p_{01}(A_1) \neq 0$ et $p_d(A_d) < \infty$. De cette dernière condition, nous obtenons [59] alors : soit $\boldsymbol{u}^T\boldsymbol{a}_d = 0$ et ceci implique $p_d(A_d) = 1$, ou soit $\boldsymbol{u}^T\boldsymbol{a}_d \neq 0$, $\boldsymbol{b}_d = \gamma_d\boldsymbol{a}_d$ avec $\gamma_d \in \mathbb{R}$ et conduit alors à $p_d(A_d) = |\alpha_d|$.

D'un autre côté et conformément au Lemme 2, nous avons une équivalence entre le fait que $p_{10}(A_0) \neq 0$ (resp. $p_{01}(A_1) \neq 0$) et l'expression de $\boldsymbol{b}_0$ (resp. $\boldsymbol{b}_1$) :

$$\boldsymbol{b}_0 = \frac{(1-c_1^0)\boldsymbol{a}_1 - c_0^0\boldsymbol{a}_0}{\boldsymbol{u}^T\boldsymbol{a}_1}, \text{ avec } \boldsymbol{u}^T\boldsymbol{a}_1 \neq 0, |c_1^0| > |c_0^0| \text{ et } \tilde{\boldsymbol{c}}_0 = \boldsymbol{0}$$

conduisant ainsi à $p_{10}(A_0) = |c_1^0| - |c_0^0|$.

En écartant les contraintes non-compatibles pour satisfaire les conditions (6.9)-(6.10) de la Proposition 4, nous obtenons :

$$\begin{aligned} &\boldsymbol{u}^T\boldsymbol{a}_0 \neq 0, \quad \boldsymbol{u}^T\boldsymbol{a}_1 \neq 0 \\ &\boldsymbol{b}_0 = \gamma_0\boldsymbol{a}_0 = \frac{(1-c_1^0)\boldsymbol{a}_1 - c_0^0\boldsymbol{a}_0}{\boldsymbol{u}^T\boldsymbol{a}_1}, \text{ avec } |c_1^0| > |c_0^0| \\ &\boldsymbol{b}_1 = \gamma_1\boldsymbol{a}_1 = \frac{(1-c_0^1)\boldsymbol{a}_0 - c_1^1\boldsymbol{a}_1}{\boldsymbol{u}^T\boldsymbol{a}_0}, \text{ avec } |c_0^1| > |c_1^1| \end{aligned}$$

En rappelant l'hypothèse selon laquelle $\boldsymbol{a}_0$ et $\boldsymbol{a}_1$ ne sont pas colinéaires, nous obtenons finalement :

$$c_1^0 = c_0^1 = 1, \quad |c_0^0| < 1,\ |c_1^1| < 1$$

Et par conséquent :

$$\boldsymbol{b}_0 = -\frac{c_0^0}{\boldsymbol{u}^T\boldsymbol{a}_1}\boldsymbol{a}_0, \qquad \boldsymbol{b}_1 = -\frac{c_1^1}{\boldsymbol{u}^T\boldsymbol{a}_0}\boldsymbol{a}_1$$

Ainsi $p_{01}(A_1) = 1 - |c_1^1|$ et $p_{10}(A_0) = 1 - |c_0^0|$. En rassemblant ces conditions, nous montrons alors le résultat suivant :

Proposition 5 Des conditions suffisantes permettant d'assurer la reconstruction parfaite avec un critère de décision basé sur la comparaison de deux seminormes, $p_0(\boldsymbol{v}) = |\boldsymbol{a}_0^T\boldsymbol{v}|$ et $p_1(\boldsymbol{v}) = |\boldsymbol{a}_1^T\boldsymbol{v}|$, où $\boldsymbol{a}_0$ et $\boldsymbol{a}_1$ sont linéairement indépendants, sont données par :

$$\boldsymbol{u}^T\boldsymbol{a}_0 \neq 0,\ \boldsymbol{u}^T\boldsymbol{a}_1 \neq 0 \text{ et } \boldsymbol{b}_0 = \frac{\beta_0}{\boldsymbol{u}^T\boldsymbol{a}_1}\boldsymbol{a}_0, \quad \boldsymbol{b}_1 = \frac{\beta_1}{\boldsymbol{u}^T\boldsymbol{a}_0}\boldsymbol{a}_1 \tag{6.11}$$

où $0 < |\alpha_0| \leq 1 - |\beta_0|$ et $0 < |\alpha_1| \leq 1 - |\beta_1|$.

Les deux dernières conditions proviennent des inégalités $1 - |\beta_0| = p_{10}(A_0) \geq p_0(A_0) = |\alpha_0|$ et $1 - |\beta_1| = p_{01}(A_1) \geq p_1(A_1) = |\alpha_1|$.

Exemple 1 Considérons les vecteurs de pondérations $\boldsymbol{a}_0 = (1, 0, 1, 0)^T$ et $\boldsymbol{a}_1 = (0, 1, 0, 1)^T$. Les résultats énoncés précédemment sont vrais dans le cas d'un signal mono-dimensionnel mais aussi dans le cas de signaux multidimensionnels, où le gradient correspondra ainsi à différents voisinages. Dans le cas 2D et en considérant le voisinage illustré sur la Fig. 6.2, les vecteurs $\boldsymbol{a}_0$ et $\boldsymbol{a}_1$ nous conduisent alors aux seminormes suivantes :

$$p_0(\boldsymbol{v}) = |v_1 + v_3| \quad \text{et} \quad p_1(\boldsymbol{v}) = |v_2 + v_4| \tag{6.12}$$

qui sont respectivement apparentées à un filtre horizontal et vertical d'ordre deux.

$y_6(\boldsymbol{n})$	$y_2(\boldsymbol{n})$	$y_5(\boldsymbol{n})$
$y_3(\boldsymbol{n})$	$x(\boldsymbol{n})$	$y_1(\boldsymbol{n})$
$y_7(\boldsymbol{n})$	$y_4(\boldsymbol{n})$	$y_8(\boldsymbol{n})$

FIG. 6.2 – Indexation des échantillons dans une fenêtre 3×3 centrée sur $x(\boldsymbol{n})$.

Ces seminormes possèdent l'interprétation géométrique suivante. Si $p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v})$ et par conséquent $d = 0$, alors la dérivée verticale $2x - y_2 - y_4$ est dominante en valeur absolue par rapport à la dérivée horizontale $2x - y_1 - y_3$. Il est alors préférable d'appliquer le filtre de mise à jour sur x dans la direction horizontale.

Comme démontré dans la Proposition 5, la reconstruction parfaite est possible si :

$$\boldsymbol{b}_0 = \frac{\beta_0}{2}(1, 0, 1, 0)^T \; \boldsymbol{b}_1 = \frac{\beta_1}{2}(0, 1, 0, 1)^T$$

avec $0 \leq \beta_0, \beta_1 < 1$.

Plus généralement, soit $\mathcal{D}_0$ un sous-ensemble de $\{1, \ldots, K\}$ avec $K \in \mathbb{N}^*$. Soit $\boldsymbol{a}_0 = (a_0(k))_{1 \leq k \leq K}$ et $\boldsymbol{a}_1 = (a_1(k))_{1 \leq k \leq K}$ avec pour tout $k \in \{1, \ldots, K\}$,

$$a_0(k) = \begin{cases} 1 & \text{si } k \in \mathcal{D}_0 \\ 0 & \text{sinon} \end{cases} \qquad a_1(k) = \begin{cases} 1 & \text{si } k \notin \mathcal{D}_0 \\ 0 & \text{sinon} \end{cases}$$

En d'autres mots, cela revient à supposer que les composantes de $\boldsymbol{a}_0$ et $\boldsymbol{a}_1$ sont complémentaires en représentation binaire. Cette caractéristique permet alors de comparer les décisions prises sur deux ensembles disjoints de voisins $\{y_k,\, k \in \mathcal{D}_0\}$ et $\{y_k,\, k \notin \mathcal{D}_0\}$. Par exemple, le choix de :

$$\boldsymbol{a}_0 = (1, 0, 1, 0, \ldots, 0)^T \in \mathbb{R}^{2K}, \quad \boldsymbol{a}_1 = (0, 1, 0, 1 \ldots, 1)^T \in \mathbb{R}^{2K}$$

permet de chercher le plus faible gradient entre $Kx - \sum_{k=0}^{K} y_{2k+1}$ et $Kx - \sum_{k=0}^{K} y_{2k}$.

Une condition suffisante pour assurer la reconstruction parfaite dans ce cas plus général est obtenue en appliquant la Proposition 5 :

$$\boldsymbol{b}_0 = \frac{\beta_0}{K_1}\boldsymbol{a}_0, \quad \boldsymbol{b}_1 = \frac{\beta_1}{K_0}\boldsymbol{a}_1$$

avec $K_0 = \operatorname{card}\mathcal{D}_0$, $K_1 = K - K_0$, $0 \leq \beta_0 < 1$ et $0 \leq \beta_1 < 1$.

Dans ce cas, le filtrage de mise à jour est proportionnel à la moyenne arithmétique des échantillons voisins. Nous pouvons alors prendre par exemple α_0 égal aux coefficients du filtre de mise à jour, c'est à dire $\alpha_0 = \beta_0/K_0$ et obtenons ainsi :

$$\beta_0 = \frac{K_0}{K_0 + 1}$$

De façon similaire, nous avons $\alpha_1 = \beta_1/K_1$ et $\beta_1 = K_1/(K_1 + 1)$.

Exemple 2 Considérons désormais les vecteurs $\boldsymbol{a}_0 = (1, 1, 0, 0)^T$ et $\boldsymbol{a}_1 = (1/2, 1/2, 1/2, 1/2)^T$. Comparons leur seminormes associées :

$$p_0(\boldsymbol{v}) = |v_1 + v_2| \quad \text{et} \quad p_1(\boldsymbol{v}) = \frac{|v_1 + v_2 + v_3 + v_4|}{2}$$

Ce qui, en d'autres termes, revient à chercher quelle moyenne entre $(y_1 + y_2)/2$ et $(y_1 + y_2 + y_3 + y_4)/4$ est la plus proche de l'échantillon x à mettre à jour.

La Proposition 5 nous permet alors de déterminer les coefficients des filtres de mise à jour (6.11) permettant d'assurer la reconstruction parfaite. Ils sont donnés par :

$$\boldsymbol{b}_0 = \frac{\beta_0}{2}(1, 1, 0, 0)^T, \quad \boldsymbol{b}_1 = \frac{\beta_1}{4}(1, 1, 1, 1)^T,$$

où nous avons encore $0 \leq \beta_0, \beta_1 < 1$.

Plus généralement, il est possible de montrer que nous pouvons comparer deux "moyennes" calculées sur des voisinages emboîtés arbitrairement, tout en assurant la reconstruction parfaite. Soit $\mathcal{D}_0 \neq \emptyset \subset \{1, \ldots, K\}$, avec $K \in \mathbb{N}^*$ et les seminormes p_0 et p_1, définies par :

$$p_0(\boldsymbol{v}) = |x - \frac{\sum_{k\in\mathcal{D}_0} y_k}{K_0}| \text{et} \quad p_1(\boldsymbol{v}) = |x - \frac{\sum_{k=1}^{K} y_k}{K}| \quad \text{avec } K_0 = \operatorname{card}\mathcal{D}_0\,.$$

Alors, $\boldsymbol{a}_0 = (a_0(k))_{1\leq k\leq K}$ où pour tout $k \in \{1, \ldots, K\}$, nous avons :

$$a_0(k) = \begin{cases} \frac{1}{K_0} & \text{si } k \in \mathcal{D}_0 \\ 0 & \text{sinon,} \end{cases}$$

et $\boldsymbol{a}_1 = \boldsymbol{u}/K$. En accord avec la Proposition 5, nous obtenons $\boldsymbol{b}_0 = \beta_0\boldsymbol{a}_0$ et $\boldsymbol{b}_1 = \beta_1\boldsymbol{a}_1$, avec $0 \leq \beta_0, \beta_1 < 1$.

Contre-exemple : Sélection entre des filtres horizontaux et verticaux

La Proposition 5 (et plus généralement la Proposition 4) donne des conditions *suffisantes* permettant d'assurer la reconstruction parfaite. Cependant et comme nous allons le voir dans ce contre-exemple, ces conditions ne sont pas *nécessaires*.

Considérons une nouvelle fois les critères de décision introduits au début de l'Exemple 1. Les deux seminormes p_0 et p_1 définies par (6.12) mesurent ainsi le gradient local horizontal et vertical d'ordre deux.

Supposons que les filtres de mise à jour U_d possèdent un support de 4 échantillons en correspondance avec les coefficients de détail indicés par $y_1, \ldots, y_4$. Les coefficients des filtres $\boldsymbol{b}_d$ sont alors choisis afin de vérifier :

$$\boldsymbol{b}_d = (\mu_d, \eta_d, \mu_d, \eta_d)^T \text{ pour } d = 0, 1 \text{ avec } (\mu_d,\ \eta_d) \in \mathbb{R}^2 \tag{6.13}$$

Ce choix signifie en particulier que seuls les quatre voisins horizontaux et verticaux y_1, y_2, y_3 et y_4 sont utilisés pour mettre à jour le signal d'approximation. Par exemple, si $d = 0$, alors l'opération de mise à jour se réduit à :

$$x' = \alpha_0 x + \mu_0(y_1 + y_3) + \eta_0(y_2 + y_4) \tag{6.14}$$

Soit $\boldsymbol{v}'$ le vecteur gradient mis en jeu lors de la synthèse, c'est à dire $v'_j = x' - y_j$. Un calcul direct montre que :

$$\begin{aligned} |v'_1 + v'_3| &= |(1 - 2\mu_d)(v_1 + v_3) - 2\eta_d(v_2 + v_4)| \\ |v'_2 + v'_4| &= |-2\mu_d(v_1 + v_3) + (1 - 2\eta_d)(v_2 + v_4)| \end{aligned}$$

Si nous pouvons choisir les coefficients μ_0, η_0, μ_1 et η_1 de telle sorte que :

$$p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) \iff p_0(\boldsymbol{v}') \leq p_1(\boldsymbol{v}')$$

il est alors possible de retrouver la décision à partir du vecteur gradient durant la synthèse, rendant ainsi possible la reconstruction parfaite.

Proposition 6 Soient p_0 et p_1 les seminormes définies par (6.12) et considérons les filtres de mise à jour donnés par la relation (6.13). Alors, les conditions nécessaires et suffisantes pour assurer la reconstruction parfaite sont données par :

$$\begin{aligned} \eta_0 \leq \mu_0 \quad \text{et} \quad \mu_0 + \eta_0 < \frac{1}{2}\,, \\ \mu_1 \leq \eta_1 \quad \text{et} \quad \mu_1 + \eta_1 < \frac{1}{2} \end{aligned}$$

Preuve. La preuve est rapportée en Annexe A. ■

Ces conditions sont clairement moins restrictives que celles déduites de la Proposition 5 car il n'est pas nécessaire que les vecteurs $\boldsymbol{b}_0$ et $\boldsymbol{b}_1$ soient respectivement colinéaires à $\boldsymbol{a}_0$ et $\boldsymbol{a}_1$. La condition suffisante (se conférer à l'Exemple 1) donne les intervalles $\eta_0 = 0$, $\mu_0 \in [0, \frac{1}{2})$ et $\mu_1 = 0$, $\eta_1 \in [0, \frac{1}{2})$, qui sont les plus grands intervalles contenus dans le domaine admissible (μ_0, η_0) et (μ_1, η_1).

6.3 Comparaison de N seminormes

Afin d'étendre les résultats obtenus dans la section précédente, nous nous intéressons maintenant à la comparaison de N seminormes entre elles, p_0, $p_1, \ldots,\ p_{N-1}$. On attribue alors en chaque point une valeur de décision d correspondant à l'indice de la seminorme

possédant la plus petite amplitude. La carte de décision à l'analyse comme à la synthèse n'est donc plus binaire et peut prendre désormais N valeurs, $d(\boldsymbol{v}) \in \{0, 1, \ldots, N-1\}$.

Durant l'analyse et sans perte de généralités, si $\min\{p_0(\boldsymbol{v}), p_1(\boldsymbol{v}), \ldots, p_{N-2}(\boldsymbol{v})\} < p_{N-1}(\boldsymbol{v})$, alors nous sommes dans l'un des cas suivants :

$$\left\{\begin{array}{lcl} p_0(\boldsymbol{v}) & < & p_{N-1}(\boldsymbol{v}) \\ p_0(\boldsymbol{v}) & \leq & p_1(\boldsymbol{v}) \\ & \vdots & \\ p_0(\boldsymbol{v}) & \leq & p_{N-2}(\boldsymbol{v}) \end{array}\right. \quad \Leftrightarrow \quad d(\boldsymbol{v}) = 0$$

$$\left\{\begin{array}{lcl} p_1(\boldsymbol{v}) & < & p_{N-1}(\boldsymbol{v}) \\ p_1(\boldsymbol{v}) & < & p_0(\boldsymbol{v}) \\ p_1(\boldsymbol{v}) & \leq & p_2(\boldsymbol{v}) \\ & \vdots & \\ p_1(\boldsymbol{v}) & \leq & p_{N-2}(\boldsymbol{v}) \end{array}\right. \quad \Leftrightarrow \quad d(\boldsymbol{v}) = 1$$

et de même jusqu'à :

$$\left\{\begin{array}{lcl} p_{N-2}(\boldsymbol{v}) & < & p_{N-1}(\boldsymbol{v}) \\ p_{N-2}(\boldsymbol{v}) & < & p_0(\boldsymbol{v}) \\ p_{N-2}(\boldsymbol{v}) & < & p_1(\boldsymbol{v}) \\ & \vdots & \\ p_{N-2}(\boldsymbol{v}) & < & p_{N-3}(\boldsymbol{v}) \end{array}\right. \quad \Leftrightarrow \quad d(\boldsymbol{v}) = N-2\,,$$

Si $\min\{p_0(\boldsymbol{v}), p_1(\boldsymbol{v}), \ldots, p_{N-2}(\boldsymbol{v})\} \geq p_{N-1}(\boldsymbol{v})$, nous avons :

$$\left\{\begin{array}{lcl} p_{N-1}(\boldsymbol{v}) & \leq & p_0(\boldsymbol{v}) \\ p_{N-1}(\boldsymbol{v}) & \leq & p_1(\boldsymbol{v}) \\ & \vdots & \\ p_{N-1}(\boldsymbol{v}) & \leq & p_{N-2}(\boldsymbol{v}) \end{array}\right. \quad \Leftrightarrow \quad d(\boldsymbol{v}) = N-1\,.$$

Durant la synthèse, nous obtenons les mêmes conditions en remplaçant $\boldsymbol{v}$ par $\boldsymbol{v}' = A_d\,\boldsymbol{v}$.

Proposition 7 Des conditions *suffisantes* pour assurer la reconstruction parfaite lors de la comparaison de N seminormes sont données par :

$$\left\{\begin{array}{lcl} p_0(\boldsymbol{v}) & < & p_{N-1}(\boldsymbol{v}) \\ p_0(\boldsymbol{v}) & \leq & p_1(\boldsymbol{v}) \\ & \vdots & \\ p_0(\boldsymbol{v}) & \leq & p_{N-2}(\boldsymbol{v}) \end{array}\right. \Rightarrow \left\{\begin{array}{lcl} p_0(A_0\,\boldsymbol{v}) & < & p_{N-1}(A_0\,\boldsymbol{v}) \\ p_0(A_0\,\boldsymbol{v}) & \leq & p_1(A_0\,\boldsymbol{v}) \\ & \vdots & \\ p_0(A_0\,\boldsymbol{v}) & \leq & p_{N-2}(A_0\,\boldsymbol{v}) \end{array}\right.$$

$$\vdots$$

$$\left\{\begin{array}{lcl} p_{N-1}(\boldsymbol{v}) & \leq & p_0(\boldsymbol{v}) \\ p_{N-1}(\boldsymbol{v}) & \leq & p_1(\boldsymbol{v}) \\ & \vdots & \\ p_{N-1}(\boldsymbol{v}) & \leq & p_{N-2}(\boldsymbol{v}) \end{array}\right. \Rightarrow \left\{\begin{array}{lcl} p_{N-1}(A_{N-1}\,\boldsymbol{v}) & \leq & p_0(A_{N-1}\,\boldsymbol{v}) \\ p_{N-1}(A_{N-1}\,\boldsymbol{v}) & \leq & p_1(A_{N-1}\,\boldsymbol{v}) \\ & \vdots & \\ p_{N-1}(A_{N-1}\,\boldsymbol{v}) & \leq & p_{N-2}(A_{N-1}\,\boldsymbol{v}) \end{array}\right.$$

Preuve. Considérons la première implication de la proposition. Supposons que $d' = 0$. Si $d \neq 0$, alors d peut prendre n'importe quelle valeur parmi $\{1, \ldots, N-1\}$. Prenons par exemple $d = 1$. Par hypothèse, il s'ensuit que :

$$\begin{cases} p_1(A_1\,\boldsymbol{v}) & < & p_{N-1}(A_1\,\boldsymbol{v}) \\ p_1(A_1\,\boldsymbol{v}) & < & p_0(A_1\,\boldsymbol{v}) \\ p_1(A_1\,\boldsymbol{v}) & \leq & p_2(A_1\,\boldsymbol{v}) \\ & \vdots & \\ p_1(A_1\,\boldsymbol{v}) & \leq & p_{N-2}(A_1\,\boldsymbol{v}) \end{cases}$$

La seconde inégalité de ce système $p_1(A_1\,\boldsymbol{v}) < p_0(A_1\,\boldsymbol{v})$ est en contradiction avec l'hypothèse $p_0(A_d\,\boldsymbol{v}) \leq p_1(A_d\,\boldsymbol{v})$ quand $d = 1$. Un argument similaire peut être utilisé pour toutes les valeurs de d $\in \{2, \ldots, N-1\}$. Nous avons alors prouvé par l'absurde que $d = 0$. Les autres implications se démontrent de la même façon. ■

Cette proposition apparaît comme une conséquence du résultat général suivant :

Proposition 8 Considérons à l'analyse, la carte de décision d définie par :

$$\begin{aligned} d : V &\to \{0, \ldots, N-1\} \\ \boldsymbol{v} &\mapsto d(\boldsymbol{v}) \end{aligned}$$

Considérons les régions de décisions $\mathcal{D}_i$, formant une partition de V :

$$\forall i \in \{0, \ldots, N-1\}, \quad \mathcal{D}_i = \{\boldsymbol{v} \in V \mid d(\boldsymbol{v}) = i\} \tag{6.15}$$

$$\text{avec } V = \bigcup_{i=0}^{N-1} \mathcal{D}_i, \quad \mathcal{D}_i \neq \emptyset \quad \text{et} \quad \mathcal{D}_i \cap \mathcal{D}_j = \phi \quad \text{si} \quad i \neq j \tag{6.16}$$

Durant la synthèse, nous avons $\boldsymbol{v}' = A_{d(v)}\,\boldsymbol{v}$ et la règle de décision est :

$$d'(\boldsymbol{v}') = d(A_{d(v)}\,\boldsymbol{v})$$

Nous avons alors :

(i) Il y a reconstruction parfaite si et seulement si $\forall \boldsymbol{v},\ d(A_{d(v)}\,\boldsymbol{v}) = d(\boldsymbol{v})$.

(ii) Une condition nécessaire et suffisante pour la vérifier est :

$$\forall i \in \{0, 1, \ldots, N-1\}, \quad \text{si } d(\boldsymbol{v}) = i, \text{ alors } d(A_i\,\boldsymbol{v}) = i \tag{6.17}$$

Preuve. La preuve de (i) est directe. Prouvons (ii).

Supposons que l'implication (6.17) soit vraie. Comme précédemment, faisons l'hypothèse que $d(A_{d(v)}\,\boldsymbol{v}) = i$ et $d(\boldsymbol{v}) \neq i$. Dans ce cas, comme $\{\mathcal{D}_i\}$, $i = 0, \ldots, N-1$ est une partition de V, il existe $j \neq i \in \{0, \ldots, N-1\}$ tel que $d(\boldsymbol{v}) = j$. En accord avec (6.17), cela implique $d(A_j\boldsymbol{v}) = j$. Mais comme nous avons $d(A_j\boldsymbol{v}) = i$, cela conduit à une contradiction car $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset$. On montre alors que la condition de reconstruction parfaite est satisfaite.

Réciproquement, si la condition de reconstruction parfaite est vérifiée alors la preuve (6.17) est établie de façon directe. ■

Une condition plus faible pour assurer la reconstruction parfaite (et donc *suffisante* pour satisfaire les conditions nécessaires et suffisantes de la proposition précédente) consiste-

rait à vérifier simultanément toutes les implications suivantes :

$$\left\{\begin{array}{lcl}
p_0(\boldsymbol{v}) < p_{N-1}(\boldsymbol{v}) & \Rightarrow & p_0(A_0\boldsymbol{v}) < p_{N-1}(A_0\boldsymbol{v}) \\
p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) & \Rightarrow & p_0(A_0\boldsymbol{v}) \leq p_1(A_0\boldsymbol{v}) \\
 & \vdots & \\
p_0(\boldsymbol{v}) \leq p_{N-2}(\boldsymbol{v}) & \Rightarrow & p_0(A_0\boldsymbol{v}) \leq p_{N-2}(A_0\boldsymbol{v}) \\
p_1(\boldsymbol{v}) < p_{N-1}(\boldsymbol{v}) & \Rightarrow & p_1(A_1\boldsymbol{v}) < p_{N-1}(A_1\boldsymbol{v}) \\
p_1(\boldsymbol{v}) < p_0(\boldsymbol{v}) & \Rightarrow & p_1(A_1\boldsymbol{v}) < p_0(A_1\boldsymbol{v}) \\
 & \vdots & \\
p_1(\boldsymbol{v}) \leq p_{N-2}(\boldsymbol{v}) & \Rightarrow & p_1(A_1\boldsymbol{v}) \leq p_{N-2}(A_1\boldsymbol{v}) \\
 & \vdots & \\
p_{N-1}(\boldsymbol{v}) \leq p_0(\boldsymbol{v}) & \Rightarrow & p_{N-1}(A_{N-1}\boldsymbol{v}) \leq p_0(A_{N-1}\boldsymbol{v}) \\
 & \vdots & \\
p_{N-1}(\boldsymbol{v}) \leq p_{N-2}(\boldsymbol{v}) & \Rightarrow & p_{N-1}(A_{N-1}\boldsymbol{v}) \leq p_{N-2}(A_{N-1}\boldsymbol{v})\,.
\end{array}\right.$$

Ces conditions peuvent être combinées par paires, conduisant ainsi à :

$$\left\{\begin{array}{lcl}
p_{N-1}(\boldsymbol{v}) \leq p_0(\boldsymbol{v}) & \Rightarrow & p_{N-1}(A_{N-1}\boldsymbol{v}) \leq p_0(A_{N-1}\boldsymbol{v}) \\
p_{N-1}(\boldsymbol{v}) > p_0(\boldsymbol{v}) & \Rightarrow & p_{N-1}(A_0\boldsymbol{v}) > p_0(A_0\boldsymbol{v})
\end{array}\right.$$

$$\left\{\begin{array}{lcl}
p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) & \Rightarrow & p_0(A_0\boldsymbol{v}) \leq p_1(A_0\boldsymbol{v}) \\
p_0(\boldsymbol{v}) > p_1(\boldsymbol{v}) & \Rightarrow & p_0(A_1\boldsymbol{v}) > p_1(A_1\boldsymbol{v})
\end{array}\right.$$

etc... De cette manière, une condition suffisante de reconstruction parfaite s'exprime comme un ensemble de $N(N-1)/2$ conditions, chacune d'entre elles mettant en jeu seulement deux seminormes. Ces équations sont en fait semblables aux conditions (6.5) obtenues lors de la comparaison de deux seminormes. Les résultats obtenus dans la section 6.2 peuvent alors être utilisés ici pour traduire les équations précédentes en conditions plus pratiques à manipuler.

Exemple 3 Un exemple de conditions nécessaires et suffisantes, assurant la reconstruction parfaite dans le cas de la Proposition 8 pour $N = 3$, est donné par :

$$\left\{\begin{array}{lcl}
p_0(\boldsymbol{v}) & < & p_2(\boldsymbol{v}) \\
p_0(\boldsymbol{v}) & \leq & p_1(\boldsymbol{v})
\end{array}\right. \Longrightarrow \left\{\begin{array}{lcl}
p_0(A_0\boldsymbol{v}) & < & p_2(A_0\boldsymbol{v}) \\
p_0(A_0\boldsymbol{v}) & \leq & p_1(A_0\boldsymbol{v})
\end{array}\right.$$

$$\left\{\begin{array}{lcl}
p_1(\boldsymbol{v}) & < & p_2(\boldsymbol{v}) \\
p_1(\boldsymbol{v}) & < & p_0(\boldsymbol{v})
\end{array}\right. \Longrightarrow \left\{\begin{array}{lcl}
p_1(A_1\boldsymbol{v}) & < & p_2(A_1\boldsymbol{v}) \\
p_1(A_1\boldsymbol{v}) & < & p_0(A_1\boldsymbol{v})
\end{array}\right.$$

$$\left\{\begin{array}{lcl}
p_2(\boldsymbol{v}) & \leq & p_0(\boldsymbol{v}) \\
p_2(\boldsymbol{v}) & \leq & p_1(\boldsymbol{v})
\end{array}\right. \Longrightarrow \left\{\begin{array}{lcl}
p_2(A_2\boldsymbol{v}) & \leq & p_0(A_2\boldsymbol{v}) \\
p_2(A_2\boldsymbol{v}) & \leq & p_1(A_2\boldsymbol{v})
\end{array}\right.$$

Des conditions suffisantes, vérifiant les conditions précédentes sont alors données par :

$$\left\{\begin{array}{lcl}
p_2(\boldsymbol{v}) \leq p_0(\boldsymbol{v}) & \Longrightarrow & p_2(A_2\boldsymbol{v}) \leq p_0(A_2\boldsymbol{v}) \\
p_2(\boldsymbol{v}) > p_0(\boldsymbol{v}) & \Longrightarrow & p_2(A_0\boldsymbol{v}) > p_0(A_0\boldsymbol{v})
\end{array}\right.$$

$$\left\{\begin{array}{lcl}
p_2(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) & \Longrightarrow & p_2(A_2\boldsymbol{v}) \leq p_1(A_2\boldsymbol{v}) \\
p_2(\boldsymbol{v}) > p_1(\boldsymbol{v}) & \Longrightarrow & p_2(A_1\boldsymbol{v}) > p_1(A_1\boldsymbol{v})
\end{array}\right.$$

$$\begin{cases} p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) & \Longrightarrow \quad p_0(A_0\boldsymbol{v}) \leq p_1(A_0\boldsymbol{v}) \\ p_0(\boldsymbol{v}) > p_1(\boldsymbol{v}) & \Longrightarrow \quad p_0(A_1\boldsymbol{v}) > p_1(A_1\boldsymbol{v}) \end{cases}$$

Considérons les seminormes $p_0(\boldsymbol{v}) = |\boldsymbol{a}_0^T\,\boldsymbol{v}|$, $p_1(\boldsymbol{v}) = |\boldsymbol{a}_1^T\,\boldsymbol{v}|$ et $p_2(\boldsymbol{v}) = |\boldsymbol{a}_2^T\,\boldsymbol{v}|$. Supposons de plus que $\boldsymbol{a}_0$, $\boldsymbol{a}_1$ et $\boldsymbol{a}_2$ sont linéairement indépendants et vérifient $\boldsymbol{u}^T\boldsymbol{a}_0 = \boldsymbol{u}^T\boldsymbol{a}_1 = \boldsymbol{u}^T\boldsymbol{a}_2 = \xi \neq 0$. La reconstruction parfaite est alors assurée par les conditions suffisantes suivantes :

$$\boldsymbol{b}_i = \frac{\beta_i \boldsymbol{a}_i}{\xi}, \quad \text{où} \quad 0 \leq \beta_i < 1\,, \quad i = 0, 1, 2$$

Par exemple, si $\boldsymbol{a}_0 = (1, 0, 1, 0)^T$, $\boldsymbol{a}_1 = (0, 1, 0, 1)^T$ et $\boldsymbol{a}_2 = \frac{1}{2}(1, 1, 1, 1)^T$ alors :

$$\boldsymbol{b}_0 = \frac{\beta_0}{2}(1, 0, 1, 0)^T, \quad \boldsymbol{b}_1 = \frac{\beta_1}{2}(0, 1, 0, 1)^T, \quad \boldsymbol{b}_2 = \frac{\beta_2}{4}(1, 1, 1, 1)^T$$

où $0 \leq \beta_i < 1,\ i = 0, 1, 2$.

Dans le cas de signaux monodimensionnels (voir l'indexation dans la Fig. 6.3), cela correspond à comparer l'information de gradient du côté gauche de l'échantillon $x(n)$ à mettre à jour, avec l'information de gradient située du côté droit et avec l'information calculée des deux côtés.

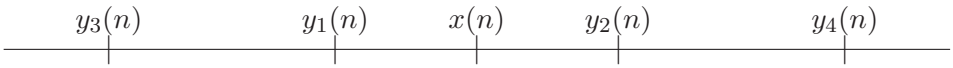


FIG. 6.3 – Exemple d'indexation des échantillons d'un signal monodimensionnel.

Dans le cas d'images (voir la Fig. 6.2), ce critère revient à comparer les gradients dans les directions horizontales et verticales avec un gradient isotrope, calculé à partir des quatre échantillons voisins.

6.4 Combinaison de deux seminormes et du critère TC

Nous pouvons aussi combiner la comparaison de deux seminormes avec le critère de seuil TC pour chacune d'entre elle. Cela revient à considérer les 4 régions de décision suivantes :

$$p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) \text{ et } p_0(\boldsymbol{v}) \leq T_0 \quad \Leftrightarrow \quad d = 0 \tag{6.18}$$
$$p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) \text{ et } p_0(\boldsymbol{v}) > T_0 \quad \Leftrightarrow \quad d = 1 \tag{6.19}$$
$$p_0(\boldsymbol{v}) > p_1(\boldsymbol{v}) \text{ et } p_1(\boldsymbol{v}) \leq T_1 \quad \Leftrightarrow \quad d = 2 \tag{6.20}$$
$$p_0(\boldsymbol{v}) > p_1(\boldsymbol{v}) \text{ et } p_1(\boldsymbol{v}) > T_1 \quad \Leftrightarrow \quad d = 3 \tag{6.21}$$

où T_0 et T_1 sont deux valeurs de seuil positives. Une règle similaire avec les seuils T_0' et T_1' est valable durant la synthèse.

Nous pouvons déduire de la Proposition 8, les conditions nécessaires et suffisantes suivantes, permettant d'assurer la reconstruction parfaite :

$$\begin{cases} p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) \\ p_0(\boldsymbol{v}) \leq T_0 \end{cases} \quad \Rightarrow \quad \begin{cases} p_0(A_0\boldsymbol{v}) \leq p_1(A_0\boldsymbol{v}) \\ p_0(A_0\boldsymbol{v}) \leq T_0' \end{cases}$$

$$\left\{\begin{array}{lcl} p_0(\boldsymbol{v}) & \leq & p_1(\boldsymbol{v}) \\ p_0(\boldsymbol{v}) & > & T_0 \end{array}\right. \Rightarrow \left\{\begin{array}{lcl} p_0(A_1\boldsymbol{v}) & \leq & p_1(A_1\boldsymbol{v}) \\ p_0(A_1\boldsymbol{v}) & > & T_0' \end{array}\right.$$

$$\left\{\begin{array}{lcl} p_0(\boldsymbol{v}) & > & p_1(\boldsymbol{v}) \\ p_1(\boldsymbol{v}) & \leq & T_1 \end{array}\right. \Rightarrow \left\{\begin{array}{lcl} p_0(A_2\boldsymbol{v}) & > & p_1(A_2\boldsymbol{v}) \\ p_1(A_2\boldsymbol{v}) & \leq & T_1' \end{array}\right.$$

$$\left\{\begin{array}{lcl} p_0(\boldsymbol{v}) & > & p_1(\boldsymbol{v}) \\ p_1(\boldsymbol{v}) & > & T_1 \end{array}\right. \Rightarrow \left\{\begin{array}{lcl} p_0(A_3\boldsymbol{v}) & > & p_1(A_3\boldsymbol{v}) \\ p_0(A_3\boldsymbol{v}) & > & T_1' \end{array}\right.$$

Encore une fois, une condition suffisante permettant de satisfaire les relations précédentes peut être déduite de ces dernières. Elle nécessite que toutes les implications suivantes soient vérifiées individuellement, c'est à dire :

$$\left\{\begin{array}{lcl} p_0(\boldsymbol{v}) \leq T_0 & \Rightarrow & p_0(A_0\boldsymbol{v}) \leq T_0' \\ p_0(\boldsymbol{v}) > T_0 & \Rightarrow & p_0(A_1\boldsymbol{v}) > T_0' \end{array}\right. \tag{6.22}$$

$$\left\{\begin{array}{lcl} p_0(\boldsymbol{v}) \leq T_1 & \Rightarrow & p_0(A_2\boldsymbol{v}) \leq T_1' \\ p_0(\boldsymbol{v}) > T_1 & \Rightarrow & p_0(A_3\boldsymbol{v}) > T_1' \end{array}\right. \tag{6.23}$$

$$\left\{\begin{array}{lcl} p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) & \Rightarrow & p_0(A_0\boldsymbol{v}) \leq p_1(A_0\boldsymbol{v}) \\ p_0(\boldsymbol{v}) > p_1(\boldsymbol{v}) & \Rightarrow & p_0(A_2\boldsymbol{v}) > p_1(A_2\boldsymbol{v}) \end{array}\right. \tag{6.24}$$

$$\left\{\begin{array}{lcl} p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) & \Rightarrow & p_0(A_1\boldsymbol{v}) \leq p_1(A_1\boldsymbol{v}) \\ p_0(\boldsymbol{v}) > p_1(\boldsymbol{v}) & \Rightarrow & p_0(A_3\boldsymbol{v}) > p_1(A_3\boldsymbol{v}) \end{array}\right. \tag{6.25}$$

On remarque ainsi que le critère de seuil TC peut être combiné de manière simple avec plusieurs seminormes. Lors de nos expérimentations en section 6.5, nous donnons trois exemples d'utilisation de ce critère de décision combiné.

Proposition 9 Considérons la règle de décision donnée par (6.18)-(6.21), avec $p_0(\boldsymbol{v}) = |\boldsymbol{a}_0^T\boldsymbol{v}|$ et $p_1(\boldsymbol{v}) = |\boldsymbol{a}_1^T\boldsymbol{v}|$, où $\boldsymbol{a}_0$ et $\boldsymbol{a}_1$ ne sont pas colinéaires. Une condition suffisante permettant d'assurer la reconstruction parfaite est donnée par les relations :

$$\boldsymbol{u}^T\boldsymbol{a}_0 \neq 0, \boldsymbol{u}^T\boldsymbol{a}_1 \neq 0,\ T_0' = |\alpha_0|\,T_0,\ T_1' = |\alpha_2|\,T_1 \tag{6.26}$$

$$\boldsymbol{b}_0 = \frac{\beta_0}{\boldsymbol{u}^T\boldsymbol{a}_1}\boldsymbol{a}_0, \quad \boldsymbol{b}_1 = \frac{\beta_1}{\boldsymbol{u}^T\boldsymbol{a}_1}\boldsymbol{a}_0 \tag{6.27}$$

$$\boldsymbol{b}_2 = \frac{\beta_2}{\boldsymbol{u}^T\boldsymbol{a}_0}\boldsymbol{a}_1, \quad \boldsymbol{b}_3 = \frac{\beta_3}{\boldsymbol{u}^T\boldsymbol{a}_0}\boldsymbol{a}_1 \tag{6.28}$$

où $\forall i \in \{0,1,2,3\}$, $0 < |\alpha_i| \leq 1 - |\beta_i|$ et $|\alpha_0| \leq |\alpha_1|$, $|\alpha_2| \leq |\alpha_3|$.

Preuve. Afin de satisfaire le critère de seuil TC (6.22)-(6.23), il est nécessaire et suffisant [59] d'avoir :

$$\boldsymbol{b}_0 = \gamma_0\boldsymbol{a}_0 \text{ et } \boldsymbol{b}_1 = \gamma_1\boldsymbol{a}_0, \quad \text{avec } \gamma_0, \gamma_1 \text{ tel que } |\alpha_0| \leq |\alpha_1|$$

$$\boldsymbol{b}_2 = \gamma_2\boldsymbol{a}_1 \text{ et } \boldsymbol{b}_3 = \gamma_3\boldsymbol{a}_1, \quad \text{avec } \gamma_2, \gamma_3 \text{ tel que } |\alpha_2| \leq |\alpha_3|$$

et de choisir $T_0' \in \left[|\alpha_0|\,T_0,\, |\alpha_1|\,T_0\right]$ et $T_1' \in \left[|\alpha_2|\,T_1,\, |\alpha_3|\,T_1\right]$. Les relations précédentes de colinéarité sont consistantes avec les équations (6.27)-(6.28). De plus, les relations (6.27)-(6.28) et la Proposition 5 garantissent alors que les équations (6.24)-(6.25) sont satisfaites. ■

Exemple 4 Considérons le cas où $\boldsymbol{a}_0 = (1,0,1,0)^T$, $\boldsymbol{a}_1 = (0,1,0,1)^T$ et $T_0 = T_1 = T$. Conformément à la proposition précédente, la propriété de reconstruction parfaite est garantie si nous choisissons :

$$\boldsymbol{b}_0 = \frac{\beta_0}{2}(1,0,1,0)^T, \quad \boldsymbol{b}_1 = \frac{\beta_1}{2}(1,0,1,0)^T$$
$$\boldsymbol{b}_2 = \frac{\beta_2}{2}(0,1,0,1)^T, \quad \boldsymbol{b}_3 = \frac{\beta_3}{2}(0,1,0,1)^T$$

avec $0 \leq \beta_1 \leq \beta_0 < 1$, $0 \leq \beta_3 \leq \beta_2 < 1$ et en choisissant les seuils durant la synthèse $T_0' = (1-\beta_0)T$ et $T_1' = (1-\beta_2)T$.

En particulier, nous pouvons prendre $\beta_1 = \beta_3 = 0$, ce qui correspond au filtre de mise à jour identité (pas de filtrage) quand une discontinuité est détectée, lorsque par exemple $\min\{p_0(\boldsymbol{v}), p_1(\boldsymbol{v})\} > T$. Dans le cas contraire, le filtrage de mise à jour est effectué en utilisant $\boldsymbol{b}_0$ ou $\boldsymbol{b}_2$, en fonction du plus faible gradient.

6.5 Résultats expérimentaux

Afin d'évaluer l'efficacité de décorrélation des transformées adaptative présentées dans les sections précédentes, nous avons procédé à plusieurs simulations de compression d'image sans perte, en utilisant différents critères de décision.

6.5.1 Protocole expérimental

Nos expérimentations ont été conduites sur des images, en considérant une structure d'analyse polyphase utilisant une étape de mise à jour adaptative et suivie de trois étapes de prédiction, comme illustré par la Fig. 6.4. Avant d'être décomposée, l'image d'entrée x_0 est préalablement scindée en composantes polyphases x, y_h, y_v et y_d selon :

$$\begin{cases} x(m,n) & = x_0(2m,2n) \\ y_h(m,n) & = x_0(2m,2n+1) \\ y_v(m,n) & = x_0(2m+1,2n) \\ y_d(m,n) & = x_0(2m+1,2n+1) \end{cases}$$

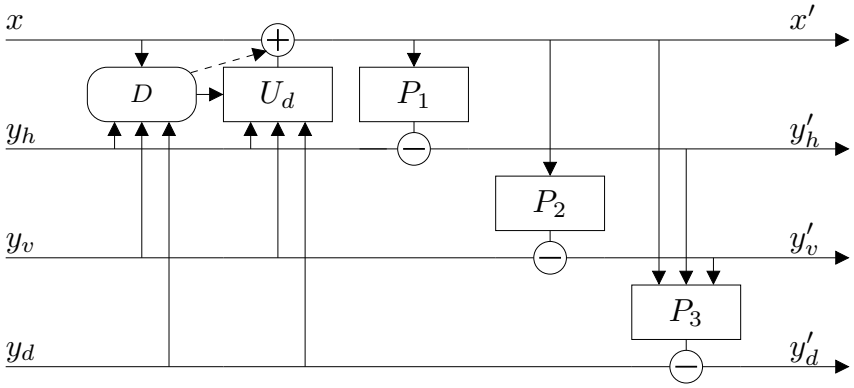


FIG. 6.4 – Structure lifting d'une décomposition 2D réalisant une étape de mise à jour adaptative U_d suivie de trois étapes de prédiction P_1, P_2 et P_3.

Dans cette décomposition, le signal x' est nommé sous-bande *d'approximation* et les signaux y'_h, y'_v et y'_d sont les sous-bande de détail, respectivement *horizontale*, *verticale* et *diagonale*. Les échantillons sont pris sur les bandes y_h, y_b et y_d en considérant l'indexation illustrée en Fig. 6.5.

$y_6(\boldsymbol{n})$	$y_2(\boldsymbol{n})$	$y_5(\boldsymbol{n})$
$y_3(\boldsymbol{n})$	$x(\boldsymbol{n})$	$y_1(\boldsymbol{n})$
$y_7(\boldsymbol{n})$	$y_4(\boldsymbol{n})$	$y_8(\boldsymbol{n})$

FIG. 6.5 – Indexation des échantillons dans une fenêtre 3×3 centrée sur $x(\boldsymbol{n})$.

Les étapes de prédictions sont fixes et sont régies par les équations :

$$\begin{aligned} y'_h &= y_h - x' \\ y'_v &= y_v - x' \\ y'_d &= y_d - x' - y'_h - y'_v \end{aligned}$$

De plus, afin de maximiser l'efficacité de codage, il est important de normaliser les coefficients d'approximation et de détail à chaque niveau de décomposition. Les coefficients d'approximation sont multipliés par une constante ζ tandis que ceux de détail sont multipliés par $1/\zeta$. Si on impose au signal d'approximation x' de préserver l'énergie du signal original x alors les coefficients du filtre de mise à jour doivent être normalisés de façon à ce que leur norme ℓ_2 soit égale à 1. Ceci implique que :

$$\zeta^2\big(\alpha^2 + \sum_{j=1}^{J} \mu_j^2\big) = 1$$

conduisant à $\zeta = 1/\sqrt{(\alpha^2 + \sum_{j=1}^{J} \mu_j^2)}$. Enfin, les valeurs des seuils T utilisées dans certaines expérimentations ont été choisies de façon heuristique et dépendent de l'expérimentation, des images et des seminormes choisies.

Afin d'illustrer les résultats théoriques obtenus dans les sections précédentes, nous avons procédé à plusieurs expérimentations en utilisant plusieurs seminormes et différents critères de comparaison entre elles, en combinant ou non un critère de seuil. Les expérimentations sont détaillées dans la sous-section suivante.

6.5.2 Détail des expérimentations

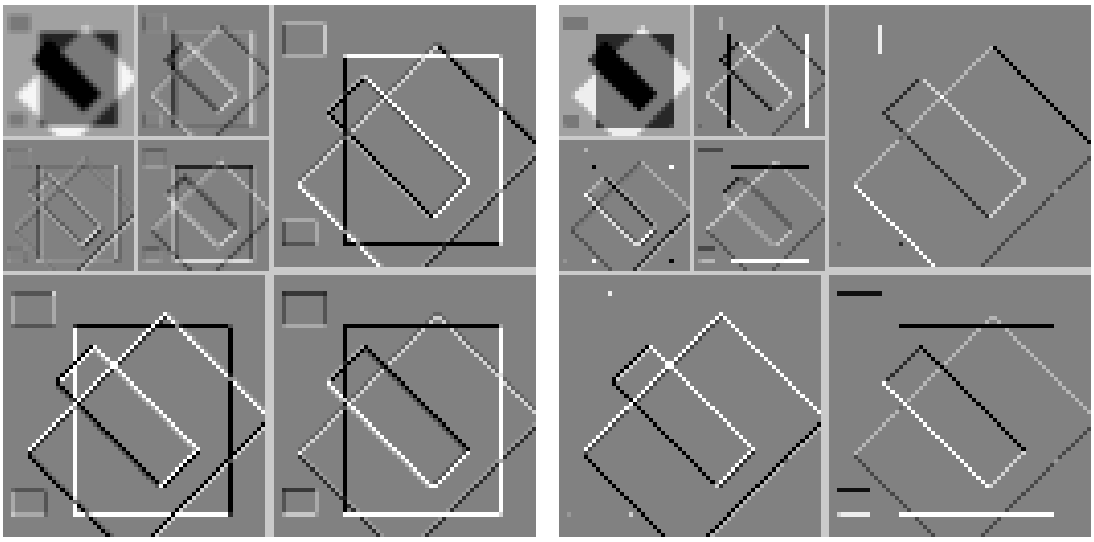
Isotrope non-adaptatif - Fig. 6.6

Cette expérimentation consiste à utiliser le même filtre de mise à jour isotrope $\boldsymbol{b} = (1,1,1,1)^T/8$ en chaque point. Ce n'est donc pas une transformation adaptative, ser-

vant cependant de référence aux autres expérimentations. La partie gauche de la Fig. 6.6 illustre la décomposition par cette transformée d'une image sur deux niveaux. On observe le gommage important des contours de l'image dans les sous-bandes de détail.

Laplacien adaptatif

La transformée Laplacienne adaptative est une décomposition mettant en œuvre le critère de seuil TC, rappelé en section 6.1. On utilise la règle de décision $d = [p(\boldsymbol{v}) > T]$ avec $p(\boldsymbol{v}) = |\boldsymbol{a}^T \boldsymbol{v}|$ et $\boldsymbol{a} = (1, 1, 1, 1, 0, 0, 0, 0)^T$. Les filtres de mise à jour correspondant pour $d = \{0, 1\}$ sont donnés par $\boldsymbol{b}_d = \gamma_d \boldsymbol{a}$ avec $\gamma_0 = 1/8$ et $\gamma_1 = 0$. Cette décomposition adaptative revient donc à filtrer uniformément un pixel durant la mise à jour si la moyenne isotrope de ses voisins est inférieure au seuil T. Sinon, le pixel n'est pas filtré.



Isotrope non-adaptatif HV

FIG. 6.6 – Décomposition multirésolution par filtrage isotrope non-adaptatif (gauche) et par sélection adaptative HV entre un filtre horizontal et vertical (droite).

HV - Sélection entre un filtre horizontal et vertical - Fig. 6.6

Cette expérimentation adaptative met en pratique le critère de décision basé sur la comparaison de deux seminormes, en utilisant le contre-exemple de la Proposition 6 de la section 6.2. Les coefficients des filtres de mise à jour sont choisis comme décrit par l'équation (6.13), c'est à dire $\boldsymbol{b}_d = (\mu_d, \eta_d, \mu_d, \eta_d)^T$ pour $d = \{0, 1\}$ avec $\mu_0 = \eta_1 = 0$ et $\mu_1 = \eta_0 = 1/4$. Les conditions de la Proposition 6 sont ainsi satisfaites. Cette expérimentation est nommée HV et revient à effectuer durant la mise à jour et pour chaque pixel, une sélection entre un filtre horizontal et un filtre vertical.

La partie droite de la Fig. 6.6 illustre le schéma adaptatif HV appliqué à la décomposition d'une image synthétique. On remarque que l'approximation possède des contours plus nets que dans le cas isotrope. De plus, les sous-bandes de détail paraissent plus "creuses" et seront plus simple à coder pour un codeur emboîté.

Nous avons détaillé les sous-bandes issues de la décomposition HV d'une image synthétique en Fig. 6.7, située en haut à gauche. La carte de décision correspondante est

présentée à sa droite, où les pixels blancs décrivent les régions dont le gradient horizontal est supérieur au gradient vertical. Le filtre de mise à jour vertical sera donc utilisé sur ces régions. Les images d'approximation et de détail horizontal sont présentées sur la deuxième ligne. Enfin, l'image de détail diagonale HV est illustrée en dernière ligne.

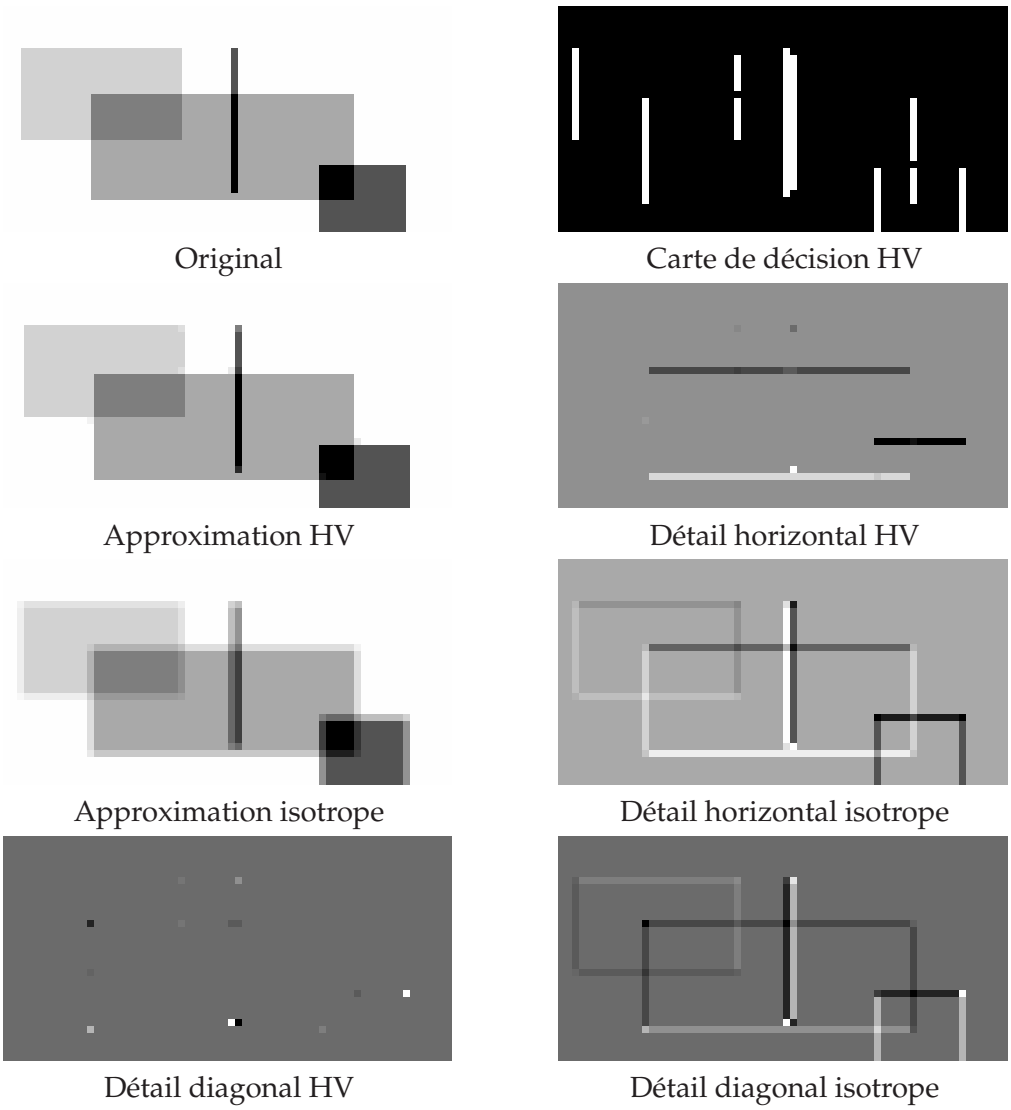


FIG. 6.7 – Sous-bandes mises en jeu dans une décomposition multirésolution par filtrage adaptatif HV et comparées à celles issues d'une décomposition isotrope non-adaptative.

Nous comparons ce schéma de décomposition à la transformée isotrope non-adaptative, où un filtrage isotropique est effectuée dans les directions verticales et horizontales (ceci revient à prendre $\mu = \eta = 1/8$). Les images d'approximation et de détail horizontal sont présentées dans la troisième ligne de la Fig. 6.7, tandis que l'image de détail diagonal occupe la position droite de la dernière ligne. Nous observons clairement que les images d'approximation obtenues dans le cas adaptatif HV préservent les contours présents dans l'image originale, contrairement au cas isotrope non-adaptatif. Par conséquent, les images de détail obtenues dans le cas adaptatif 'capturent' de façon plus compacte les contours que dans le cas non-adaptatif.

HVDD - Sélection entre un filtre horizontal, vertical et 2 diagonaux - Fig. 6.8

Dans cette expérimentation nommée HVDD, le critère de décision est basé sur la comparaison de 4 seminormes, afin d'illustrer les résultats théoriques de la section 6.3. Nous considérons les seminormes $p_i(\boldsymbol{v}) = |\boldsymbol{a}_i^T\, \boldsymbol{v}|$ définies pour $i = 0, \ldots, 3$ par :

$$\begin{aligned}
\boldsymbol{a}_0 &= (1,0,1,0,0,0,0,0)^T \\
\boldsymbol{a}_1 &= (0,1,0,1,0,0,0,0)^T \\
\boldsymbol{a}_2 &= (0,0,0,0,1,0,1,0)^T \\
\boldsymbol{a}_3 &= (0,0,0,0,0,1,0,1)^T
\end{aligned}$$

correspondant respectivement aux directions horizontales, verticales et aux directions diagonales. Pour assurer la reconstruction parfaite, nous utilisons les filtres de mise à jour $\boldsymbol{b}_i = \frac{1}{4}\boldsymbol{a}_i$ (et donc $\beta_i = 1/2$) pour $i = 0, \ldots, 3$.

Conformément à la section 6.3, ce critère de décision revient à filtrer l'image selon sa direction de plus faible gradient. La partie gauche de la Fig.6.8 montre une image originale et les partitions de la carte de décisions obtenues lors de cette expérimentation sur un niveau de décomposition. Les pixels blancs dénotent les régions où $d = 1$ (détails verticaux, en haut à droite), $d = 2$ (détails diagonaux, en bas à gauche) et $d = 3$ (détails diagonaux, en bas à droite). La carte de décision associée à $d = 0$ (détails horizontaux, non représentée) se déduit par complémentarité de l'union des autres cartes.

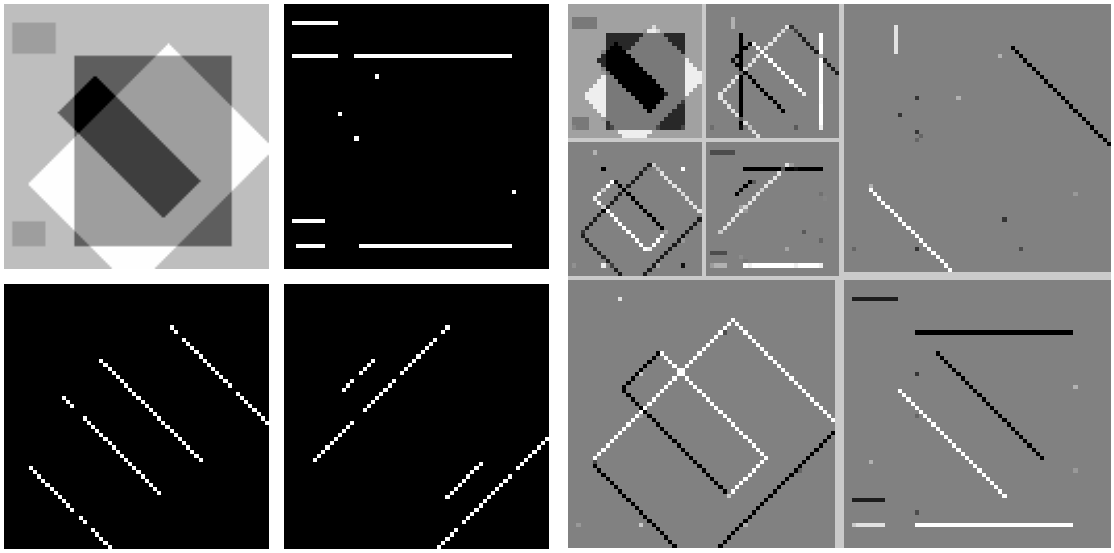


FIG. 6.8 – Image originale et cartes de décision associée à l'expérimentation HVDD (gauche). Décomposition HVDD sur un niveau spatial (droite).

La partie droite de la Fig. 6.8 illustre la décomposition adaptative HVVD d'une l'image sur deux niveaux. Comme dans le cas d'une décomposition dyadique classique, on observe les sous-bandes d'approximation et les sous-bandes de détail horizontal, vertical et diagonal des deux niveaux de décomposition.

En comparant cette décomposition avec celle obtenue dans le cas non-adaptatif isotrope de la Fig. 6.6, on observe que les contours sont mieux préservés dans l'image d'approximation. La carte de décision associée à ce schéma est ainsi capable de distinguer

parmi les quatre directions, laquelle est la plus adaptée pour appliquer le filtrage passe-bas de mise à jour. Ceci permet d'éviter un gommage des contours dans les images d'approximation et conduit à des images de détail contenant moins de coefficients, laissant ainsi entrevoir une efficacité de codage accrue.

HVI - Sélection entre un filtre horizontal, vertical et isotrope - Fig. 6.10

Dans cette expérimentation nommée HVI, nous nous proposons d'utiliser des filtres de mise à jour orientés dans les directions horizontale, verticale et isotrope. Ceci revient ainsi à comparer les 3 seminormes $p_i(\boldsymbol{v}) = |\boldsymbol{a}_i^T\, \boldsymbol{v}|$, définies pour $i = 0, \ldots, 2$ par :

$$\begin{aligned}\boldsymbol{a}_0 &= (1,0,1,0,0,0,0,0)^T\\ \boldsymbol{a}_1 &= (0,1,0,1,0,0,0,0)^T\\ \boldsymbol{a}_2 &= (1,1,1,1,0,0,0,0)^T\end{aligned}$$

correspondant respectivement aux directions horizontale, verticale et isotrope, c'est à dire dans le sens horizontal *et* vertical. Comme dans le cas précédent, cette expérimentation revient à filtrer l'image dans la direction possédant le plus faible gradient.

Afin d'assurer la reconstruction parfaite, nous utilisons encore les coefficients $\beta_i = 1/2$ pour $i = 0, \ldots, 2$, conduisant ainsi à utiliser les filtres $\boldsymbol{b}_i = \frac{1}{4}\boldsymbol{a}_i$ pour $i = 0, 1$ et $\boldsymbol{b}_2 = \frac{1}{8}\boldsymbol{a}_2$.

Nous illustrons en Fig. 6.9 les partitions de la carte de décision pour $d = 0$ (à gauche) et $d = 1$ (à droite). La décision $d = 3$ (non-représentée) se déduit par complémentarité et désigne les régions qui subiront un filtrage isotrope. La partie droite de la Fig. 6.10 montre la décomposition multirésolution sur deux niveaux obtenue avec ce critère de décision. Comparée avec l'expérimentation précédente, on observe une approximation plus floutée et des images de détail contenant plus de coefficients, particulièrement dans les directions diagonales.

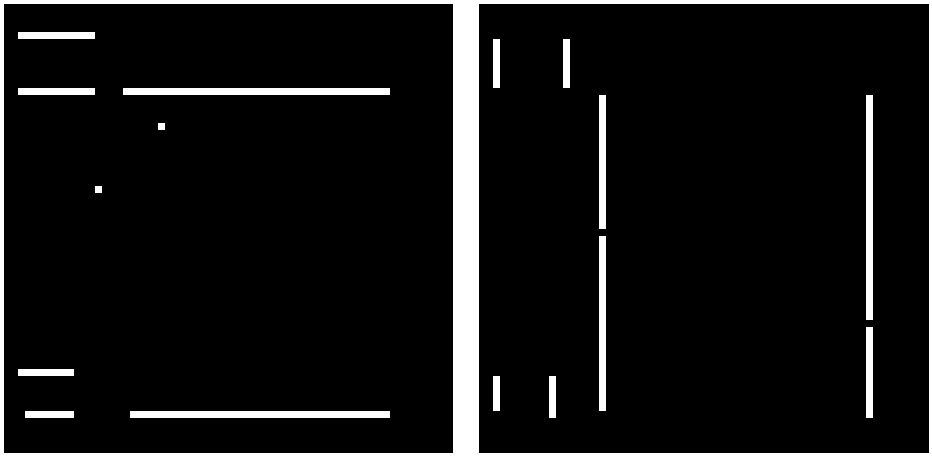


FIG. 6.9 – Partitions de la carte de décision associées à l'expérimentation HVI.

HV+TC - Sélection entre un filtre horizontal et vertical + TC - Fig. 6.10

Cette expérimentation correspond à l'Example 4, où nous utilisons un critère de décision combinant la comparaison de deux seminormes avec le critère de seuil TC. Afin

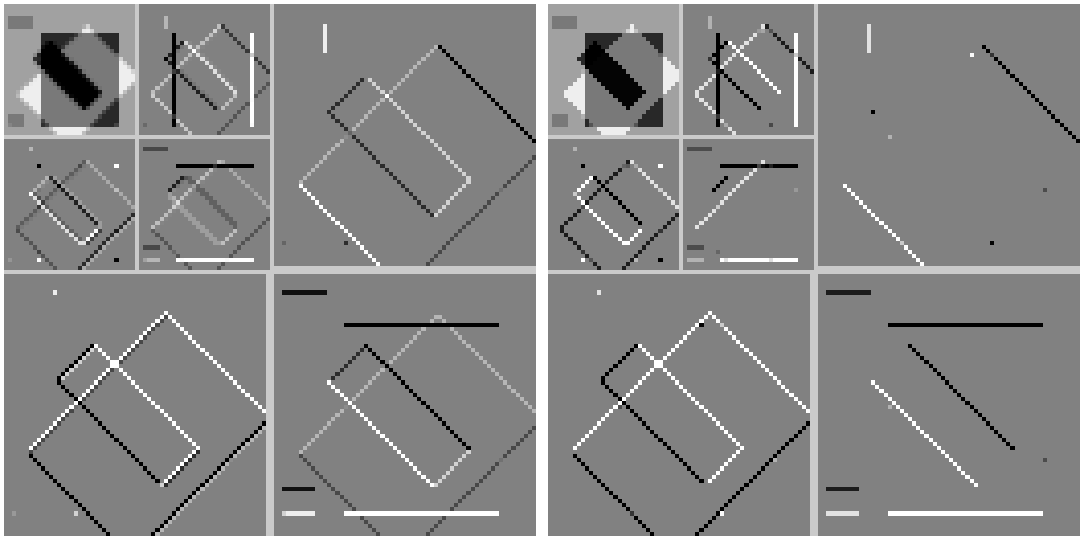


FIG. 6.10 – Décomposition par sélection adaptative entre un filtre horizontal, vertical et isotrope HVI (gauche) et par sélection adaptative HV + TC entre un filtre horizontal et vertical, combiné à un critère de seuil (droite).

d'assurer la reconstruction parfaite, nous utilisons les filtres de mise à jour $\beta_0 = \beta_2 = 1/2$ et $\beta_1 = \beta_3 = 0$. En prenant l'image originale utilisée dans les expérimentations précédentes, nous illustrons la décomposition obtenue dans la partie droite de la Fig. 6.10. On remarque que les images de détail contiennent légèrement moins de coefficients que lors des expérimentations précédentes HVDD (Fig 6.8, droite) et HVI (Fig 6.10, gauche).

HVHV+TC - Sélection entre quatre filtres + TC - Fig. 6.11

Nous souhaitons désormais utiliser un critère combinant la comparaison de deux seminormes avec le critère de seuil TC appliqué à deux autres seminormes, afin d'illustrer les résultats théoriques de la section 6.4. Le critère de décision s'écrit alors :

$$\begin{aligned} p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) \text{ et } p_2(\boldsymbol{v}) \leq T_0 \quad &\Leftrightarrow \quad d = 0 \\ p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) \text{ et } p_2(\boldsymbol{v}) > T_0 \quad &\Leftrightarrow \quad d = 1 \\ p_0(\boldsymbol{v}) > p_1(\boldsymbol{v}) \text{ et } p_3(\boldsymbol{v}) \leq T_1 \quad &\Leftrightarrow \quad d = 2 \\ p_0(\boldsymbol{v}) > p_1(\boldsymbol{v}) \text{ et } p_3(\boldsymbol{v}) > T_1 \quad &\Leftrightarrow \quad d = 3 \end{aligned}$$

où chaque seminorme $p_i(\boldsymbol{v}) = |\boldsymbol{a}_i^T|$ est définie par son vecteur $\boldsymbol{a}_i$ associé : $\boldsymbol{a}_0 = (1,0,1,0)^T$, $\boldsymbol{a}_1 = (0,1,0,1)^T$, $\boldsymbol{a}_2 = (1,1/2,1,1/2)^T$ et $\boldsymbol{a}_3 = (1/2,1,1/2,1)^T$. Nous choisissons des seuils identiques $T_0 = T_1$ et les filtres de mise à jour suivants :

$$\boldsymbol{b}_0 = \frac{1}{4}\boldsymbol{a}_2\,, \quad \boldsymbol{b}_2 = \frac{1}{4}\boldsymbol{a}_3$$

correspondant resp. à un filtrage à prédominance horizontale et verticale, et les filtres $\boldsymbol{b}_1 = \boldsymbol{b}_3 = \boldsymbol{0}$, impliquant qu'aucun filtrage ne sera effectué.

La décomposition sur deux niveaux obtenue par ce schéma est illustrée sur la partie gauche de la Fig. 6.11. Il apparaît qu'elle est visuellement indistinguable de celle obtenue dans le cas Laplacien adaptatif mais présente un peu moins de coefficients de détail que lors des expérimentations précédentes.

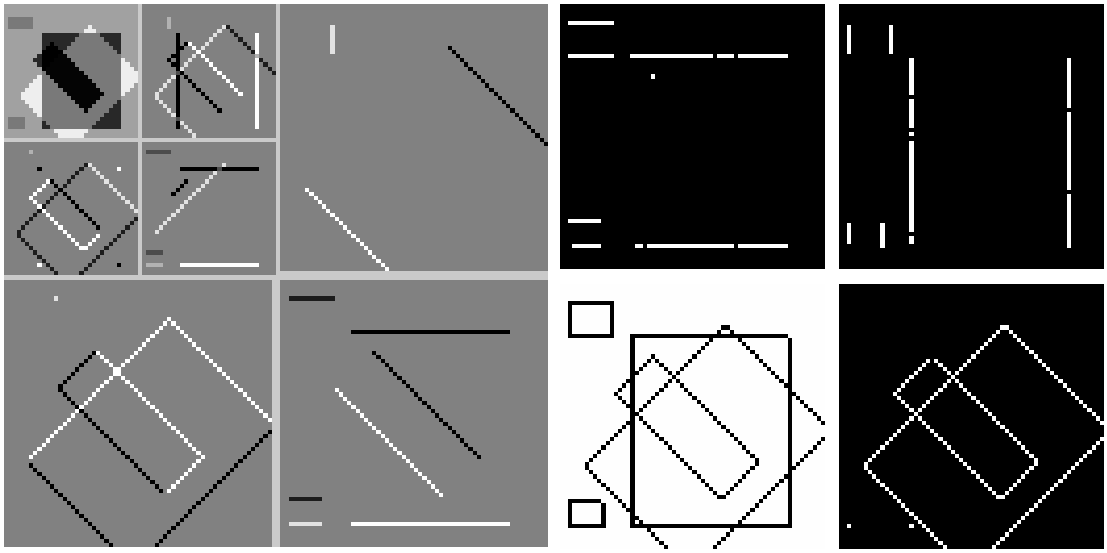


FIG. 6.11 – Décomposition par sélection adaptative HVHV + TC entre quatre filtres, combinée à un critère de seuil (droite). Cartes de décisions issues de la décomposition adaptative HVI + TC par sélection entre le filtre horizontal, vertical et isotrope, combinée à un critère de seuil (gauche).

HVI+TC - Sélection entre trois filtres + TC - Fig. 6.11

On combine ici le critère de sélection utilisé dans l'expérimentation HVI avec le critère de seuil TC, de telle manière qu'aucun filtrage ne soit appliqué si une des seminormes p_i est supérieure à un seuil donné.

La partie droite Fig. 6.11 montre les cartes de partition obtenues pour la décision $d = 0$ (en haut à droite), $d = 2$ (en haut à gauche), $d = 4$ (en bas à gauche) et l'union des partitions associées aux décisions $d = i$ pour $i = 1, 3, 5$ (en bas à droite). Cette dernière correspond aux régions où aucun filtrage ne sera effectué.

6.5.3 Efficacité de codage sans perte

Afin d'évaluer l'efficacité de codage réelle apportée par nos transformées adaptatives, nous considérons dans un premier temps un ensemble d'images synthétiques sur lesquelles nous réalisons les expérimentations décrites précédemment. Nous présentons alors des résultats concernant l'entropie des décompositions et procédons à des simulations réelles de codage sans perte. Dans un deuxième temps, nous appliquons le même protocole expérimental sur un autre ensemble test, constitué cette fois d'images naturelles.

Dans ces simulations, nous utilisons un schéma de prédiction fixe symétrique, comparable à celui utilisé dans le filtre biorthogonal $5/3$:

$$
\begin{aligned}
y'_h(\boldsymbol{n}) &= y_h(\boldsymbol{n}) - \big(x'(n+1,m) + x(\boldsymbol{n})\big)/2 \\
y'_v(\boldsymbol{n}) &= y_v(\boldsymbol{n}) - \big(x'(n,m+1) + x(\boldsymbol{n})\big)/2 \\
y'_d(\boldsymbol{n}) &= y_d(\boldsymbol{n}) - \big(x'(n,m+1) + x(\boldsymbol{n})\big)/2 - y'_h(\boldsymbol{n}) - y'_v(\boldsymbol{n})
\end{aligned}
$$

où $\boldsymbol{n}$ désigne le pixel d'indice spatial (n, m).

L'entropie du premier ordre des décompositions est calculée par la relation :

$$h = 2^{-2K} H(x^K) + \sum_{k=1}^{K} 2^{-2k} \sum_{j=1}^{3} H\left(y_{b_j}^k\right)$$

où K est le nombre de niveaux de décomposition et $H(x)$ dénote l'entropie du premier ordre de l'image x.

Images synthétiques

Nous considérons l'ensemble test d'images synthétiques illustré en Fig. 6.12.

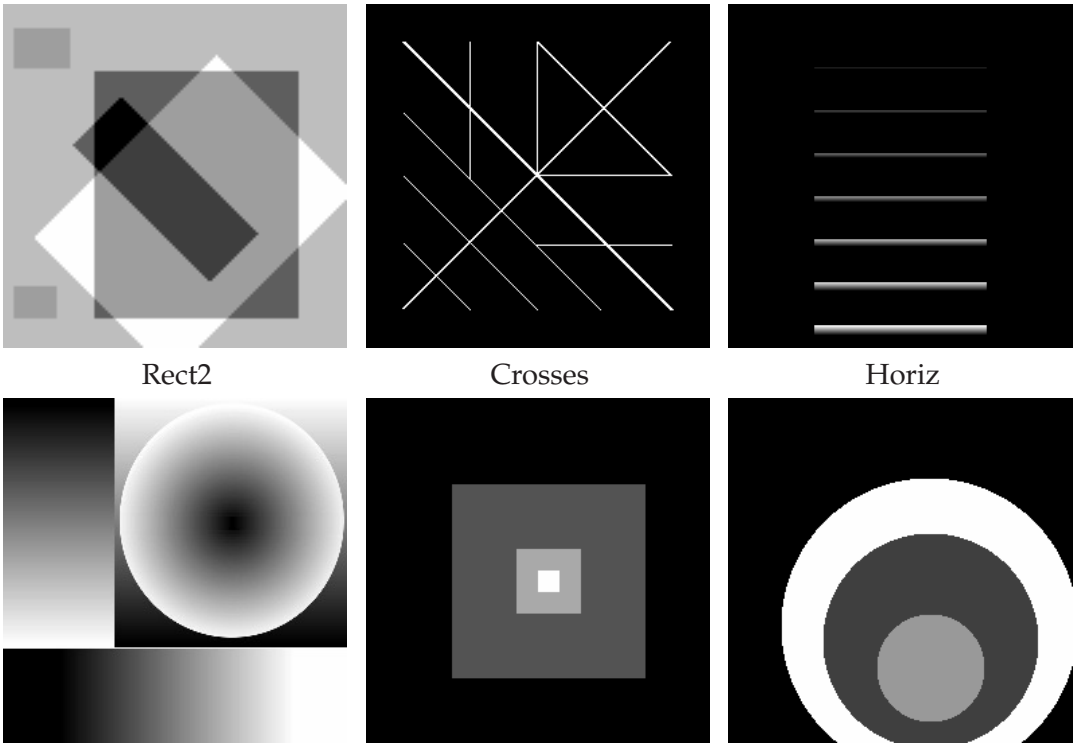


FIG. 6.12 – Ensemble test d'images synthétiques.

Le Tab.6.1 présente les valeurs d'entropie obtenues lors de la décomposition des images test sur deux niveaux de résolution, par les expérimentations décrites précédemment. La mention "Original" indique l'entropie originale des images non transformées. L'expérimentation "5/3" correspond à la transformée 5/3 non-adaptative, entière et séparable du codec JPEG-2000 en mode sans perte. Nous évaluons de plus dans le Tab. 6.2 le débit nécessaire à encoder ces images sans perte au moyen du codec d'image EZBC [60], décrit en section 1.2.4.

Une première observation pourrait être la constatation que les entropies minimales sont atteintes avec le filtre Laplacien adaptatif et les décompositions combinant un filtrage directionnel et ayant la possibilité de ne pas filtrer. Dans le cas de l'image 'Horiz', le filtre Laplacien n'est pas une bonne alternative et les expérimentations possédant un opérateur de filtrage horizontal donnent des entropies plus faibles. Concernant l'image

'Slope' où les bords ne sont pas bien définis, la décomposition HVHV+TC donne les meilleurs résultats, suivie par le filtre Laplacien adaptatif.

Les débits présentés dans le Tab. 6.2 sont assez consistants avec les entropie précédemment obtenues. Une remarque intéressante concerne les expérimentations HVHV+TC et HVI+TC qui conduisent à des débits plus faibles que le filtre Laplacien, bien que ces trois transformées mènent à des entropies très proches. La coïncidence des débits pour le filtre Laplacien et l'expérimentation HVI+TC est due au fait que cette dernière est une combinaison d'un filtre Laplacien et d'une sélection de filtres horizontal et vertical.

	Rect2	Crosses	Horiz	Slope	Squares	Circles
Original	2.016	0.188	0.449	7.517	1.077	1.781
Isotrope	1.257	0.933	0.877	2.210	0.441	0.859
Laplacien	**0.428**	**0.375**	0.566	2.035	**0.187**	**0.366**
HV	0.783	0.689	0.356	2.132	0.192	0.597
HVDD	0.498	0.399	0.356	2.184	0.193	0.514
HVI	0.785	0.689	0.357	2.111	0.193	0.596
HV+TC	**0.428**	**0.375**	0.351	2.113	**0.187**	**0.366**
HVHV+TC	**0.428**	**0.375**	0.560	**1.970**	**0.187**	**0.366**
HVI+TC	**0.428**	**0.375**	0.351	2.048	**0.187**	**0.366**
5/3	1.757	1.056	**0.318**	2.133	0.257	0.964

TAB. 6.1 – Mesures d'entropies (en bpp) en utilisant 2 niveaux de décomposition.

	Rect2	Crosses	Horiz	Slope	Squares	Circles
Isotrope	1.423	0.927	0.417	1.195	0.270	0.822
Laplacien	0.719	**0.369**	0.289	1.064	0.136	**0.483**
HV	1.084	0.692	0.238	1.095	0.109	0.644
HVDD	0.800	0.409	0.238	1.109	0.109	0.588
HVI	1.077	0.748	0.208	1.102	0.138	0.651
HV+TC	0.756	0.385	0.234	1.084	0.103	0.489
HVHV+TC	0.823	0.403	0.276	**1.011**	**0.101**	0.527
HVI+TC	**0.718**	**0.369**	**0.204**	1.086	0.136	**0.483**
5/3	1.714	1.077	0.216	1.350	0.135	0.874

TAB. 6.2 – Débits de codage sans perte (en bpp) en utilisant 2 niveaux de décomposition.

Images naturelles

Nous considérons désormais l'ensemble test d'images naturelles illustré en Fig.6.13.

Les Tabs. 6.3 et 6.4 présentent les entropies obtenues lors des expérimentations précédentes pour $K = 2$ et $K = 4$ niveaux de décompositions sur l'ensemble d'image naturelles. Nous remarquons que l'expérimentation HVHV + TC semble afficher les meilleurs résultats. De plus, nous observons que les expérimentations effectuées avec 4 niveaux de décomposition fournissent une représentation plus compacte que les décompositions n'utilisant que 2 niveaux.

Les débits nécessaires à l'encodage sans perte de notre ensemble test d'images naturelles sont présentés dans les Tab. 6.5 et 6.6. Encore une fois, les résultats sont cohérents

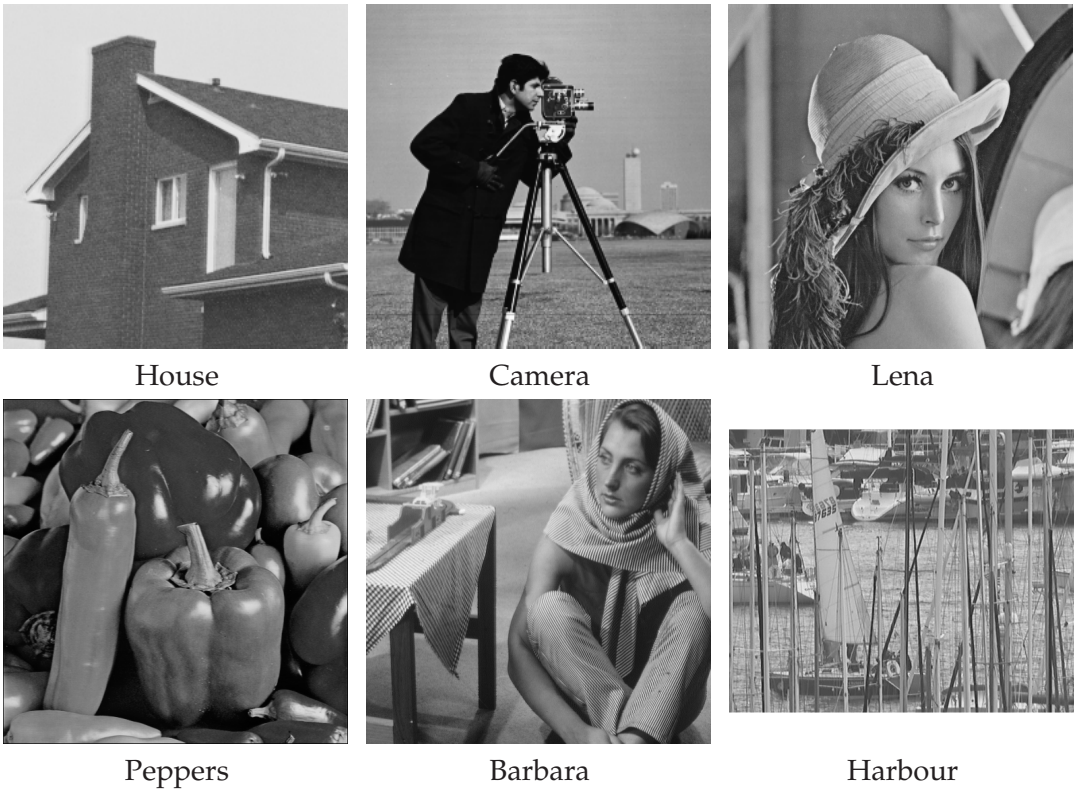


FIG. 6.13 – Ensemble test d'images naturelles.

avec les entropies observées et l'expérimentation HVHV + TC donne les meilleurs résultats de codage.

Le filtre adaptatif Laplacien donne des résultats moins bons que le schéma isotrope non-adaptatif sur les images *Lena* et *Barbara*. Dans les régions texturées, il est probable que la carte de décision associée à ce filtre oscille entre 0 et 1, laissant l'avantage au filtre uniforme.

	House	Camera	Lena	Peppers	Barbara	Harbour
Original	6.232	7.009	7.445	7.402	7.632	7.305
Isotrope	4.258	4.479	4.099	3.892	5.021	4.613
Laplacien	4.220	4.450	4.098	3.834	5.026	4.592
HV	4.363	4.545	4.211	3.931	5.097	4.593
HVDD	4.396	4.582	4.247	3.949	5.156	4.592
HVI	4.267	4.462	4.121	3.850	5.019	4.515
HV+TC	4.363	4.550	4.211	3.934	5.101	4.593
HVHV+TC	**4.052**	**4.300**	**3.917**	**3.673**	**4.857**	**4.436**
HVI+TC	4.267	4.469	4.121	3.851	5.019	4.515
5/3	4.633	4.869	4.655	4.071	5.252	4.476

TAB. 6.3 – Mesures d'entropies (en bpp) en utilisant 2 niveaux de décomposition.

Comme les exemples et les simulations le laisse présager, nos schémas de décomposition adaptatifs laissent une grande liberté dans le choix des règles de décision. De plus,

	House	Camera	Lena	Peppers	Barbara	Harbour
Isotrope	4.139	4.319	3.926	3.730	4.864	4.511
Laplacien	4.099	4.291	3.927	3.670	4.873	4.496
HV	4.257	4.403	4.051	3.779	4.958	4.500
HVDD	4.299	4.446	4.095	3.801	5.035	4.501
HVI	4.154	4.311	3.954	3.690	4.783	4.415
HV+TC	4.257	4.409	4.051	3.782	4.965	4.500
HVHV+TC	**3.913**	**4.123**	**3.727**	**3.493**	**4.692**	**4.328**
HVI+TC	4.154	4.318	3.954	3.691	4.874	4.415
5/3	4.562	4.772	4.346	3.954	5.146	4.418

TAB. 6.4 – Mesures d'entropies (en bpp) en utilisant 4 niveaux de décomposition.

	House	Camera	Lena	Peppers	Barbara	Harbour
Isotrope	3.377	3.565	3.346	3.096	4.065	3.859
Laplacien	3.375	3.565	3.346	3.055	4.076	3.843
HV	3.489	3.684	3.505	3.169	4.180	3.771
HVDD	3.546	3.738	3.581	3.204	4.268	3.774
HVI	3.410	3.597	3.408	3.079	4.088	3.693
HV+TC	3.489	3.688	3.505	3.169	4.182	3.771
HVHV+TC	**3.134**	**3.349**	**3.079**	**2.822**	**3.858**	**3.662**
HVI+TC	3.410	3.603	3.408	3.079	4.090	3.693
5/3	4.229	4.445	4.157	3.785	4.639	4.002

TAB. 6.5 – Débits de codage sans perte (en bpp) en utilisant 2 niveaux de décomposition.

	House	Camera	Lena	Peppers	Barbara	Harbour
Isotrope	3.252	3.463	3.262	3.016	3.999	3.810
Laplacien	3.244	3.461	3.262	2.974	4.011	3.802
HV	3.387	3.597	3.430	3.095	4.115	3.727
HVDD	3.455	3.658	3.515	3.135	4.210	3.731
HVI	3.300	3.503	3.331	3.001	4.023	3.644
HV+TC	3.387	3.603	3.430	3.095	4.118	3.727
HVHV+TC	**3.015**	**3.221**	**2.985**	**2.731**	**3.776**	**3.613**
HVI+TC	3.299	3.508	3.331	3.001	4.026	3.644
5/3	4.190	4.400	4.122	3.751	4.607	3.987

TAB. 6.6 – Débits de codage sans perte (en bpp) en utilisant 4 niveaux de décomposition.

la variation des pondérations et des seuils mis en jeu dans les expérimentations permet d'imposer un filtre avec une direction privilégiée, par exemple. Enfin, bien que la modification des seuils et des pondérations influence l'entropie et les débits calculés lors de ces simulations, nous avons choisis nos paramètres de manière empirique et ne les avons pas optimisés. Il serait cependant intéressant de construire un algorithme d'estimation optimale des seuils utilisés dans ces expérimentations, par minimisation des débits de codage.

6.6 Conclusion

Dans ce chapitre, nous avons construit des décompositions adaptatives utilisant des cartes de décision multivaluées, capables de "discriminer" les événements géométriques présents dans une image. Ces décompositions sont basées sur des structures lifting où l'opérateur de mise à jour est modifié à chaque pixel, selon une décision prise en fonction d'un gradient local calculé sur l'image d'entrée. Les décisions ne sont pas transmises dans le flux compressé et nous avons déterminé les conditions nécessaires et suffisantes dans plusieurs cas pour que ces décisions soient retrouvées lors de la synthèse, permettant ainsi la reconstruction parfaite de l'image originale.

Plusieurs exemples et simulations ont été mis en œuvre pour illustrer les résultats théoriques obtenus et montrer l'intérêt pratique de ces structures adaptatives non-redondantes dans des applications de compression sans perte. Nos expérimentations ont été conduites avec le codec d'image fixe EZBC et nous avons comparé nos résultats avec la transformée fixe 5/3 entière utilisée dans JPEG-2000. Nous avons alors montré le gain en efficacité de codage important offert par nos structures de décomposition adaptatives.

Conclusion générale

Cette thèse s'est inscrite dans la construction de transformées adaptées à la représentation scalable et parcimonieuse de séquences vidéos. Nous nous sommes basés essentiellement sur le schéma de codage vidéo $t + 2D$ et avons consacré nos efforts à l'optimisation et la construction de nouvelles transformées spatio-temporelles impliquées dans ce schéma.

Synthèse des travaux

1 – Optimisation du filtrage temporel

Transformée temporelle 5/3

La transformée en ondelettes $5/3$ est bidirectionnelle, possède un support plus large et constitue une candidate idéale pour assurer la transformée temporelle mise en jeu dans un schéma de codage $t + 2D$. Nous avons proposé une étude systématique sur la construction d'une transformée temporelle $5/3$ compensée en mouvement, au moyen du schéma de lifting temporel.

Sa mise en place au sein du codec MC-EZBC permet d'atteindre des gains en PSNR atteignant 1 dB par rapport à la transformée de Haar. De plus, le schéma de codage résultant possède une efficacité de codage nettement supérieure à celle offerte par les codecs vidéo hybrides MPEG-2, MPEG-4 et Windows Media 9, pourtant non-scalables.

Algorithme d'estimation de mouvement bidirectionnel conjoint

Les transformations temporelles bidirectionnelles utilisent deux champs de mouvement pour prédire certaines images à partir de leurs voisines. La plupart des schémas de codage estiment ces deux champs de mouvement séparément, par une minimisation successive de la différence des blocs des trames à gauche puis à droite.

Nous avons proposé un algorithme d'estimation jointe des champs de mouvement, qui minimise directement l'erreur de prédiction. Cet algorithme est itératif et converge rapidement, en moins de 3 itérations, vers une solution quasi-optimale. Cette optimisation a lieu au sein de l'opérateur de prédiction et peut s'appliquer dans toutes les transformées temporelles mettant en œuvre une prédiction bidirectionnelle. Elle apporte des gains importants en terme de PSNR, pour une complexité équivalente à une estimation séparée des champs de mouvement.

Transformée 5/3 uniforme et prédiction optimisée des zones découvertes

Un inconvénient majeur des transformations temporelles de Haar ou $5/3$ est leur propension à créer des zones découvertes, non-prédites par les images voisines. Ces zones

créent des artefacts fantômes sur les images décodées à bas débit, ressemblant à des réminiscences locales d'objets présents dans des images antérieures. Ces artefacts sont visuellement désagréables, complexifient le codage des images de détail et se propagent dans les images d'approximation, dégradant alors l'efficacité de la prédiction dans les niveaux suivants de la décomposition temporelle.

Nous avons proposé une transformée temporelle basée sur l'ondelette 5/3, utilisant deux champs de mouvement orientés dans la même direction : la transformée 5/3 uniforme. Par construction, elle ne crée pas de zones découvertes et n'engendre pas de tels artefacts. Elle permet d'améliorer nettement la qualité visuelle des images d'approximation et augmente de manière significative l'efficacité de codage du schéma par rapport au filtre 5/3 classique.

Comme précédemment, il est possible de mettre en œuvre un algorithme quasi-optimal d'estimation bidirectionnelle conjointe des champs de mouvement mis en jeu dans cette transformée. Cet algorithme consiste à optimiser la mise en correspondance des zones semi-découvertes lors de l'étape de prédiction, par minimisation de la distorsion des sous-bandes temporelles de détail. Le schéma résultant offre une très bonne efficacité de codage en termes de PSNR, surpassant même celle du codeur à l'état de l'art H.264/AVC, pourtant non-scalable.

Modération de la latence créée par le filtrage temporel

Les schémas de décomposition temporelle utilisés en codage vidéo introduisent un retard à l'encodage ou à la reconstruction trop important pour des applications en temps réel comme la vidéoconférence. Ce retard est introduit par les opérateurs de prédiction et de mise à jour "en avant" utilisés lors de la décomposition temporelle.

Nous avons présenté une transformée temporelle flexible basée sur le filtre 5/3 et capable de respecter un délai imposé. Sa mise en œuvre expérimentale a montré l'existence d'un compromis entre le délai imposé et l'efficacité de codage obtenue, s'étalant du cas non-contraint au cas de délai nul et offrant ainsi une large plage de possibilités en fonction des besoins de l'application.

2 – Filtrage spatial par transformées M-bandes

Construction et étude de transformées spatiales M-bandes

Dans le cadre du schéma de codage vidéo $t + 2D$, les sous-bandes temporelles sont décomposées spatialement après transformation temporelle, afin d'exploiter leur redondance spatiale. L'utilisation de la transformée biorthogonale 9/7 est justifiée pour la transformation des sous-bandes temporelles d'approximation, ressemblant à des images naturelles, mais ne l'est pas dans le cas des sous-bandes de détail qui comportent de larges zones de texture à hautes fréquences.

De part leur grande sélectivité fréquentielle et leur flexibilité, les bancs de filtres M-bandes sont des candidats idéaux pour décomposer ces sous-bandes temporelles de détail. Dans un premier temps, nous avons construit un filtre spatial 4-bandes dont la sélectivité fréquentielle et la régularité sont plus adaptées à la décomposition des images de détail que l'ondelette biorthogonale 9/7. Nous avons alors observé un gain en efficacité de codage par rapport à cette dernière. Enfin, nous avons procédé à une étude plus complète avec d'autres transformées M-bandes couramment utilisées en compression d'image (LOT, LBT, MLT).

Scalabilité fractionnaire

En dépit de leurs qualités, les bancs de filtres M-bandes ne possèdent cependant pas de propriétés de scalabilité aussi fines que leurs homologues dyadiques et ne peuvent fournir seulement que des facteurs de scalabilité d'ordre M. Afin de pallier à cet inconvénient, nous avons montré qu'une simple modification du banc de filtres de synthèse permet d'obtenir un facteur de scalabilité rationnel quelconque P/M, où P est un entier inférieur à M. Cette propriété autorise ainsi la construction de schémas de décodage dotés de scalabilité fractionnaire et permet de disposer d'une vaste gamme de facteurs de scalabilité, pour n'importe quelle transformée M-bandes et sans nécessiter un changement du banc d'analyse. Nous avons alors montré la réduction en complexité offerte par cette approche, comparée à une stratégie qui consistait à reconstruire entièrement le signal puis à le redimensionner. Enfin, nous avons prouvé expérimentalement l'intérêt de cette propriété lors d'un scénario de diffusion de contenu à destination d'un parc de récepteurs possédant des écrans de tailles différentes.

3 – Décompositions spatiales adaptatives

Dans la continuation des travaux de Piella et al. sur les décompositions adaptatives, nous avons proposé une nouvelle classe de décompositions adaptatives capables de mieux appréhender l'information contenue dans les images, à des fins de compression. Notre méthode exploite les propriétés des seminormes pour construire des schémas lifting capables de sélectionner un filtre parmi *plusieurs* filtres de mise à jour, en fonction de décisions prises sur un gradient local du signal d'entrée. Ces décisions sont calculées à partir de critères géométriques et autorisent le choix parmi plusieurs filtres de mise à jour. Nous avons alors établi les conditions nécessaires et suffisantes pour que ces décisions puissent être retrouvées lors de la synthèse, permettant ainsi une reconstruction parfaite, sans transmission d'informations annexes.

Plusieurs expérimentations de compression sans perte ont été conduites avec le codec d'image fixe EZBC afin de comparer l'efficacité de nos structures adaptatives non-redondantes avec la transformée fixe $5/3$ entière utilisée dans JPEG-2000. Les résultats expérimentaux ont alors montré le gain en efficacité de codage important offert par nos structures de décompositions adaptatives.

Perspectives

À l'issue des travaux menés tout au long de cette thèse, nous avons mis en évidence plusieurs structures de décompositions spatio-temporelles capables de donner une représentation multirésolution et parcimonieuse d'une séquence vidéo. Plusieurs problèmes restent cependant ouverts et les perspectives que nous envisageons dans le prolongement de nos travaux s'articulent autour de deux axes principaux. Dans un premier temps, il nous semble essentiel d'étendre nos structures de décompositions adaptatives au cas 3D, afin de mieux capturer la géométrie spatio-temporelle d'une séquence vidéo. Enfin, il serait intéressant de poursuivre nos travaux sur la scalabilité fractionnaire afin d'affiner et d'améliorer les propriétés de scalabilité spatiale du schéma de codage $t + 2D$.

Décompositions spatio-temporelles adaptatives

Les structures de décomposition adaptatives basées sur la combinaison de seminormes ont été expérimentées sur des images, afin d'illustrer l'intérêt de posséder des cartes de décisions multivaluées. Il est cependant tout à fait envisageable d'étendre ces structures au cas 3D, en utilisant des critères de décision et des filtres de mise à jour définis dans le pavé spatio-temporel 3D. De plus, la formulation lifting autoriserait alors assez simplement d'introduire le mouvement dans ces critères et ces filtres de mises à jour, de façon à opérer le filtrage adaptatif selon les trajectoires du mouvement. Enfin, une étude théorique des effets de la quantification au sein de ces transformées adaptatives permettrait de les mettre en œuvre dans des applications de compression d'image et de vidéo *avec perte*.

D'autres décompositions adaptatives peuvent être envisagées, comme les structures utilisant des cartes de décisions destinées à être transmises. L'inconvénient de ces transformées résidant souvent dans le coût des carte de décisions, il est possible de concevoir des décompositions spatio-temporelles adaptatives dont les cartes de décisions sont calculées à partir des champs de mouvement. Ces derniers étant encodés sans perte, la reconstruction parfaite serait alors assurée car les décisions pourraient être retrouvées lors de la synthèse.

Scalabilité fractionnaire : cas 3D et amélioration

Un inconvénient majeur des schémas de codage vidéo $t + 2D$ utilisant des décompositions spatiales en ondelettes classiques réside dans la scalabilité spatiale relativement grossière qu'ils offrent : seuls des facteurs de résolution dyadique peuvent être obtenus. Il serait souhaitable d'utiliser les filtres spatiaux M-bandes et la propriété de scalabilité fractionnaire dans ces schémas de codage vidéo afin d'obtenir une scalabilité spatiale beaucoup plus fine. Pour des raisons de performance, il serait sans doute nécessaire de mettre en place un codage scalable des champs de mouvement offrant la même finesse de scalabilité spatiale. Enfin, il serait utile d'étudier les propriétés de la scalabilité fractionnaire en présence d'un opérateur de quantification afin de concevoir des filtres de rééchantillonnage optimaux en terme d'efficacité de codage.

Annexe A : Preuves

1 – Preuve du Lemme 2

Montrons tout d'abord que si $\widetilde{\boldsymbol{c}} \neq \mathbf{0}$, alors $p_{10}(A_0) = 0$.

Preuve. Choisissons $\boldsymbol{v} = v_0\boldsymbol{a}_0 + v_1\boldsymbol{a}_1 + \tilde{\boldsymbol{v}}$ avec $\tilde{\boldsymbol{v}} \in \mathrm{Span}^{\perp}\{\boldsymbol{a}_0, \boldsymbol{a}_1\}$, tel que :

$$\left\{ \begin{array}{rcl} \boldsymbol{a}_0^T\boldsymbol{v} & = & 0 \\ \boldsymbol{a}_1^T\boldsymbol{v} & = & 1 \end{array} \right. \tag{.29}$$

Ceci implique $\|\boldsymbol{a}_0\|^2 v_0 + \boldsymbol{a}_0^T\boldsymbol{a}_1 v_1 = 0$ et $\boldsymbol{a}_1^T\boldsymbol{a}_0 v_0 + \|\boldsymbol{a}_1\|^2 v_1 = 1$; le déterminant du système (.29) vaut donc :

$$\|\boldsymbol{a}_0\|^2\|\boldsymbol{a}_1\|^2 - (\boldsymbol{a}_0^T\boldsymbol{a}_1)^2 \neq 0$$

car $\boldsymbol{a}_0$ et $\boldsymbol{a}_1$ ne sont pas colinéaires. Ceci signifie que le système possède une solution unique qui vérifie $p_1(A_0\boldsymbol{v}) = |c_1 + \tilde{\boldsymbol{c}}^T\tilde{\boldsymbol{v}}|$. Du fait que $\|\tilde{\boldsymbol{c}}\| \neq 0$, nous pouvons alors prendre $\tilde{\boldsymbol{v}} = -\frac{c_1}{\|\tilde{\boldsymbol{c}}\|^2}\tilde{\boldsymbol{c}}$, conduisant à $p_1(A_0\boldsymbol{v}) = 0$ et donc à $p_{10}(A_0) = 0$. ■

Si $\tilde{\boldsymbol{c}} = \mathbf{0}$, nous avons :

$$p_1(A_0\boldsymbol{v}) = |c_0\boldsymbol{a}_0^T\boldsymbol{v} + c_1\boldsymbol{a}_1^T\boldsymbol{v}| \geq |c_1||\boldsymbol{a}_1^T\boldsymbol{v}| - |c_0||\boldsymbol{a}_0^T\boldsymbol{v}|$$

• Si $|c_1| > |c_0|$, il faut que nous montrons $p_{10}(A_0) = |c_1| - |c_0|$.

Preuve. Choisissons $\boldsymbol{v}$ tel que :

$$\left\{ \begin{array}{rcl} \boldsymbol{a}_0^T\boldsymbol{v} & = & -\mathrm{sign}\, c_0 \\ \boldsymbol{a}_1^T\boldsymbol{v} & = & \mathrm{sign}\, c_1 \end{array} \right. \tag{.30}$$

Ces hypothèses sont compatibles avec la contrainte : $|\boldsymbol{a}_0^T\boldsymbol{v}| \leq |\boldsymbol{a}_1^T\boldsymbol{v}| = 1$. En injectant $\boldsymbol{v} = \boldsymbol{a}_0 v_0 + \boldsymbol{a}_1 v_1$, nous obtenons un système d'équations ayant le même déterminant que (.29). Il ne possède ainsi qu'une seule solution $(v_0, v_1) \in \mathbb{R}^2$ permettant à l'équation (.30) d'être vérifiée. On a alors $p_1(A_0\boldsymbol{v}) = |c_1| - |c_0| = p_{10}(A_0)$. ■

• Si $|c_1| \leq |c_0|$, montrons que $p_{10}(A_0) = 0$.

Preuve. Il suffit de trouver un vecteur $\boldsymbol{v} \in V$ tel que $|\boldsymbol{a}_1^T\boldsymbol{v}| = 1$ et $p_1(A_0\boldsymbol{v}) = c_0\boldsymbol{a}_0^T\boldsymbol{v} + c_1\boldsymbol{a}_1^T\boldsymbol{v} = 0$. Par exemple, nous pouvons choisir $\boldsymbol{v}$ tel que :

$$\left\{ \begin{array}{rcl} \boldsymbol{a}_1^T\boldsymbol{v} & = & 1 \\ \boldsymbol{a}_0^T\boldsymbol{v} & = & -\frac{c_1}{c_0} \end{array} \right.$$

Ce choix est compatible avec la condition $|\boldsymbol{a}_0^T\boldsymbol{v}| = |\frac{c_1}{c_0}| \leq 1$. En utilisant les mêmes arguments que précédemment, le système possède une solution unique. On a alors $p_{10}(A_0) = 0$. ■

En conclusion, si $\boldsymbol{a}_0$ et $\boldsymbol{a}_1$ ne sont pas colinéaires, nous avons :

$$p_{10}(A_0) = \left\{ \begin{array}{ll} |c_1| - |c_0| & \text{si } \boldsymbol{c} = c_0\boldsymbol{a}_0 + c_1\boldsymbol{a}_1 \text{ et } |c_1| > |c_0| \\ 0 & \text{sinon} \end{array} \right.$$

Dans le premier cas, comme $\boldsymbol{c} = A_0^T\boldsymbol{a}_1 = (I - \boldsymbol{u}\boldsymbol{b}_0^T)^T\boldsymbol{a}_1 = \boldsymbol{a}_1 - \boldsymbol{u}^T\boldsymbol{a}_1\boldsymbol{b}_0$, nous obtenons $\boldsymbol{u}^T\boldsymbol{a}_1\boldsymbol{b}_0 = (1 - c_1)\boldsymbol{a}_1 - c_0\boldsymbol{a}_0$. Si $\boldsymbol{u}^T\boldsymbol{a}_1 = 0$, alors $\boldsymbol{a}_0$ et $\boldsymbol{a}_1$ seraient colinéaires, ce qui est en contradiction avec les hypothèses. Sinon, si $\boldsymbol{u}^T\boldsymbol{a}_1 \neq 0$, alors $\boldsymbol{b}_0 = \dfrac{(1 - c_1)\boldsymbol{a}_1 - c_0\boldsymbol{a}_0}{\boldsymbol{u}^T\boldsymbol{a}_1}$.

2 – Preuve de la Proposition 5

Preuve. Soit $H = v_1 + v_3$ et $V = v_2 + v_4$. Nous avons alors $p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) \Leftrightarrow |H| \leq |V|$. Ceci est vérifié si et seulement si $H = V = 0$ or $V \neq 0$ and $|H/V| \leq 1$.

Supposons maintenant que $\boldsymbol{v}' = A_0\boldsymbol{v}$. Nous avons $p_0(\boldsymbol{v}') \leq p_1(\boldsymbol{v}') \Leftrightarrow |(1-2\mu_0)H - 2\eta_0 V| \leq |-2\mu_0 H + (1-2\eta_0)V|$. Ce qui est équivalent à $(H-V)\left[(1-4\mu_0)H + (1-4\eta_0)V\right] \leq 0$.

Si $V = 0$, alors $H = 0$ ou $\mu_0 \geq 1/4$. Si $V \neq 0$, nous pouvons alors écrire :

$$\left(\frac{H}{V} - 1\right)\left[(1-4\mu_0)\frac{H}{V} + 1 - 4\eta_0\right] \leq 0$$

Cette égalité ne met en jeu que la seule variable H/V. Elle est vérifiée si et seulement si :

$$\frac{H}{V} - 1 \leq 0 \quad \text{et} \quad (1-4\mu_0)\frac{H}{V} + 1 - 4\eta_0 \geq 0$$

ou si :

$$\frac{H}{V} - 1 \geq 0 \quad \text{et} \quad (1-4\mu_0)\frac{H}{V} + 1 - 4\eta_0 \leq 0$$

Nous pouvons distinguer trois cas :

– $1 - 4\mu_0 > 0$. L'inégalité précédente devient alors :

$$\frac{4\eta_0 - 1}{1-4\mu_0} \leq \frac{H}{V} \leq 1 \text{ ou } 1 \leq \frac{H}{V} \leq \frac{4\eta_0 - 1}{1-4\mu_0}$$

– $1 - 4\mu_0 = 0 \Leftrightarrow \mu_0 = 1/4$. Ceci conduit à :

$$\left[\frac{H}{V} \leq 1 \quad \text{et} \quad \eta_0 \leq \frac{1}{4}\right] \text{ ou } \left[\frac{H}{V} \geq 1 \quad \text{et} \quad \eta_0 \geq \frac{1}{4}\right]$$

– $1 - 4\mu_0 < 0$. Dans ce cas, nous obtenons :

$$\left[\frac{H}{V} \leq 1 \quad \text{et} \quad \frac{H}{V} \leq \frac{1-4\eta_0}{4\mu_0 - 1}\right] \text{ ou } \left[\frac{H}{V} \geq 1 \quad \text{et} \quad \frac{H}{V} \geq \frac{1-4\eta_0}{4\mu_0 - 1}\right]$$

En conclusion, nous avons $p_0(A_0\boldsymbol{v}) \leq p_1(A_0\boldsymbol{v}) \Leftrightarrow (H,V) \in \mathcal{A}'_0$, avec :

– si $\mu_0 < \frac{1}{4}$, alors :

$$\mathcal{A}'_0 = \Big\{(H,V) \mid V = H = 0 \text{ ou } V \neq 0 \text{ et } \left[\frac{4\eta_0 - 1}{1-4\mu_0} \leq \frac{H}{V} \leq 1 \text{ ou } 1 \leq \frac{H}{V} \leq \frac{4\eta_0 - 1}{1-4\mu_0}\right]\Big\}.$$

– si $\mu_0 = \frac{1}{4}$ et $\eta_0 \leq 1/4$, alors $\mathcal{A}'_0 = \Big\{(H,V) \mid V = 0 \quad \text{ou} \quad \left[V \neq 0 \text{ and } \frac{H}{V} \leq 1\right]\Big\}$.

– si $\mu_0 = \frac{1}{4}$ et $\eta_0 > 1/4$, alors $\mathcal{A}'_0 = \Big\{(H,V) \mid V = 0 \quad \text{ou} \quad \left[V \neq 0 \text{ and } \frac{H}{V} > 1\right]\Big\}$.

– si $\mu_0 > \frac{1}{4}$, alors :

$$\mathcal{A}'_0 = \Big\{(H,V) \mid V = 0 \text{ ou } V \neq 0 \text{ et } \left[\left(\frac{H}{V} \leq 1 \text{ et } \frac{H}{V} \leq \frac{4\eta_0 - 1}{1-4\mu_0}\right) \text{ ou } \left(\frac{H}{V} \geq 1 \text{ et } \frac{H}{V} \geq \frac{4\eta_0 - 1}{1-4\mu_0}\right)\right]\Big\}.$$

Au final, nous obtenons pour tout $\boldsymbol{v}$ la première condition permettant de satisfaire la propriété de reconstruction parfaite : $p_0(\boldsymbol{v}) \leq p_1(\boldsymbol{v}) \Rightarrow p_0(A_0\boldsymbol{v}) \leq p_1(A_0\boldsymbol{v})$ si et seulement si $\mathcal{A}_0 \subset \mathcal{A}'_0$, avec

$\mathcal{A}_0 = \Big\{(V,H) \mid V = H = 0 \quad \text{ou} \quad \big[V \neq 0 \text{ et } \big|\frac{H}{V}\big| \leq 1\big]\Big\}$. Nous pouvons examiner les quatres cas précédents :

– si $\mu_0 < \dfrac{1}{4}$, alors $\mathcal{A}_0 \subset \mathcal{A'}_0 \Leftrightarrow \dfrac{4\eta_0 - 1}{1 - 4\mu_0} \leq -1 \Leftrightarrow \eta_0 \leq \mu_0$.

– si $\mu_0 = \dfrac{1}{4}$ et $\eta_0 \leq \dfrac{1}{4}$, alors $\mathcal{A}_0 \subset \mathcal{A'}_0$.

– si $\mu_0 = \dfrac{1}{4}$ et $\eta_0 > \dfrac{1}{4}$, alors $\mathcal{A}_0 \not\subset \mathcal{A'}_0$.

– si $\mu_0 > \dfrac{1}{4}$, alors $\mathcal{A}_0 \subset \mathcal{A'}_0 \Leftrightarrow \dfrac{4\eta_0 - 1}{1 - 4\mu_0} \geq 1 \Leftrightarrow \eta_0 + \mu_0 \leq \dfrac{1}{2}$.

Enfin, $\mathcal{A}_0 \subset \mathcal{A'}_0$ si et seulement si $\eta_0 \leq \mu_0 \leq 1/4$ ou $[\mu_0 > 1/4$ et $\mu_0 + \eta_0 \leq 1/2]$. De plus, nous avons la condition $\alpha_0 \neq 0 \Leftrightarrow \mu_0 + \eta_0 \neq 1/2$. Ceci prouve la première condition de l'énoncé de la Proposition 5.

Afin de prouver la seconde condition, nous utilisons la propriété de symétrie déjà constatée dans la Remarque 1. En remplaçant $\boldsymbol{b}_0$ par $-\boldsymbol{b}_1/\alpha_1$, nous obtenons alors :

$$\frac{-\eta_1}{1 - 2\mu_1 - 2\eta_1} \leq \frac{-\mu_1}{1 - 2\mu_1 - 2\eta_1} \quad \text{et} \quad -\frac{\mu_1 + \eta_1}{1 - 2\mu_1 - 2\eta_1} \leq \frac{1}{2}$$

Il est alors simple de montrer que cette relation est équivalente à la seconde condition de la Proposition 5. ■

Annexe B : Filtrage temporel M-bandes et codage H.264

Cette annexe reproduit un article de revue [21] où nous relatons nos travaux sur des filtres temporels M-bandes mis en œuvre au sein du codec H.264, afin de lui fournir des propriétés de scalabilité temporelle. Son contenu se situe légèrement en dehors de nos axes de recherches et n'a pas été développé dans le manuscrit afin d'éviter de le surcharger.

Publications

Articles de revues internationales

1. C. Bergeron, C. Lamy-Bergot, G. Pau, and B. Pesquet-Popescu. Temporal scalability through adaptive M-band filterbanks for robust H.264/AVC video coding. À paraître dans *Hindawi Journal of Applied Signal Processing*, 2006.
2. G. Piella, B. Pesquet-Popescu, H. Heijmans, and G. Pau. Combining seminorms in adaptive lifting schemes and applications to image analysis and compression. À paraître dans *Springer Journal of Mathematical Imaging and Vision*, 2006.
3. G. Pau, B. Pesquet-Popescu, and G. Piella. Modified M-band synthesis filter bank for fractional scalability of images. *IEEE Signal Processing Letters*, Juin 2006.
4. G. Pau, C. Tillier, B. Pesquet-Popescu, and H. Heijmans. Motion compensation and scalability in lifting-based video coding. *Signal Processing : Image Communication*, 19 :577–600, Août 2004. Special issue on Wavelet Video Coding.

Articles de conférences internationales

1. G. Pau and B. Pesquet-Popescu. Image coding with rational spatial scalability. In *Proc. of EUSIPCO*, Florence, Italie, Septembre 2006.
2. G. Piella, G. Pau, and B. Pesquet-Popescu. Adaptive lifting schemes combining seminorms for lossless image compression. In *Proc. of the IEEE Int. Conf. on Image Processing*, Gênes, Italie, Septembre 2005.
3. G. Pau, J. Viéron, and B. Pesquet-Popescu. Video coding with flexible MCTF structures for low end-to-end delay. In *Proc. of the IEEE Int. Conf. on Image Processing*, Gênes, Italie, Septembre 2005.
4. G. Pau and B. Pesquet-Popescu. Comparison of spatial M-band filter banks for t+2D video coding. In *Proc. of SPIE Visual Communications and Image Processing*, Beijing, China, July 2005.
5. G. Pau and B. Pesquet-Popescu. Four-band linear-phase orthogonal spatial filter bank for subband video coding. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Philadelphie, PA, Mars 2005.
6. G. Pau and B. Pesquet-Popescu. Optimized prediction of uncovered areas in subband video coding. In *Proc. of the Picture Coding Symposium*, Décembre 2004.
7. G. Pau and B. Pesquet-Popescu. Uniform motion-compensated 5/3 filterbank for subband video coding. In *Proc. of the IEEE Int. Conf. on Image Processing*, Singapour, Octobre 2004.
8. G. Pau, B. Pesquet-Popescu, M. van der Schaar, and J. Viéron. Delay-performance trade-offs in motion-compensated scalable subband video compression. In *Proc. of Advanced Concepts for Intelligent Vision Systems (ACIVS)*, Septembre 2004.

9. G. Pau, C. Tillier, and B. Pesquet-Popescu. Optimization of the predict operator in lifting-based motion compensated temporal filtering. In *Proc. of SPIE Visual Communications and Image Processing*, San Jose, CA, Janvier 2004.

Contributions MPEG

1. G. Pau, S. Brangoulo, and B. Pesquet-Popescu. Integration of bidirectional joint motion estimation for Vidwav software. Doc. M13011, MPEG 75th meeting, Janvier 2006.
2. G. Pau and B. Pesquet-Popescu. Proposal of Vidwav OBMC bug fix. Doc. M12616, MPEG 74th meeting, Octobre 2005.
3. R. Xiong X. Ji, D. Zhang, J. Xu, G. Pau, M. Trocan and V. Bottreau. Vidwav Wavelet Video Coding Specifications. Doc. M12339, MPEG 73th meeting, Juillet 2005.
4. G. Pau, M. Trocan and B. Pesquet-Popescu. Bidirectional Joint Motion Estimation for Vidwav Software. Doc. M12303, MPEG 73th meeting, Juillet 2005.
5. V. Bottreau, G. Pau and J. Xu. Vidwav evaluation software manual. Doc. M12176, MPEG 73th meeting, Juillet 2005.
6. G. Pau and B. Pesquet-Popescu. Comparison of Spatial M-band Filter Banks for t+2D Video Coding. Doc. M12176, MPEG 72th meeting, Avril 2005.
7. C. Tillier, G. Pau and B. Pesquet-Popescu. Coding performance comparison of entropy coders in wavelet video coding. Doc. M12056, MPEG 72th meeting, Avril 2005.
8. G. Pau, J. Viéron and B. Pesquet-Popescu. Wavelet Video Coding with Flexible MCTF Structures for Low End-to-End Delay. Doc. M11741, MPEG 71th meeting, Janvier 2005.
9. G. Pau and B. Pesquet-Popescu. Four-Band Linear-Phase Orthogonal Spatial Filter Bank in Wavelet Video Coding. Doc. M11739, MPEG 71th meeting, Janvier 2005.
10. G. Pau and B. Pesquet-Popescu. Optimized Prediction of Uncovered Areas in Wavelet Video Coding. Doc. M11738, MPEG 71th meeting, Janvier 2005.
11. J. Viéron, G. Boisson, E. François, G. Pau and B. Pesquet-Popescu Proposal for SVC CE1 : Time and Level adaptive MCTF architectures for low delay video coding. Doc. M11673, MPEG 71th meeting, Janvier 2005.
12. V. Bottreau, E. François, S. Pateux, G. Pau, B. Timmerman and M. Wien. SVC CE1 - "DANAE + Thomson" verification of MSRA contribution. Doc. M11364, MPEG 70th meeting, Octobre 2004.
13. DANAE partners. SVC CE1 - Technical Description of the "DANAE + Thomson" Proposal. Doc. M11363, MPEG 70th meeting, Octobre 2004.
14. G. Pau and B. Pesquet-Popescu. Response to the SVC CE-3 on coding efficiency with low delay constraints. Doc. M11343, MPEG 70th meeting, Octobre 2004.
15. M. Trocan, G. Pau and B. Pesquet-Popescu. Cross-Verification of RWTH Results on SVC CE-4. Doc. M11318, MPEG 70th meeting, Octobre 2004.
16. G. Pau and B. Pesquet-Popescu. Cross-verification of RWTH results on SVC CE-1e. Doc. M11085, MPEG 69th meeting, Juillet 2004.
17. G. Pau, B. Pesquet-Popescu, M. van der Schaar and J. Viéron. Delay-Performance Trade-Offs in Motion-Compensated Scalable Subband Video Compression. Doc. M11084, MPEG 69th meeting, Juillet 2004.
18. G. Pau, C. Tillier and B. Pesquet-Popescu. Motion-compensated scalable subband video codec with optimized temporal prediction and update operators - A Standalone Tool in Response to the CfP on Scalable Video Coding Technology. Doc. M10537, MPEG 68th meeting, Mars 2004.

Table des figures

0.1 Schéma de principe d'un encodeur vidéo $t+2D$. 11

1.1 Scalabilité spatiale. Exemples de facteurs de résolution dyadiques obtenus avec le codec scalable JPEG-2000. 16
1.2 Scalabilité en qualité. Exemples de différentes qualités obtenues avec le codec JPEG-2000 lors du décodage à différents débits, exprimés en bits par pixel. 16
1.3 Banc de filtres d'analyse en quadrature miroir. 20
1.4 Banc de filtres d'analyse assurant une décomposition en ondelettes sur $j_{max} = 3$ niveaux de résolution. Elle correspond à une projection sur $\boldsymbol{V}_3$, $\boldsymbol{W}_3$, $\boldsymbol{W}_2$ et $\boldsymbol{W}_1$. 21
1.5 Banc de filtres de synthèse. 22
1.6 Banc de filtres d'analyse-synthèse. 23
1.7 Ondelette de Haar (gauche) et ondelette Daubechies-4 (droite). 24
1.8 Ondelette CDF 5/3 d'analyse ψ et sa duale $\widetilde{\psi}$. 25
1.9 Ondelette CDF 9/7 d'analyse ψ et sa duale $\widetilde{\psi}$. 26
1.10 Décomposition successive d'une image en ondelettes sur trois niveaux. On remarquera la disposition pyramidale des sous-bandes d'approximation A_k de niveau k et des sous-bandes de détail horizontal H_k, vertical V_k et diagonal D_k. 27
1.11 Décomposition en ondelettes séparables biorthogonales 9/7 de *Lena* sur 3 niveaux de résolution. 27
1.12 Prise en compte par le codec emboîté EZW de la dépendance hiérarchique spatiale des coefficients d'ondelettes entre niveaux de résolutions. 28
1.13 Structure d'analyse en lifting à deux étages. 31
1.14 Structure de synthèse en lifting. 32
1.15 Stencils : supports de prédiction utilisés dans la représentation de Cohen et Matei. Les pixels de support sont en clair tandis que ceux qui seront prédits sont grisés. 38
1.16 Étapes de prédiction adaptative et de mise à jour utilisées dans la décomposition en ondelettes orientées de Chappelier. 38

2.1 Schéma de principe d'un encodeur vidéo hybride avec boucle de rétroaction. 42
2.2 Agencement des modes de prédiction IBBPBBP d'un groupe d'images. . . 43
2.3 Structure de l'encodeur du schéma de codage SVC. 45
2.4 Schéma de principe d'un encodeur vidéo $t+2D$. 47
2.5 Filtrage de Haar compensé en mouvement selon la technique de Ohm. . . 48
2.6 Décomposition temporelle de Haar d'un groupe d'images sur 3 niveaux. . 50
2.7 Organisation hiérarchique du flux vidéo *bitstream* compressé MC-EZBC. . 53

3.1 Structure lifting d'un filtre temporel compensé en mouvement. 63

3.2 Opérateur de prédiction mis en jeu dans la transformée 5/3. Tous les pixels $\boldsymbol{n}_k$ sont connectés des deux côtés. 64
3.3 Opérateur de mise à jour utilisé dans la transformée 5/3. Le pixel $\boldsymbol{m}_1$ est connecté des deux côtés, les pixels $\boldsymbol{m}_2$ et $\boldsymbol{m}_3$ sont simplement connectés tandis que le pixel $\boldsymbol{m}_4$ n'est pas connecté. 65
3.4 Analyse multirésolution temporelle 5/3 sur 3 niveaux d'une séquence vidéo. 70
3.5 Analyse multirésolution temporelle 5/3 sur 3 niveaux d'un extrait de la séquence *Foreman*. 71
3.6 Schéma de fonctionnement du module de traitement au fil de l'eau de la transformée temporelle 5/3. On y observe l'évolution du buffer cyclique de filtrage. 73
3.7 Zooms sur une région d'une image d'approximation du quatrième niveau temporel obtenu avec une transformée temporelle de Haar (gauche) et une transformée 5/3 (droite) sur la séquence *Tempête* au format CIF. 75

4.1 Opérateur de prédiction mis en jeu dans la transformée 5/3 et minimisation de la distorsion du bloc $\mathcal{B}$. 79
4.2 Courbes et tableaux de débit-distorsion obtenus pour différents filtres temporels et différents débits sur la séquence *Stefan* CIF 30 Hz. 85
4.3 Courbes et tableaux de débit-distorsion obtenus pour différents filtres temporels et différents débits sur la séquence *Mobile* CIF 30 Hz. 86
4.4 Courbes et tableaux de débit-distorsion obtenus pour différents filtres temporels et différents débits sur la séquence *Tempête* CIF 30 Hz. 87
4.5 Courbes et tableaux de débit-distorsion obtenus pour différents filtres temporels et différents débits sur la séquence *Foreman* CIF 30 Hz. 88
4.6 Comparaison de l'évolution du PSNR des images reconstruites de la séquence *Vintage Car* décodée à la résolution 704×896 à 30 Hz avec un débit de 1024 kbs, avec les codecs Vidwav et SVC. 90
4.7 Comparaison de l'évolution du PSNR des images reconstruites de la séquence *Vintage Car* décodée à la résolution 352×448 à 30 Hz avec un débit de 384 kbs, avec les codecs Vidwav et SVC. 91
4.8 Reconstruction d'une image issue du codage de la séquence *Vintage Car* de résolution 704×896 à 30 Hz pour un débit de 1024 kbs avec le codec Vidwav muni de l'estimation jointe (gauche) et avec le codec SVC JSVM 2.0 (droite). 92
4.9 Présence d'artefacts fantômes sur les sous-bandes d'approximation issues du troisième niveau de la décomposition temporelle 5/3 de la séquence CIF 30 Hz *Stefan*. 93
4.10 Opérateur de prédiction (gauche) et de mise à jour (droite) mis en jeu dans la transformée temporelle 5/3. 95
4.11 Orientation du mouvement dans la transformée 5/3 uniforme. 96
4.12 Opérateur de prédiction mis en jeu dans la transformée 5/3 uniforme. Le pixel $\boldsymbol{n}_1$ est simplement connecté tandis que le pixel $\boldsymbol{n}_2$ est connecté des deux côtés. 97
4.13 Opérateur de mise à jour utilisé dans la transformée 5/3 uniforme. Le pixel $\boldsymbol{n}_1$ est simplement connecté tandis que le pixel $\boldsymbol{n}_2$ est connecté des deux côtés. 97

4.14 Compensation inverse d'une image $\mathcal{C}^{-1}(x_{2t}, \boldsymbol{v}_{2t})$ provenant de la séquence *Foreman*. Les zones noires représentent les zones découvertes. 99
4.15 Images d'approximation du quatrième niveau issues de la décomposition temporelle de la séquence *Stefan* CIF 30 Hz obtenue avec le filtre 5/3 classique (à gauche) et avec le filtre 5/3 uniforme (à droite). 102
4.16 Délai maximal d'encodage d'une trame dans une analyse temporelle 5/3 à 3 niveaux. Le chemin en gras désigne le chemin de traitement lié à l'image ayant le plus grand délai d'encodage $N_e = 14$. 106
4.17 Délais maximaux de décodage et de reconstruction d'une trame dans une synthèse temporelle 5/3 à 3 niveaux. Le chemin en gras désigne le chemin de traitement lié à l'image ayant le plus grand délai de décodage $N_d = 11$ et de reconstruction $N_r = 21$. 107
4.18 Délai maximal d'encodage d'une trame dans l'analyse temporelle à 3 niveaux par la transformée $(P, Q) = (1, 1)$. 112
4.19 Délais maximaux de décodage et de reconstruction d'une trame dans la synthèse à 3 niveaux par la transformée $(P, Q) = (1, 1)$. 113
4.20 Structure lifting de la transformée en ondelettes de Daubechies-4 115
4.21 Décomposition temporelle en ondelettes Daubechies-4 116
4.22 Schéma de fonctionnement du module de traitement au fil de l'eau de la transformée temporelle Daubechies-4. On y observe l'évolution du buffer cyclique de filtrage. 117

5.1 Banc de filtres d'analyse-synthèse M-bandes. 122
5.2 Réponses fréquentielles des filtres d'analyse $\{h_k\}_{0 \leq k < 4}$ de la MLT 4-bandes MLT (gauche) et de la MLBT 4-bandes (droite). 127
5.3 Sous-bandes issues de la décomposition temporelle 5/3 sur 4 niveaux de la séquence *Foreman* CIF. De gauche à droite et de haut en bas, on observe successivement une sous-bande de détail du premier, deuxième et troisième niveau, suivie d'une sous-bande d'approximation du quatrième niveau temporel. 128
5.4 Réponses fréquentielles des filtres d'analyse $\{h_k\}_{0 \leq k < 4}$ des bancs de filtres 4-bandes FB1 (gauche) et FB2 (droite). 133
5.5 Fonction d'échelle et ondelettes associées au banc de filtres 4-bandes FB1. 134
5.6 Opérateur de rééchantillonnage $[\downarrow \frac{M}{P}]$. 140
5.7 Banc de filtres M-bandes d'analyse et de synthèse. Ce dernier est en mesure de fournir une image réduite d'un facteur M/P. 142
5.8 Reconstruction à pleine résolution par un banc de synthèse M-bandes avec une étape de redimensionnement. 143
5.9 Illustration de différents facteurs de scalabilité M/P obtenus par la méthode proposée sur l'image *Lena* avec $M = 8$, $P = \{2, 3, 4, 5, 6, 7\}$ et $Q = P$. 145
5.10 Illustration de différentes reconstructions en utilisant un nombre variable de sous-bandes Q sur l'image *Barbara* avec $M = 8$, $P = 5$ et $Q = \{1, 2, 3, 4, 5, 8\}$. 146
5.11 Influence du filtre de rééchantillonnage w_e et w_t, illustrée par différentes reconstructions en utilisant un nombre de variable de sous-bandes Q sur l'image *Barbara* avec $M = 8$, $P = 5$ et $Q = \{3, 4, 8\}$. 147

5.12 Bases de décomposition utilisées pour les transformées 8-bandes (gauche) et pour la transformée dyadique 9/7 (droite). Q et Q' dénotent le nombre de sous-bandes utilisées pour la reconstruction. 148
5.13 Zooms sur des images reconstruites à pleine résolution ($P = Q = 8$) de *Lena* avec différentes transformées, à un débit de 0.1 bpp. 149
5.14 Zooms sur des images reconstruites à la résolution réduite de 8/3 ($P = Q = 3$) de *Lena* à 0.4 bpp. 152
5.15 Zooms sur des images reconstruites à la résolution réduite de 8/6 ($P = Q = 6$) de *Barbara* à 0.4 bpp. 152

6.1 Structure d'analyse lifting avec mise à jour adaptative. 157
6.2 Indexation des échantillons dans une fenêtre 3×3 centrée sur $x(\boldsymbol{n})$. 163
6.3 Exemple d'indexation des échantillons d'un signal monodimensionnel. . . 169
6.4 Structure lifting d'une décomposition 2D réalisant une étape de mise à jour adaptative U_d suivie de trois étapes de prédiction P_1, P_2 et P_3. 171
6.5 Indexation des échantillons dans une fenêtre 3×3 centrée sur $x(\boldsymbol{n})$. 172
6.6 Décomposition multirésolution par filtrage isotrope non-adaptatif (gauche) et par sélection adaptative HV entre un filtre horizontal et vertical (droite). 173
6.7 Sous-bandes mises en jeu dans une décomposition multirésolution par filtrage adaptatif HV et comparées à celles issues d'une décomposition isotrope non-adaptative. 174
6.8 Image originale et cartes de décision associée à l'expérimentation HVDD (gauche). Décomposition HVDD sur un niveau spatial (droite). 175
6.9 Partitions de la carte de décision associées à l'expérimentation HVI. 176
6.10 Décomposition par sélection adaptative entre un filtre horizontal, vertical et isotrope HVI (gauche) et par sélection adaptative HV + TC entre un filtre horizontal et vertical, combiné à un critère de seuil (droite). 177
6.11 Décomposition par sélection adaptative HVHV + TC entre quatre filtres, combinée à un critère de seuil (droite). Cartes de décisions issues de la décomposition adaptative HVI + TC par sélection entre le filtre horizontal, vertical et isotrope, combinée à un critère de seuil (gauche). 178
6.12 Ensemble test d'images synthétiques. 179
6.13 Ensemble test d'images naturelles. 181

Bibliographie

[1] ISO/IEC 11172-2 :1993. Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part 2 : Video. 1993.

[2] ISO/IEC 13818-2 :1996. Information technology – Generic coding of moving pictures and associated audio information : Video. 1996.

[3] ISO/IEC 14496-2 :2001. Information technology – Coding of audio-visual objects – Part 2 : Visual. 2001.

[4] ISO/IEC 14496-2 :2001/AMD.2. Streaming video profile. 2001.

[5] ISO/IEC 15444-1 :2004. JPEG-2000 Standard : Core coding system. 2004.

[6] MPEG Video AhG. Call for proposals on scalable video coding technology. Doc. N5958, Brisbane MPEG 66th meeting, Octobre 2003.

[7] MPEG Video AhG. Registered responses to the cfp on scalable video coding. Doc. M10569, Munich MPEG 68th meeting, Mars 2004.

[8] MPEG Video SVC AhG. Call for proposals on scalable video coding technology. Doc. N6193, Hawaii MPEG meeting, Décembre 2003.

[9] MPEG Wavelet Video AhG. Wavelet codec reference document and software manual. Doc. N7334, Poznan MPEG 73th meeting, Juillet 2005.

[10] I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang. Video transcoding : An overview of various techniques and research issues. *IEEE Transactions on Multimedia*, 7(5) :793 – 803, Octobre 2005.

[11] N. Ahmed, T. Natarjan, and K. R. Rao. Discrete cosine transform. *IEEE Transactions on Computers*, 23(1) :90–93, Janvier 1974.

[12] O. Alkin and H. Caglar. Design of efficient M-band coders with linear-phase and perfect-reconstruction properties. *IEEE Transactions on Signal Processing*, 43 :1579–1590, 1995.

[13] T. André, M. Cagnazzo, M. Antonini, M. Barlaud, N. Bozinovic, and J. Konrad. (N,0) motion-compensated lifting-based wavelet transform. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Montréal, Canada, Mai 2004.

[14] Y. Andreopoulos, A. Munteanu, J. Barbarien, M. van der Schaar, J. Cornelis, and P. Schelkens. In-band motion compensated temporal filtering. *Signal Processing : Image Communication*, 19 :653–673, Août 2004. Special issue on Wavelet Video Coding.

[15] Y. Andreopoulos, A. Munteanu, G. van der Auwera, P. Schelkens, and J. Cornelis. Scalable wavelet video-coding with in-band prediction - Implementation and experimental results. In *Proc. of the IEEE Int. Conf. on Image Processing*, Rochester, New York, Septembre 2002.

[16] Y. Andreopoulos, A. Munteanu, G. van der Auwera, P. Schelkens, and J. Cornelis. Wavelet-based fully scalable video coding with in-band prediction. In *Proc. of IEEE Benelux Signal Processing Symposium*, Louvain, Belgique, Mars 2002.

[17] T. André, B. Pesquet-Popescu, M. Gastaud, M. Antonini, and M. Barlaud. Motion estimation using chrominance for wavelet-based video coding. In *Proc. of the Picture Coding Symposium*, San Francisco, CA, Décembre 2004.

[18] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on Image Processing*, 1(2) :205–220, Avril 1992.

[19] A. Baussard, F. Nicolier, and F. Truchetet. Rational multiresolution analysis and fast wavelet transform : application to wavelet shrinkage denoising. *Signal Processing*, 84 :1735–1747, 2004.

[20] F. Bellard. FFMpeg - Multimedia system. Logiciel disponible sur le site *http ://ffmpeg.sourceforge.net/index.php*, Mars 2006.

[21] C. Bergeron, C. Lamy-Bergot, G. Pau, and B. Pesquet-Popescu. Temporal scalability through adaptive M-band filterbanks for robust H.264/AVC video coding. À paraître dans *Hindawi Journal of Applied Signal Processing*, 2006.

[22] T. Blu. A new design algorithm for two-band orthonormal rational filter banks and orthonormal rational wavelets. *IEEE Transactions on Signal Processing*, 46 :1494–1504, 1998.

[23] G. Boisson, E. François, and C. Guillemot. Accuracy-scalable motion coding for efficient scalable video compression. In *Proc. of the IEEE Int. Conf. on Image Processing*, Singapour, Octobre 2004.

[24] V. Bottreau, M. Benetière, B. Felts, and B. Pesquet-Popescu. A fully scalable 3D subband video codec. In *Proc. of the IEEE Int. Conf. on Image Processing*, Thessalonique, Grèce, Octobre 2001.

[25] S. Brangoulo, R. Leonardi, T. Oelbaum, and J. Xu. Exploration experiments in wavelet video coding. Doc. N7572, Nice MPEG 74th meeting, Octobre 2005.

[26] M. Cagnazzo, T. André, M. Antonini, and M. Barlaud. A smoothly scalable and fully JPEG-2000-compatible video coder. In *Proc. IEEE International Workshop on Multimedia Signal Processing*, pages 91–94, Sienne, Italie, Septembre 2004.

[27] M. Cagnazzo, F. Castaldo, T. Andre, M. Antonini, and M. Barlaud. Optimal motion estimation for wavelet video coding. À paraître dans *IEEE Transactions on Circuits and Systems for Video Technology*, Mai 2006.

[28] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. *Applied and Computational Harmonic Analysis*, 5 :332–369, Août 1998.

[29] E. Candès. The curvelet transform for image denoising. In *Proceedings of the IEEE International Conference on Image Processing*, page 7, Thessaloniki, Greece, October 7-10, 2001.

[30] E. J. Candès and D. L. Donoho. Curvelets - A surprisingly effective nonadaptive representation for objects with edges, curves and surfaces. *Curves and Surfaces*, Juin 1999. Vanderbilt University Press, Nashville, TN.

[31] E. J. Candès and D. L. Donoho. Ridgelets : A key to higher-dimensional intermittency ? *Phil. Trans. R. Soc. Lond. A*, 357(1760) :2495–2509, Septembre 1999.

[32] V. Chappelier and C. Guillemot. Oriented wavelet transform on a quincunx pyramid for image compression. In *Proc. of the IEEE Int. Conf. on Image Processing*, Gênes, Italie, Septembre 2005.

[33] V. Chappelier, C. Guillemot, and S. Marinkovic. Image coding with iterated contourlet and wavelet transforms. In *Proc. of the IEEE Int. Conf. on Image Processing*, Singapour, Octobre 2004.

[34] S.-R. Chen, C.-P. Chang, and C.-W. Lin. MPEG-4 FGS coding performance improvement using adaptive inter-layer prediction. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Montréal, Canada, Mai 2004.

[35] T. Chen and P. P. Vaidyanathan. Vector space framework for unification of one- and multidimensional filter bank theory. *IEEE Transactions on Signal Processing*, 42 : 2006–2021, 1994.

[36] W. H. Chen, C. H. Smith, and S. C. Fralick. A fast computational algorithm for the discrete cosine transform. *IEEE Transactions on Communications*, 25(9) :1004–1009, Septembre 1977.

[37] Y.-J. Chen, S. Oraintara, and T. Nguyen. Video compression using integer DCT. In *Proc. of the IEEE Int. Conf. on Image Processing*, Vancouver, Canada, Septembre 2000.

[38] S. J. Choi and J. W. Woods. Motion-compensated 3-D subband coding of video. *IEEE Transactions on Image Processing*, 8(2) :155–167, Février 1999.

[39] C. Chrysafis and A. Ortega. Line-based, reduced memory, wavelet image compression. *IEEE Transactions on Image Processing*, 9(3) :378–388, Mars 2000.

[40] R. L. Claypoole, G. M. Davis, W. Sweldens, and R. G. Baraniuk. Nonlinear wavelet transforms for image coding via lifting. *IEEE Transactions on Image Processing*, 12 (12) :1449–1459, Décembre 2003.

[41] A. Cohen, I. Daubechies, O. G. Guleryuz, and M. T. Orchard. On the importance of combining wavelet-based nonlinear approximation with coding strategies. *IEEE Transactions on Information Theory*, 48(7) :1895–1921, Juillet 2002.

[42] A. Cohen and B. Matei. Compact representation of images by edge adapted multiscale transforms. In *Proc. of the IEEE Int. Conf. on Image Processing*, Thessaloniki, Grèce, Octobre 2001.

[43] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4(3) :245–267, 1998.

[44] W. De Neve, P. Lambert, S. Lerouge, and R. Van de Walle. Assessment of the compression efficiency of the MPEG-4 AVC specification. In *Proc. of SPIE Visual Communications and Image Processing*, volume 5308, pages 1082–1093, San Jose, Janvier 2004.

[45] R. L. de Queiroz, T. Q. Nguyen, and K. R. Rao. Generalised lapped orthogonal transforms. *Electronics Letters*, 30(2) :107–108, Janvier 1994.

[46] M. N. Do and M. Vetterli. Finite ridgelet transform for image compression. *IEEE Transactions on Image Processing*, 12(1) :16–28, Janvier 2003.

[47] M. N. Do and M. Vetterli. The contourlet transform : an efficient directional multiresolution image representation. *IEEE Transactions on Image Processing*, 14(12) : 2091–2106, Décembre 2005.

[48] D. L. Donoho and M. R. Duncan. Digital curvelet transform : Strategy, implementation and experiments. In *Proc. of SPIE*, volume 4056, pages 12–29, Novembre 2000.

[49] P. L. Dragotti and M. Vetterli. Wavelet footprints : Theory, algorithms and applications. *IEEE Transactions on Signal Processing*, 51(5), Mai 2003.

[50] E. Francois, G. Marquant, N. Burdin, and J. Viéron. Extended spatial scalability. doc. m11669, Hong Kong MPEG meeting, Janvier 2005.

[51] Ö. N. Gerek and A. E. Cetin. Adaptive polyphase subband decomposition structures for image compression. *IEEE Transactions on Image Processing*, 9 :1649–1659, October 2000.

[52] B. Girod and S. Han. Optimum motion-compensated lifting. *IEEE Signal Processing Letters*, 12(2), Février 2005.

[53] A. Golwelkar. Motion compensated temporal filtering and motion vector coding using longer filters. PhD thesis, RPI, ECSE Dept., Septembre 2004.

[54] A. Golwelkar and J. W. Woods. Scalable video compression using longer motion compensated temporal filters. In *Proc. of Visual Communications and Image Processing*, Lugano, Suisse, Juillet 2003.

[55] R. Gopinah and C. S. Burrus. On upsampling, downsampling, and rational sampling rate filter banks. *IEEE Transactions on Signal Processing*, 42(4) :812–824, Avril 1994.

[56] A. Gouze, M. Antonini, M. Barlaud, and B. Macq. Design of signal-adapted multidimensional lifting scheme for lossy coding. *IEEE Transactions on Image Processing*, 13(12) :1589–1603, Décembre 2004.

[57] K. Hanke, J.-R. Ohm, and T. Rusert. Adaptation of filters and quantization in spatio-temporal wavelet coding with motion compensation. In *Proc. of the Picture Coding Symposium*, pages 49–54, St. Malo, France, Avril 2003.

[58] H. J. A. M. Heijmans and J. Goutsias. Nonlinear multiresolution signal decomposition schemes : Part II : morphological wavelets. *IEEE Transactions on Image Processing*, 9(11) :1897–1913, 2000.

[59] H. J. A. M. Heijmans, B. Pesquet-Popescu, and G. Piella. Building nonredundant adaptive wavelets by update lifting. Research Report PNA-R0212, CWI, 2002.

[60] S. Hsiang and J. Woods. Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling. In *Proc. of the IEEE International Symposium on Circuits and Systems*, pages 662–665, Genève, Suisse, Mai 2000.

[61] S.-T. Hsiang and J. W. Woods. Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank. *Signal Processing : Image Communication*, 16(8) :705–724, Mai 2001. Special issue on Wavelet Video Coding.

[62] S.-T. Hsiang, J.W. Woods, and J.-R. Ohm. Invertible temporal subband/wavelet filter banks with half-pixel-accurate motion compensation. *IEEE Transactions on Image Processing*, 13 :1018–1028, Août 2004.

[63] J. Hua, Z. Xiong, and X. Wu. High-performance 3-D embedded wavelet video (EWV) coding. In *Proc. of IEEE Workshop on Multimedia Signal Processing*, pages 569–574, Cannes, France, Octobre 2001.

[64] G. Karlsson and M. Vetterli. Three-dimensional subband coding of video. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, volume 3, pages 1100–1103, New York, NY, Avril 1988.

[65] R. G. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-29(6) :1153–1160, Décembre 1981.

[66] B.-J. Kim and W. A. Pearlman. An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (SPIHT). In *Proceedings of IEEE Data Compression Conference*, pages 251–260, Mars 1997.

[67] N. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Applied and Computational Harmonic Analysis*, 10 :234–253, Mai 2001.

[68] K. Kojima and H. Kiya. Generalizations of the image resolution conversions in DCT-domain using 8 points inversed DCT. In *Proc. of IEEE Asia-Pacific Conf. on Circuits and Systems*, Novembre 1998.

[69] J. Konrad. Transversal versus lifting approach to motion-compensated temporal discrete wavelet transform of image sequences : equivalence and tradeoffs. In *Proc. of SPIE Visual Communications and Image Processing*, volume 5308, pages 452–463, San Jose, CA, Janvier 2004.

[70] J. Kovačević and M. Vetterli. Perfect reconstruction filter banks with rational sampling factors. *IEEE Transactions on Signal Processing*, 41(6) :2047–2065, Juin 1993.

[71] F. Lazzaroni, R. Leonardi, and A. Signoroni. High-performance embedded morphological wavelet coding. *IEEE Signal Processing Letters*, 10(10), Octobre 2003.

[72] E. Le Pennec and S. G. Mallat. Sparse geometric image representation with bandelets. *IEEE Transactions on Image Processing*, 14(4), Avril 2005.

[73] R. Leonardi, N. Adami, A. Signoroni, and M. Brescianini. Fully embedded entropy coding with arbitrary multiple adaptation capabilities. Doc. M11378, Palma de Mallorca MPEG 70th meeting, Octobre 2004.

[74] R. Leonardi, N. Adami, A. Signoroni, and M. Brescianini. SVC CE1 : STool - a native spatially scalable approach to SVC. Doc. M11368, Palma de Mallorca MPEG 70th meeting, Octobre 2004.

[75] M. Li and T. Nguyen. Optimal wavelet filter design in scalable video coding. In *Proc. of the IEEE Int. Conf. on Image Processing*, Genova, Italy, Septembre 2005.

[76] Y. Lu and M. N. Do. Crisp-contourlets : a critically sampled directional multiresolution image representation. In *Proc. of SPIE Conf. on Wavelet Applications in Signal and Image Processing*, San Diego, CA, Août 2003.

[77] L. Luo, F. Wu, S. Li, Z. Xiong, and Z. Zhuang. Advanced motion threading for 3D wavelet video coding. *Signal Processing : Image Communication*, 19 :601–616, Août 2004. Special issue on Wavelet Video Coding.

[78] D. Maestroni, M. Tagliasacchi, and S. Tubaro. In-band adaptive update step based on local content activity. In *Proc. of SPIE Visual Communications and Image Processing*, Pékin, Chine, Juillet 2005.

[79] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, California, 1998.

[80] S. G. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$. *Trans. Amer. Math. Soc.*, 315 :69–87, Septembre 1989.

[81] H. Malvar. Lapped transforms software for image processing. Logiciels disponible sur le site *http ://research.microsoft.com/∼malvar/software/programs.aspx*, Janvier 2006.

[82] H. S. Malvar. Modulated QMF filter banks with perfect reconstruction. *Electronics Letters*, 26(13) :906–907, Juin 1990.

[83] H. S. Malvar. Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts. *IEEE Transactions on Signal Processing*, 46(4) :1043–1053, Avril 1998.

[84] H. S. Malvar. Fast progressive image coding without wavelets. In *Proceedings of IEEE Data Compression Conference*, pages 243–252, Snowbird, UT, Mars 2000.

[85] H. S. Malvar and D. H. Staelin. The LOT : transform coding without blocking effects. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-37 :553–559, Avril 1989.

[86] G. Marquant, E. Francois, N. Burdin, P. Lopez, and J. Viéron. Extended spatial scalability for non dyadic video formats : from SDTV to HDTV. In *Proc. of SPIE Visual Communications and Image Processing*, Beijing, China, Juillet 2005.

[87] B. Matei. Méthodes multirésolution non-linéaires. Applications à la compression des images. Thèse de doctorat de l'Université Paris-6, Novembre 2002.

[88] N. Mehrseresht and D. Taubman. Spatial scalability and compression efficiency within a flexible motion compensated 3D-DWT. In *Proc. of the IEEE Int. Conf. on Image Processing*, Singapour, Octobre 2004.

[89] Y. Meyer. *Ondelettes et Opérateurs, tome 1*. Hermann, Paris, 1990.

[90] T. Nakachi, T. Sawabe, J. Suzuki, and T. Fujii. A study on non-octave resolution conversion based on JPEG-2000 extensions. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Mai 2004.

[91] T. Nakachi, T. Sawabe, J. Suzuki, and T. Fujii. A study on non-octave scalable coding with filter bank and its performance evaluation using EBCOT. In *Intemational Symposium on Communications and Information Technologies (ISCIT'04)*, Sapporo, Japan, Octobre 2004.

[92] J.-R. Ohm. Three-dimensional subband coding with motion compensation. *IEEE Transactions on Image Processing*, 3(5) :559–589, Septembre 1994.

[93] C. Parisot, M. Antonini, and M. Barlaud. 3D scan based wavelet transform for video coding. In *Proc. of IEEE Workshop on Multimedia Signal Processing*, Cannes, France, Octobre 2001.

[94] C. Parisot, M. Antonini, and M. Barlaud. Motion-compensated scan based wavelet transform for video coding. In *Proc. of Tyrrhenian International Workshop on Digital Communications*, Capri, Italie, Septembre 2002.

[95] H. W. Park and H. S. Kim. Motion estimation using low-band-shift method for wavelet-based moving-picture coding. *IEEE Transactions on Image Processing*, 9(4) : 577–587, Avril 2000.

[96] Y. S. Park and H. W. Park. Arbitrary-ratio image resizing using fast DCT of composite length for DCT-based transcoder. *IEEE Transactions on Image Processing*, 15(2) : 494–500, Février 2006.

[97] G. Pau, S. Brangoulo, and Béatrice Pesquet-Popescu. Integration of bidirectional joint motion estimation for vidwav software. Doc. M13011, Bangkok MPEG 75th meeting, Janvier 2006.

[98] G. Pau and B. Pesquet-Popescu. Optimized prediction of uncovered areas in subband video coding. In *Proc. of the Picture Coding Symposium*, Décembre 2004.

[99] G. Pau and B. Pesquet-Popescu. Uniform motion-compensated 5/3 filterbank for subband video coding. In *Proc. of the IEEE Int. Conf. on Image Processing*, Singapour, Octobre 2004.

[100] G. Pau and B. Pesquet-Popescu. Comparison of spatial M-band filter banks for t+2D video coding. In *Proc. of SPIE Visual Communications and Image Processing*, Beijing, China, Juillet 2005.

[101] G. Pau and B. Pesquet-Popescu. Four-band linear-phase orthogonal spatial filter bank for subband video coding. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Philadelphie, PA, Mars 2005.

[102] G. Pau and B. Pesquet-Popescu. Image coding with rational spatial scalability. In *Proc. of EUSIPCO*, Florence, Italie, Septembre 2006.

[103] G. Pau, B. Pesquet-Popescu, and G. Piella. Modified M-band synthesis filter bank for fractional scalability of images. *IEEE Signal Processing Letters*, Juin 2006.

[104] G. Pau, B. Pesquet-Popescu, M. van der Schaar, and J. Viéron. Delay-performance trade-offs in motion-compensated scalable subband video compression. In *Proc. of Advanced Concepts for Intelligent Vision Systems (ACIVS)*, Septembre 2004.

[105] G. Pau, C. Tillier, and B. Pesquet-Popescu. Optimization of the predict operator in lifting-based motion compensated temporal filtering. In *Proc. of SPIE Visual Communications and Image Processing*, San Jose, CA, Janvier 2004.

[106] G. Pau, C. Tillier, B. Pesquet-Popescu, and H. Heijmans. Motion compensation and scalability in lifting-based video coding. *Signal Processing : Image Communication*, 19 :577–600, Août 2004. Special issue on Wavelet Video Coding.

[107] G. Pau, J. Viéron, and B. Pesquet-Popescu. Video coding with flexible MCTF structures for low end-to-end delay. In *Proc. of the IEEE Int. Conf. on Image Processing*, Gênes, Italie, Septembre 2005.

[108] B. Pesquet-Popescu and V. Bottreau. Three-dimensional lifting schemes for motion compensated video compression. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Salt Lake City, UT, Mai 2001.

[109] B. Pesquet-Popescu and F. Roueff. Les signaux à temps continu et leurs représentations. Support de cours, Département TSI de Télécom-Paris, 1999.

[110] G. Peyré and S. Mallat. Discrete bandelets with geometric orthogonal filters. In *Proc. of the IEEE Int. Conf. on Image Processing*, Gênes, Italie, Septembre 2005.

[111] G. Piella and H. J. A. M. Heijmans. Adaptive lifting schemes with perfect reconstruction. *IEEE Transactions on Signal Processing*, 50(7) :1620–1630, Juillet 2002.

[112] G. Piella, G. Pau, and B. Pesquet-Popescu. Adaptive lifting schemes combining seminorms for lossless image compression. In *Proc. of the IEEE Int. Conf. on Image Processing*, Gênes, Italie, Septembre 2005.

[113] G. Piella, B. Pesquet-Popescu, and H. Heijmans. Adaptive update lifting with a decision rule based on derivative filters. *IEEE Signal Processing Letters*, 9(10) :329–322, Octobre 2002.

[114] G. Piella, B. Pesquet-Popescu, H. Heijmans, and G. Pau. Combining seminorms in adaptive lifting schemes and applications to image analysis and compression. À paraître dans *Springer Journal of Mathematical Imaging and Vision*, 2006.

[115] G. Piella, B. Pesquet-Popescu, and H. J. A. M. Heijmans. Adaptive update lifting with a decision rule based on derivative filters. *IEEE Signal Processing Letters*, 9(10) : 329–332, Octobre 2002.

[116] W. K. Pratt. *Digital Image Processing*. Wiley, New York, 1978.

[117] J. Reichel and F. Ziliani. Controlled temporal haar transform for video coding. In *Proc. of the IEEE Int. Conf. on Image Processing*, Barcelone, Septembre 2003.

[118] I. H. Richardson. *H.264 and MPEG-4 Video Compression : Video Coding for Next-generation Multimedia*. Wiley, John Wiley & Sons Ltd., Chichester, UK, 2003.

[119] T. Rusert. Interframe wavelet video coding with operating point adaptation. In *Proc. of SPIE Visual Communications and Image Processing*, San Jose, CA, Janvier 2004.

[120] T. Rusert, K. Hanke, and C. Mayer. Enhanced interframe wavelet video coding considering the interrelation of spatio-temporal transform and motion compensation. *Signal Processing : Image Communication*, 19 :617–635, Août 2004.

[121] T. Rusert, K. Hanke, and M. Wien. Optimization for locally adaptive MCTF based on 5/3 lifting. In *Proc. of the Picture Coding Symposium*, Décembre 2004.

[122] A. Said and W. A. Pearlman. A new, fast and efficient image codec based on set partioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3) :243 – 250, Juin 1996.

[123] H. Schwarz, D. Marpe, and T. Wiegand. SNR-scalable extension of H.264/AVC. In *Proc. of the IEEE Int. Conf. on Image Processing*, Singapour, Octobre 2004.

[124] H. Schwarz, J. Shen, D. Marpe, and T. Wiegand. Technical description of the HHI proposal for SVC CE3. in doc. M11246, Palma MPEG meeting, Oct. 2004.

[125] A. Secker and D. Taubman. Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting. In *Proc. of the IEEE Int. Conf. on Image Processing*, pages 1029–1032, Thessalonique, Grèce, Octobre 2001.

[126] A. Secker and D. Taubman. Highly scalable video compression using a lifting-based 3D wavelet transform with deformable mesh motion compensation. In *Proc. of the IEEE Int. Conf. on Image Processing*, volume 3, pages 749–752, Rochester, New York, Septembre 2002.

[127] A. Secker and D. Taubman. Highly scalable video compression with scalable motion coding. *IEEE Transactions on Image Processing*, 13(8) :1029–1041, Août 2004.

[128] D. Seidner. Polyphase antialiasing in resampling of images. *IEEE Transactions on Image Processing*, 14(11) :1879–1889, Novembre 2005.

[129] S. D. Servetto, K. Ramchandran, and M. T. Orchard. Image coding based on a morphological representation of wavelet data. *IEEE Transactions on Image Processing*, 8 (9) :1161–1174, Septembre 1999.

[130] J. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12) :3445–3462, Décembre 1993.

[131] J. Solé and P. Salembier. Adaptive discrete generalized lifting for lossless compression. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Montréal, Canada, Mai 2004.

[132] A. Soman, P. P. Vaidyanathan, and T. Q. Nguyen. Linear phase paraunitary filter banks : Theory, factorizations, and applications. *IEEE Transactions on Signal Processing*, 41 :3480–3496, Décembre 1993.

[133] L. Song, J. Xu, H. Xiong, and F. Wu. Content adaptive update steps for lifting-based motion compensated temporal filtering. In *Proc. of the Picture Coding Symposium*, San Francisco, CA, Décembre 2004.

[134] J.-L. Starck, E. J. Candès, and D. L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6) :670–683, Juin 2002.

[135] P. Steffen, P. N. Heller, R. A. Gopinah, and C. S. Burrus. Theory of regular M-band wavelet bases. *IEEE Transactions on Signal Processing*, 41(12) :3497–3511, Décembre 1993.

[136] W. Sweldens. The lifting scheme : A new philosophy in biorthogonal wavelet constructions. In A. F. Lain and M. Unser, editors, *Wavelet Applications in Signal and Image Processing III*, volume 2569, pages 68–79. Proceedings of SPIE, 1995.

[137] D. Taubman and A. Zakhor. Multirate 3-D subband coding of video. *IEEE Transactions on Image Processing*, 3(5) :572–588, Septembre 1994.

[138] D. S. Taubman. High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9 :1158–1170, Juillet 2000.

[139] D. S. Taubman and M. W. Marcellin. *JPEG-2000 : Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002.

[140] ISO/IEC MPEG & ITU-T VCEG Joint Video Team. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264, ISO/IEC 14496-10 AVC), Doc. JVT-G050r1. Mai 2003.

[141] ISO/IEC MPEG & ITU-T VCEG Joint Video Team. Joint scalable video model JSVM-3, Doc. JVT-P202. Text of the Joint Scalable Video Model, Juillet 2005.

[142] C. Tillier. Scalabilité et robustesse dans le codage vidéo à base d'ondelettes. PhD thesis, GET-Télécom Paris, Juin 2005.

[143] C. Tillier and B. Pesquet-Popescu. 3D, 3-band, 3-tap temporal lifting for scalable video coding. In *Proc. of the IEEE Int. Conf. on Image Processing*, Barcelone, Espagne, Septembre 2003.

[144] C. Tillier, B. Pesquet-Popescu, and M. van der Schaar. Highly scalable video coding by bidirectional predict-update 3-band schemes. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Montréal, Mai 2004.

[145] C. Tillier, B. Pesquet-Popescu, and M. van der Schaar. Improved update operators for lifting-based motion-compensated temporal filtering. *IEEE Signal Processing Letters*, 12(2), Février 2005.

[146] C. Tillier, B. Pesquet-Popescu, Y. Zhang, and H. Heijmans. Scalable video compression with temporal lifting using 5/3 filters. In *Proc. of the Picture Coding Symposium*, pages 55–58, St. Malo, France, Avril 2003.

[147] T. D. Tran. The BinDCT : Fast multiplierless approximation of the DCT. *IEEE Signal Processing Letters*, 7(6), Juin 2000.

[148] W. Trappe and K. J. R. Liu. Adaptivity in the lifting scheme. In *33th Conference on Information Sciences and Systems*, pages 950–955, Baltimore, March 1999.

[149] M. Trocan, C. Tillier, and B. Pesquet-Popescu. Joint wavelet packets for group of frames in MCTF. In *Proc. of SPIE Optics and Photonics*, San Diego, CA, Juillet 2005.

[150] S. S. Tsai and H. M. Hang. Motion information scalability for MC-EZBC. *Signal Processing : Image Communication*, 19(7) :675–684, Août 2004.

[151] D. Turaga, M. van der Schaar, and B. Pesquet-Popescu. Complexity scalable motion compensated wavelet video encoding. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(8), Août 2003.

[152] D. S. Turaga, M. van der Schaar, and B. Pesquet-Popescu. Temporal prediction and differential coding of motion vectors in the MCTF framework. In *Proc. of the IEEE Int. Conf. on Image Processing*, Barcelone, Espagne, Septembre 2003.

[153] M. Unser and T. Blu. Mathematical properties of the JPEG2000 wavelet filters. *IEEE Transactions on Image Processing*, 12(9) :1080–1090, Septembre 2003.

[154] B. Usevitch. Optimal bit allocation for biorthogonal wavelet coding. In *Proc. of Data Compression Conference*, pages 387–395, Snowbird, UT, Mars 1996.

[155] P. P. Vaidyanathan. *Multirate systems and filter banks*. Prentice Hall, Englewood Cliffs, 1993.

[156] V. Valentin, M. Cagnazzo, M. Antonini, and M. Barlaud. Scalable context-based motion vector coding for video compression. In *Proc. of the Picture Coding Symposium*, pages 63–70, St. Malo, France, Avril 2003.

[157] M. van der Schaar and D. S. Turaga. Unconstrained motion compensated temporal filtering (UMCTF) framework for wavelet video coding. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Hong Kong, Chine, Avril 2003.

[158] J. Viéron, G. Boisson, E. Francois, G. Pau, and B. Pesquet-Popescu. Proposal for SVC CE1 : Time and level adaptive MCTF architectures for low delay video coding. Doc. M11673, Hong Kong MPEG meeting, Janvier 2005.

[159] J. D. Villasenor, B. Belzer, and J. Liao. Wavelet filter evaluation for image compression. *IEEE Transactions on Image Processing*, 4(8) :1053–1060, Août 1995.

[160] G. Wolberg. *Digital Image Warping*. IEEE Computer Society Press, Los Alamitos, CA, 1990.

[161] F. Wu, S. Li, and Y.-Q. Zhang. A framework for efficient progressive fine granularity scalable video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(3), Mars 2001.

[162] S. W. Wu and A. Gersho. Joint estimation of forward and backward motion vectors for interpolative prediction of video. *IEEE Transactions on Image Processing*, 3(5) : 684–687, Septembre 1994.

[163] R. Xiong, F. Wu, S. Li, Z. Xiong, and Y.-Q. Zhang. Exploiting temporal correlation with adaptive block-size motion alignment for 3D wavelet coding. In *Proc. of SPIE Visual Communications and Image Processing*, San Jose, CA, Janvier 2004.

[164] R. Xiong, F. Wu, J. Xu, S. Li, and Y.-Q. Zhang. Barbell lifting wavelet transform for highly scalable video coding. In *Proc. of the Picture Coding Symposium*, San Francisco, CA, Décembre 2004.

[165] J. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang. 3D embedded subband coding with optimal truncation (3D-ESCOT). *Applied and Computational Harmonic Analysis*, 10 :589, Mai 2001.

Index

Algorithme
    d'Alkin et Caglar, 130, 147
    itératif, 80
Analyse multirésolution, 18, 27
Approximation multirésolution, 18
Arbre dual, 39
Artefacts fantômes, 93

Banc de filtres, 20
    M-bandes, 122
Banc de synthèse, 140
Bandelettes, 37
Base
    d'ondelettes, 19, 22
    séparable, 22
Biorthogonalité, 24
Boucle fermée, 41, 56

CABAC, 44
Carte de décision, 156
Chrominance, 52
Codage
    2D+t, 57
    hybride, 41
    inband, 57
    robuste, 54
    t+2D, 46
Connectivité, 66, 95, 101
Contourlets, 36
Critère de seuil TC, 157
Curvelets, 36

DCT, 42, 124, 147
DFD, 128
DivX, 43
Dual-tree, 39
Décimation, 20
Délai, 105
    d'encodage, 105
    de décodage, 106
    de reconstruction, 106
    point à point, 106

EBCOT, 29
Elagage d'arbre, 87
EMDC, 29
Equations à deux échelles, 20
ESCOT, 57
Estimation de mouvement, 42
    indépendante, 79
    jointe, 80
EZBC, 29, 54, 179
EZW, 28

FGS, 44
Filtre
    compensé en mouvement, 48
    d'analyse, 21
    de rééchantillonnage, 140
    de synthèse, 22
    FB1, 132, 137
    spatial, 155
    temporel, 46
    temporel 5/3, 61, 69
    temporel de Haar, 51
Fonction d'échelle, 18
Footprints, 39

GOP, 53
Group of Pictures, 53

H.264/AVC, 44, 45, 89, 124
HDTV, 139
HVSBM, 52, 79

Inversibilité, 32, 51, 65
Isotropie, 35

JPEG-2000, 29
JSVM, 45
JVT, 90

KLT, 124

Latence, 105
LBT, 125, 137, 147
Lifting, 30
    adaptatif, 155
    temporel, 51, 62
LOT, 125, 137, 147

MC-EZBC, 51
MCTF, 46, 47
Mise à jour, 31
    adaptative, 67, 156
MLBT, 126, 137
MLT, 126, 136
Modes, 56, 65
Moment nul, 22
MPEG-1, 43
MPEG-2, 43, 74
MPEG-4
    Part. 10, 44
    Part. 2, 43, 73

Ondelette
    biorthogonale 5/3, 25, 33, 34
    biorthogonale 7/5, 114
    biorthogonale 9/7, 26, 129, 136
    CDF, 25
    complexe, 39
    Daubechies-4, 25, 34, 114
    de Haar, 24, 34
    géométrique, 30, 35
    M-bandes, 123
    mère, 19
    orientée, 38
    orthogonale, 19
    père, 18
    séparable, 22, 26, 35
Optimisation
    de la latence, 104
    de la prédiction, 78, 98
Opérateur
    de compensation de mouvement, 69
    de mise à jour, 31, 65
    de prédiction, 31, 63
Orthogonalité, 20, 24

Polyphase, 33
Prédiction, 31
    des zones découvertes, 98
PSNR, 74

PTC, 137

Quantification, 42

Reconstruction, 22
    parfaite, 23
Redimensionnement, 140
Retard, 105
Ridgelets, 35
RLE, 42

SAD, 79, 84
Scalabilité, 15, 47, 54
    fractionnaire, 138
    rationnelle, 149
    temporelle, 75
Seminorme, 157
    pondérée, 161
Simulcast, 17
Sous-bande, 21
    d'approximation, 61, 128
    de détail, 61, 128
    temporelle, 53, 128
Sous-échantillonnage, 20
SPIHT, 29
Structure lifting, 30
Sur-échantillonnage, 21
SVC, 45, 90

Traitement au fil de l'eau, 70
Transformée
    (P,Q), 110
    5/3 uniforme, 92
    en blocs, 41, 123
    en cosinus discrète, 124
    en ondelettes, 19
    en ondelettes entières, 35
    en ondelettes M-bandes, 122
    en ondelettes rapide, 20
    temporelle D4, 114
    à recouvrement, 125

UMCTF, 56, 114

Vidwav, 57, 89, 135
Vidéoconférence, 77
VLC, 43

Windows Media 9, 73