Codage video scalable par maillages et ondelettes t+2D

Nathalie Cammas

N° d'ordre: 3063

THÈSE

présentée

DEVANT L'UNIVERSITE DE RENNES 1

pour obtenir

le grade de : DOCTEUR DE L'UNIVERSITE DE RENNES 1
Mention INFORMATIQUE

par

Nathalie CAMMAS

Équipe d'accueil : France Télécom R&D/TECH/IRIS
École doctorale : MATISSE
Composante universitaire : TEMICS/IRISA

Titre de la thèse :
Codage vidéo scalable par maillages et ondelettes t+2D

soutenue le 5 novembre 2004 devant la commission d'examen

M. :	Kadi	BOUATOUCH	Président
MM. :	Michel	BARLAUD	Rapporteurs
	Jean-Marc	CHASSERY	
MM. :	Christine	GUILLEMOT	Examinateurs
	David	TAUBMAN	
	Stéphane	PATEUX	

Remerciements

Ce travail a été réalisé dans le cadre d'un contrat Cifre en collaboration avec l'IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires) et le laboratoire TECH/IRIS de France Télécom R&D.
Je tiens à remercier tout particulièrement Christine Guillemot, Directrice de recherche à l'INRIA et responsable du projet Temics de m'avoir accueillie dans ce projet, de m'avoir permis de réaliser ces travaux de thèse et d'avoir accepté de juger ce travail. Je remercie également Henri Sanson et Vincent Marcatté pour m'avoir accueillie au sein de leur laboratoire à France Télécom R&D et de m'avoir fait confiance pour mener à bien cette thèse.

Je remercie M. Kadi Bouatouch, professeur à l'Université de Rennes 1 et repsonsable du projet SIAMES, qui m'a fait l'honneur de présider le jury de cette thèse.
Je remercie M. Jean-Marc Chassery, Directeur de Recherche CNRS, LIS-ENSIEG, et M. Michel Barlaud, Professeur à l'ESSI et responsable de l'équipe CReATIVe au laboratoire I3S, d'avoir bien voulu accepter la charge de rapporteur.
Je remercie M. David Taubman, professeur à l'Université de New South Wales, d'avoir accepté de juger ce travail.

Je remercie tout particulièrement Stéphane Pateux, Ingénieur de Recherche et Développement à France Télécom R&D, pour m'avoir encadrée et dirigée tout au long de cette thèse, pour sa disponibilité et la confiance qu'il m'a accordée.
Je remercie Nathalie Laurent, Ingénieur de Recherche et Développement à France Télécom R&D pour m'avoir encadrée durant cette thèse.
Je tiens également à remercier Luce Morin, Maître de Conférence à l'Université de Rennes 1, pour tous les conseils qu'elle a pu me donner et pour les discussions que nous avons eues.

Enfin, je remercie tous les membres du projet Temics à l'IRISA et du laboratoire IRIS à France Télécom avec qui j'ai eu le plaisir de travailler et de discuter durant cette thèse.

Table des matières

Table des figures

Glossaire

CABAC	*Context-based Adaptive Binary Arithmetic Coding* Codage arithmétique avec utilisation d'un contexte adaptatif
CIF	*Common Intermediate Format* Format d'images pour la vidéo 352x288
DCT	*Discrete Cosine Transform*
DFD	*Displaced Frame Difference*
DPCM	*Differential Pulse Coding Modulation* Mode de codage différentiel
DWT	*Discrete Wavelet Transform*
EBCOT	*Embedded Block Coding with Optimized Truncation* Algorithme de compression d'images fixes (utilisé dans JPEG-2000)
EQM	*Erreur Quadratique Moyenne*
EZBC	*Embedded ZeroBlocks coding based on Context modeling* Algorithme de compression d'images fixes
EZW	*Embedded Zerotree of Wavelet* Algorithme de compression d'images fixes
FGS	*Fine Granular Scalability*
GOF	*Group Of Frames*
H.26x	H.261, H.262, H.263x, H.26L sont des normes de compression vidéo numérique bas débit issues de l'ITU. H.264 est issue du groupe de travail commun ISO et ITU et prolonge le norme H264L.
ISO	*Organisation Internationale de normalisation*
ITU	*International Telecommunications Union*
INTRA (I)	Mode de codage sans utilisation de prédiction temporelle
INTER (P,B)	Mode de codage avec utilisation de prédiction(s) temporelle(s) Le mode INTER regroupe le codage par Prediction simple (P) et par prédiction Birectionnelle (B)
JPEG	*Joint Picture Expert Group* JPEG et JPEG-2000 sont des normes de compression d'images fixes
kb/s	kilo-bits par secondes
MPEG	*Motion Picture Expert Group*
PSNR	*Peak Signal to Noise Ratio*
SA	Shape-Adapted, codage basé forme
SPIHT	*Set Partitioning In Hierarchical Tree* Algorithme de compression d'images fixes

Introduction

Aujourd'hui, la vidéo numérique permet d'offrir aux utilisateurs un grand nombre de services tels que vidéo sur réseaux, DVD, broadcasting, vidéo à la demande, etc... Ces applications ont nécessité le développement d'outils de codage efficaces et rapides afin de compresser au maximum le flux vidéo tout en gardant une qualité visuelle optimale. Face à cette offre de services, tous les utilisateurs ne sont pas égaux en terme de ressources disponibles pour pouvoir visualiser ces flux vidéo (types de réseaux de transmission, terminaux de visualisation, ...). Les fournisseurs de services doivent donc faire face à cette hétérogénéité et nécessitent de nouveaux outils de codage vidéo toujours aussi efficaces en terme de compression mais permettant également de résoudre ce problème d'hétérogénéité. Pour cela, la notion de scalabilité et de codage vidéo scalable apparaît. Le mot scalabilité en français est la traduction de l'anglais scalability qui signifie le fait d'être échelonnable. Bien que ce mot scalabilité n'existe pas (encore) dans le vocabulaire français, nous nous permettons de l'utiliser dans ce manuscrit et de lui donner le même sens qu'en langue anglaise.
Cette étude s'inscrit dans le cadre du codage vidéo scalable. Les deux objectifs à atteindre dans ce domaine sont l'efficacité en compression et la scalabilité. Or, considérant les standards de compression vidéo actuels, il est très difficile de pouvoir atteindre ces deux objectifs simultanément. Le codeur H264/AVC offre de très bonnes performances en terme de compression mais le flux compressé fourni n'est pas scalable. Le codeur MPEG-4 FGS offre une scalabilité fine du flux compressé mais ses performances en compression chutent beaucoup par rapport à une version du codeur non scalable.
Nous voyons alors apparaître le besoin de définir de nouveaux standards de compression vidéo scalable offrant d'une part des performances en compression similaires à celles d'un codeur état de l'art non scalable et d'autre part permettant l'adaptation des caractéristiques du flux aux utilisateurs via la scalabilité.
Dans ce contexte, les ondelettes semblent être un outil intéressant tant au niveau de la scalabilité qu'au niveau de la compression. En effet, elles ont prouvé leur efficacité en terme de compression en codage d'images fixes. Elles offrent également une représentation hiérarchique naturelle de l'information qui peut être aisément exploitable pour obtenir un flux scalable.
L'adaptation des ondelettes à la vidéo effectue une décomposition de l'information le long de l'axe temporel et des axes spatiaux. Pour être efficace, la décorrélation le long de l'axe temporel nécessite d'exploiter le mouvement dans la décomposition, comme cela est fait dans les schémas de codage vidéo classique par prédiction.

Dans les schémas par transformée ondelettes, l'exploitation du mouvement dans la transformée temporelle est une étape cruciale dont dépendent les performances de décorrélation de la transformée, ainsi que les performances en compression du schéma de codage. La première partie de ce manuscrit porte sur l'étude de l'exploitation du mouvement dans un schéma de codage vidéo mettant en œuvre une transformée en ondelettes temporelles.
Au niveau de la représentation du mouvement, les maillages 2D offrent des propriétés intéressantes. Ils permettent un suivi long terme de la texture et fournissent une continuité de la texture bien adaptée pour une transformée en ondelettes temporelles.
Par rapport à une représentation par blocs, les maillage permettent de représenter de manière continue le champ de mouvement. Cependant, le mouvement réel d'une scène n'est pas totalement continue. Des discontinuités apparaissent dans les zones d'occlusions (découvrements, recouvrements). Dans ces zones, l'estimation du mouvement échoue provoquant une mauvaise décorrélation de l'information de texture et donc une perte d'efficacité en terme de compression.

Le manuscrit est organisé de la manière suivante:

- Le premier chapitre définit la notion de scalabilité et montre la nécessité de définir de nouveaux standards de codage vidéo scalable performants en compression.
- Le deuxième chapitre montre un état de l'art des techniques de codage vidéo scalable par ondelettes spatio-temporelles. Il présente leurs avantages et inconvénients et montre l'importance de la bonne exploitation du mouvement dans la transformée temporelle.
- Le schéma de codage vidéo que nous proposons est présenté dans le troisième chapitre. Nous présentons les choix que nous avons faits au niveau de chaque brique du schéma de codage et comment la scalabilité est mise en œuvre au sein du schéma.
- Le chapitre 4 présente les résultats de codage obtenus par le schéma proposé. Il valide les choix faits dans les différentes briques du schéma de codage. Il montre les performances en compresssion par une comparaison avec un codeur état de l'art du codage vidéo non scalable H264/AVC, ainsi que les performances en terme de scalabilité par une comparaison avec plusieurs codeurs vidéo scalables proposés en normalisation MPEG lors du CFP [cfp 03].
- Le chapitre 5 présente un état de l'art des techniques de gestion des zones d'occlusions dans la représentation du mouvement par maillages.
- Le chapitre 6 propose une nouvelle structure de maillages prenant en compte les discontinuités du mouvement.
- Enfin, le chapitre 7 présente les résultats obtenus avec la nouvelle structure de maillages proposée, ainsi que des perspectives relatives à cette étude.

Chapitre 1

Codage vidéo et scalabilité

Ce premier chapitre définit la notion de scalabilité et présente quelques-unes de ses applications. Les standards de codage vidéo actuels sont ensuite étudiés ainsi que la mise en œuvre de la scalabilité au sein de ces schémas. Enfin, nous positionnons l'approche proposée dans ce manuscrit par rapport aux codeurs actuels.

1.1 La scalabilité et les applications visées en vidéo

1.1.1 Définition de la scalabilité

La scalabilité est la possibilité de pouvoir représenter un signal à différents niveaux d'information. Le signal est codé dans un seul flux binaire de manière à offrir la possibilité de décoder un sous-flux de base et des sous-flux englobant successifs permettant de raffiner successivement les sous-flux emboités. La figure 1.1 illustre ce principe. L'information à coder est hiérarchisée de manière à concentrer l'information pertinente dans le flux de base, la pertinence des informations contenues dans les sous-flux décroît avec l'augmentation du nombre de niveaux. Nous allons maintenant étudier les différents types de scalabilité.

La scalabilité en qualité
La scalabilité en qualité consiste à diminuer la distorsion de quantification entre le signal original et le signal reconstruit. A chaque niveau de hiérarchie est associé une qualité de reconstruction. La couche de base permet de reconstruire le signal avec une qualité minimale, l'ajout d'information supplémentaire permet d'améliorer cette qualité.

La scalabilité temporelle
La scalabilité temporelle consiste à diminuer ou à augmenter la fréquence temporelle du signal. En vidéo, cela consiste à ajouter des images intermédiaires entre les images reconstruites pour le niveau de base.

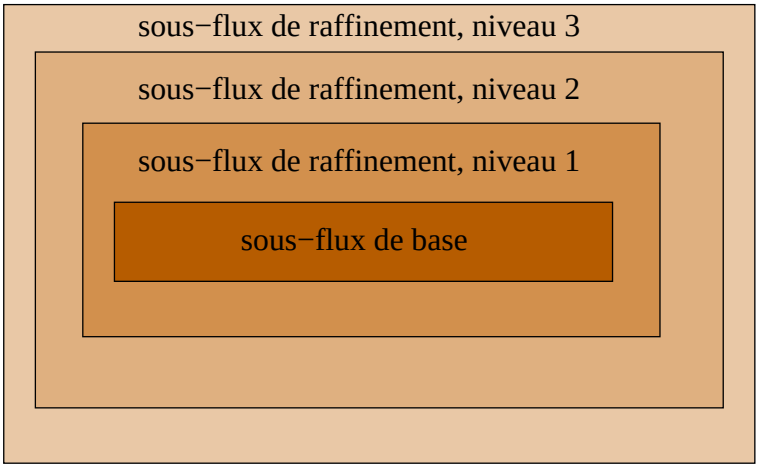


FIG. 1.1 – *Emboîtement des sous-flux pour un codage scalable*

La scalabilité spatiale
La scalabilité spatiale consiste à proposer plusieurs niveaux de résolution spatiale des images. La résolution de décodage est dépendante de la résolution du terminal utilisé pour la visualisation, par exemple pour une visualisation sur mobile ou sur un téléviseur.

La scalabilité en complexité
La scalabilité en complexité consiste à adapter la complexité du processus de décodage du signal compressé en fonction des capacités de la machine effectuant ce décodage.

La figure 1.2 illustre les principales scalabilités appliquées à des signaux image et vidéo.
La scalabilité permet d'adapter les caractéristiques du signal aux capacités des réseaux de transmission et des terminaux de visualisation de la vidéo, mais elle offre également une robustesse aux erreurs de transmission. La couche de base contient l'information la plus importante, cette couche est à transmettre en priorité et sur un canal fiable. Elle permet d'avoir une reconstruction minimale du signal. Les couches de raffinement, moins importantes, peuvent être transmises sur un canal plus sujet aux erreurs.

1.1.2 Le contexte de normalisation MPEG

Le groupe de travail MPEG, créé en 1988 au sein du groupe ISO, a pour but de développer des standards de codage de documents numériques vidéo et audio. La première norme éditée par le groupe est le standard MPEG-1. MPEG-1 permet le codage de séquences d'images et audio pour le stockage numérique à des débits allant jusqu'à 1,5Mbits/s. Ensuite, la norme MPEG-2 a permis de définir un standard de codage vidéo et audio pour la distribution de télévision numérique et le stockage de vidéo et audio sur DVD. Avec la norme MPEG-4, le champ d'action du groupe MPEG

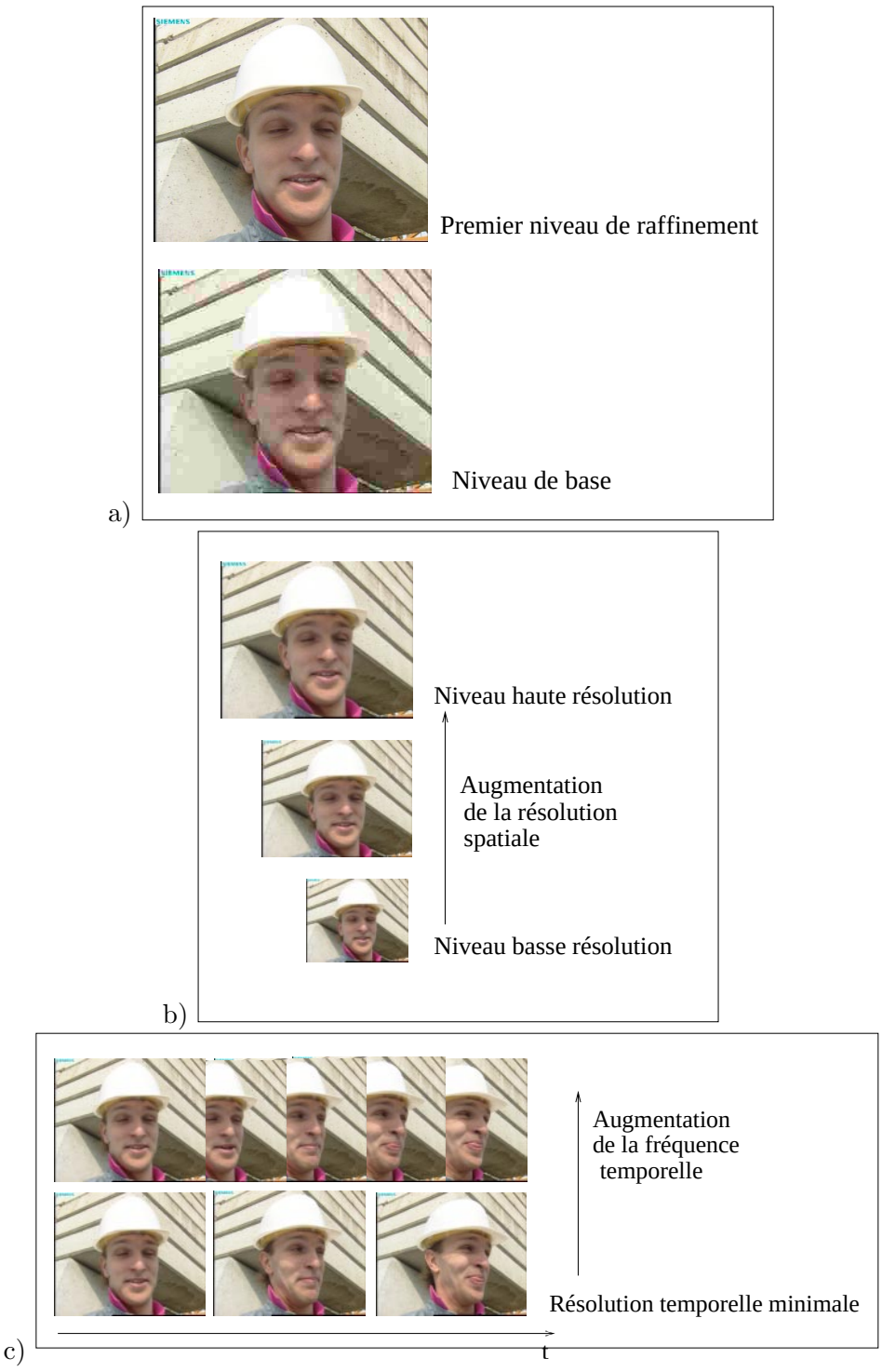


FIG. 1.2 – *Différentes scalabilités: a) qualité, b) spatiale, c) temporelle*

s'étend à tout document numérique: vidéo, audio, texte, objets synthétiques, ... Le nouveau standard propose des technologies permettant la production, la distribution et la manipulation de documents audiovisuels sur des réseaux fixes ou mobiles. Avec ce standard, on commence à voir apparaître un besoin d'homogénéisation de la représentation des contenus par rapport à l'hétérogénéité des applications, des ressources des utilisateurs, des capacités des réseaux et des types de réseaux, ainsi qu'un besoin de protection des contenus audiovisuels distribués. Aujourd'hui, le groupe MPEG s'intéresse à définir de nouveaux standards adressant les applications émergentes telles que l'indexation de documents audiovisuels (MPEG-7) ou l'uniformisation du schéma multimédia (MPEG-21).
La norme MPEG-4 vise un grand nombre d'applications mettant en jeu des capacités et des ressources très diverses. Elle vise la transmission des données multimédia sur des réseaux mobiles ou fixes qui ont des bandes passantes variables, des qualités de service différentes et qui sont soumis aux erreurs de transmission. De plus, elle vise des utilisateurs ayant des ressources variées au niveau des terminaux et des réseaux de connexion. Les terminaux diffèrent par les résolutions spatiales et temporelles d'affichage des documents, les capacités de décodage et d'affichage. Face à cette diversité, la norme MPEG-4 introduit la notion de scalabilité en fournissant des éléments technologiques permettant d'assurer cette fonctionnalité. Une scalabilité en couches est proposée, une couche de base au niveau qualité, fréquence temporelle et résolution spatiale est d'abord codée, puis des couches de raffinement successives sont codées en différentiel par rapport à la couche de base. Selon les caractéristiques des utilisateurs, plus ou moins de couches de raffinement sont envoyées. La scalabilité à grain fin est aussi proposée dans le codeur MPEG-4 FGS (Fine Grain Scalability). Un flux de base est toujours généré mais une seule couche de réhaussement est ensuite codée de manière progressive, cette couche est ensuite tronquée en fonction des caractéristiques de l'utilisateur à qui elle est envoyée.
En parallèle de la norme MPEG-4, des études sur le codage scalable ont été menées au sein du groupe MPEG. Ces études ont portées sur les applications visées et les conditions requises pour un codage scalable, divers schémas de codage scalable ont été proposés. En décembre 2003, ces études ont conduit à un appel à propositions de codeurs vidéo scalables.
Les conditions requises et les applications visées sont présentées dans le paragraphe suivant.

1.1.3 Conditions requises et applications pour le codage scalable

Le rapport [N6025 03] présente les conditions requises pour un codeur scalable ainsi que des exemples d'applications visées.
Un codeur permettant la scalabilité doit avant tout offrir des performances en terme de compression au moins aussi bonnes qu'un codeur non scalable issu de l'état de l'art. La référence en terme de codeur non scalable aujourd'hui est le codeur issu du JVT (Joint Video Team): H264 (ITU) ou MPEG-4 part 10 AVC (Advanced Video Coding). Le codeur doit pouvoir proposer toutes les scalabilités présentées au début de ce chapitre et doit pouvoir les combiner. Une application pour un codeur scalable est la produc-

tion et la distribution de contenus vidéo. Un utilisateur peut choisir de visualiser un contenu vidéo sur son mobile, son ordinateur ou sur une télévision haute définition, le contenu doit alors pouvoir s'adapter aux caractéristiques du terminal d'affichage: résolution spatiale, fréquence d'affichage, qualité visuelle optimale. Le flux vidéo doit aussi pouvoir s'adapter aisément au réseau de transmission (IP, mobile, sans fil, ...) et à ses fluctuations.
Le codeur scalable doit offrir une robustesse aux erreurs de transmission intervenant sur le réseau et une adaptation rapide aux fluctuations de la bande passante. Par exemple, la transmission sur des réseaux sans fil est sujette à d'importantes fluctuations dues notamment aux interférences, au recouvrement par un autre réseau sans fil, un trafic important, la mobilité des utilisateurs, etc.
Le codeur scalable doit pouvoir traiter les régions d'intérêts, comme dans la norme MPEG-4. Un exemple d'application est le commerce via les mobiles. Aujourd'hui, un publicitaire qui veut diffuser un contenu vidéo sur une télévision et un mobile doit créer deux versions de son contenu. Un codeur scalable permettrait de n'avoir qu'un contenu à diffuser, le contenu est diffusé en pleine résolution et haute qualité sur la télévision, tandis que l'accent est mis sur une région d'intérêt pour la visualisation sur mobile. De même avec les régions d'intérêt interactives, un utilisateur peut choisir de voir une certaine région avec une meilleure qualité que le reste du contenu. Cette fonctionnalité est demandée notamment dans des systèmes de surveillance.
Le délai de codage et de décodage du contenu est important. On comprend l'importance de ce délai dans des applications de surveillance où l'utilisateur doit pouvoir réagir dans des délais très courts par rapport à ce qu'il visualise. Cependant, le délai de codage et de décodage autorisé dépend fortement des applications visées.
Enfin, les technologies de codage scalable doivent fournir un moyen de manipuler et d' adapter aisément les flux scalables.

1.2 Les standards de codage vidéo scalable

Cette section présente tout d'abord le principe du codage vidéo utilisé dans les standards de codage vidéo actuels, puis nous étudierons la mise en œuvre de la fonctionnalité de scalabilité dans ce type de schéma. Enfin, nous verrons les limitations inhérentes à un codage prédictif auxquelles se heurtent ces codeurs et les nouvelles approches envisagées pour y faire face.

1.2.1 Le principe du codage prédictif

Le principe du codage vidéo se base sur l'élimination de la redondance temporelle existant entre des images successives d'une séquence vidéo. La figure 1.3 présente un schéma de codage vidéo de type MPEG. Le principe du codeur est de coder une première image en mode Intra, comme une image fixe seule (image I), puis d'effectuer une prédiction de l'image suivante par rapport à la première image. Pour cela, une estimation du mouvement est d'abord effectuée entre les deux images, puis la première image est compensée en mouvement par rapport à la deuxième et l'erreur de

prédiction entre l'image compensée prédite et la deuxième image est calculée et codée. La deuxième image est codée en mode Inter (image P). Une extension de ce principe est la prédiction bidirectionnelle des images (images B), une image est alors prédite par deux images références, les images références sont des images I ou P. Un GOP (Group of Object Planes) est composé d'images I, P et B, en commençant toujours par une image I. Les images I sont décodables indépendamment des autres images et permettent des fonctions d'accès aléatoires dans le flux et de robustesse. La figure 1.4 montre un exemple de structure de GOP.
Pour le codage, les images sont segmentées en macro-blocs de taille 16x16, chaque macro-bloc est divisé en quatre blocs 8x8. Le codage des images I et de l'erreur de prédiction des images P ou B est fait à l'aide d'une transformée en cosinus discrète (DCT) appliquée sur les blocs 8x8 de l'image. La DCT permet de décorréler spatialement les informations à coder en générant des coefficients plus ou moins pertinents. Ces coefficients sont ensuite quantifiés et codés via un codeur entropique. La perte irréversible d'informations au niveau du codage est dans l'étape de quantification, toutes les autres étapes sont réversibles.
Le codage prédictif doit fonctionner en boucle fermée afin d'éviter les effets de dérives dues aux erreurs de quantification intervenant sur les images de référence. Le schéma fonctionne en boucle fermée si le calcul de l'erreur de prédiction se fait entre l'image de référence codée-décodée compensée en mouvement et l'image originale à prédire.
Le mouvement permettant d'effectuer la compensation doit aussi être codé. Ce mouvement est représenté par blocs qui peuvent être de taille variable ou fixe selon les codeurs. A chaque bloc ou macro-bloc est associé un vecteur mouvement. Les vecteurs peuvent être codés par prédiction comme les images, l'erreur est ensuite quantifiée et codée par un codage entropique.
D'un codeur vidéo simple (type MPEG-2, MPEG-4), diverses améliorations ont été apportées permettant d'augmenter l'efficacité en compression. Ces améliorations portent entre autres sur l'amélioration de la prédiction des images et sur l'utilisation de techniques de codage plus efficaces. Les images peuvent être prédites par de multiples références pouvant aller jusqu'à cinq images références, la taille des blocs de segmentation de l'image devient variable, les macro-blocs peuvent être divisés en blocs de taille 8x8, 4x4, 8x4, ... s'adaptant ainsi mieux au contenu des images. La prédiction des blocs des images Intra est aussi améliorée grâce à l'augmentation du nombre de mode de prédiction de coefficients. De plus, la précision sous-pixellique du mouvement augmente, il est possible d'avoir des précisions au quart ou au huitième de pixels. Pour limiter l'effet de blocs sur les images reconstruites et sur les images utilisées pour la prédiction, une boucle de filtrage adaptatif (Deblocking Filter) est insérée dans la boucle de rétroaction du codage. Le codage entropique par VLC (Variable Length Coding) est remplacé par un codage adaptatif arithmétique contextuel (CABAC:Context Adaptative Binary Arithmetic Coding). Une étape d'optimisation débit/distorsion permet de coder la séquence vidéo en utilisant les modes de codage optimaux. Toutes ces améliorations ont permis d'augmenter l'efficacité en compression par rapport aux codeurs standards actuels mais au détriment de la complexité opératoire. Elles sont présentes dans le codeur vidéo H264 proposé par l'ITU qui représente aujourd'hui l'état

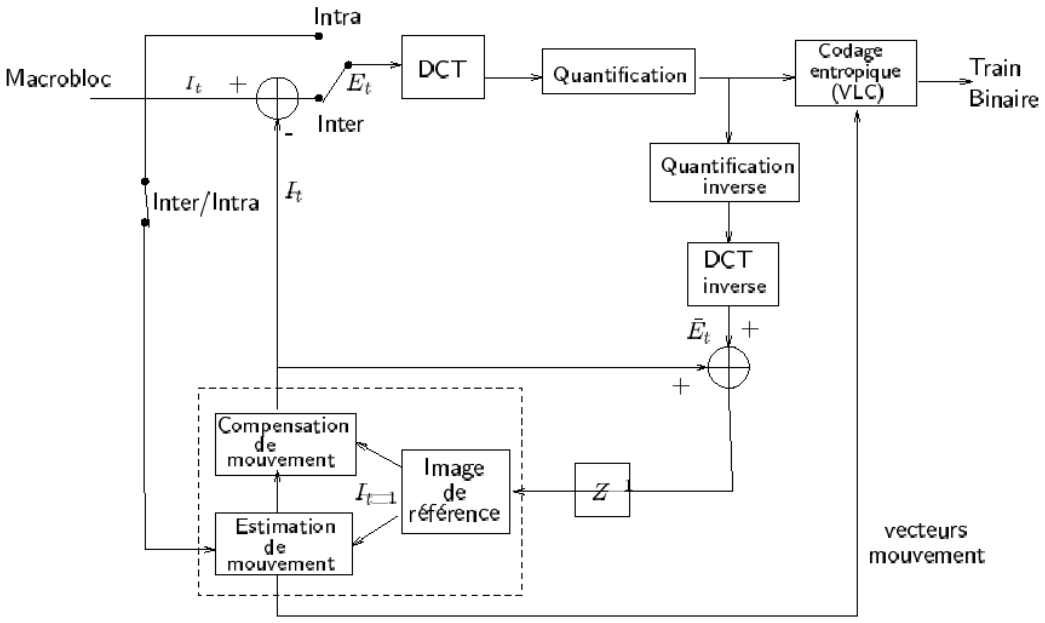


FIG. 1.3 – *Schéma de principe d'un codeur vidéo prédictif*

de l'art en terme de codage vidéo.

1.2.2 Mise en œuvre de la scalabilité au sein d'un codeur standardisé

La mise en œuvre de la scalabilité dans un codeur du type décrit précédent est assez délicate dans le sens où c'est une fonctionnalité qui est ajoutée au schéma du codeur et qui n'a pas été pensée comme une propriété inhérente de ce codeur.
La scalabilité temporelle est facilement ajoutée par l'introduction ou la suppression d'images B et/ou P dans la séquence vidéo, figure 1.5. Par exemple, la fonctionnalité d'avance rapide sur un DVD est obtenue par le décodage des images I uniquement.
La scalabilité spatiale est obtenue par un codage en couche. Une couche de base code d'abord la séquence vidéo à une résolution basse, puis des couches de raffinement successives codent les informations nécessaires pour le décodage à des résolutions supérieures. Ces couches de raffinement sont codées par rapport aux couches inférieures. Elles peuvent être codées soit par prédiction inter-image par les images du niveau de résolution de la couche de raffinement, soit par prédiction spatiale par les images de résolution inférieure, figure 1.6.
La scalabilité en qualité est obtenue en codant une couche de base avec une qualité minimale, puis des couches de raffinement successives améliorant la qualité. Les couches de raffinement peuvent être codées par prédiction Intra de l'image à affiner ou par prédiction inter-images entre les images de la couche de raffinement. Le procédé est

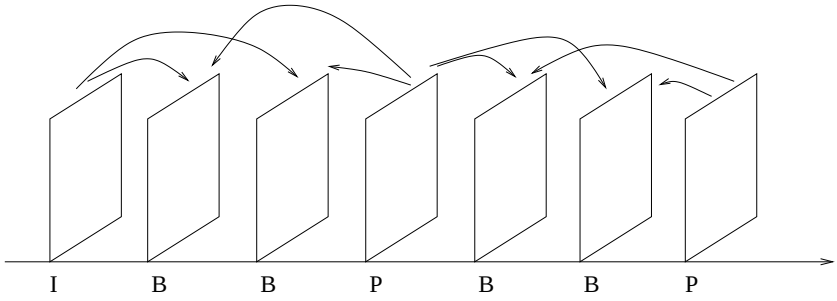


FIG. 1.4 – *Exemple de structure d'un GOP*

semblable à la scalabilité spatiale.
L'inconvénient du codage par couches de raffinement successives est de ne pas rendre le flux très souple en terme d'adaptation de la qualité. Le nombre de couches de raffinement est très limité. Autant au niveau de la scalabilité spatiale, le nombre de couches de raffinement n'est pas très important, un nombre de 3 voire 5 est suffisant pour la plupart des applications, autant en scalabilité SNR, il serait intéressant d'avoir une scalabilité fine, qui permettrait d'adapter le flux finement par rapport aux variations de la bande passante par exemple. Cette scalabilité fine a été mise en œuvre dans le standard MPEG-4 dans la partie MPEG-4 FGS (Fine Granularity Scalability). Deux couches seulement sont codées: une couche de base typique d'un codeur MPEG et une couche de raffinement codant l'erreur de quantification de la couche de base. La couche de raffinement est codée de manière progressive en utilisant un codage par plans de bits. Selon que l'on décode plus ou moins de plans de bits du flux de la couche de raffinement, la qualité de la couche de base est plus ou moins améliorée. La figure 1.7 illustre ce principe.

1.2.3 Limitations de ces codeurs

Nous venons de voir que la scalabilité dans un codeur prédictif est obtenue par une structuration du codage en une couche de base à laquelle s'ajoutent une ou plusieurs couches de réhaussement. Cependant, cette scalabilité ne permet pas une adaptation très fine du flux. On remarque aussi que la scalabilité dans ce type de schémas correspond à l'ajout d'une fonctionnalité dans un codeur initialement non scalable et non à une caractéristique propre du codeur.
De plus, si l'on s'intéresse aux résultats en terme de compression de ces codeurs, on note que les performances chutent de manière très significatives par rapport à leur version non scalable. Dans [Li 01], l'auteur compare les performances du codeur vidéo MPEG-4 non scalable et le codeur MPEG-4 FGS, les résultats montrent une perte allant jusqu'à de 2dB en défaveur du codeur scalable.
De même, l'état de l'art en terme de codage vidéo est aujourd'hui le codeur MPEG-4 AVC ou H264. Nous comparons les résultats obtenus par ce codeur et ceux obtenus par le codeur scalable MPEG-4 FGS avec les courbes de la figure 1.8. Les courbes montrent

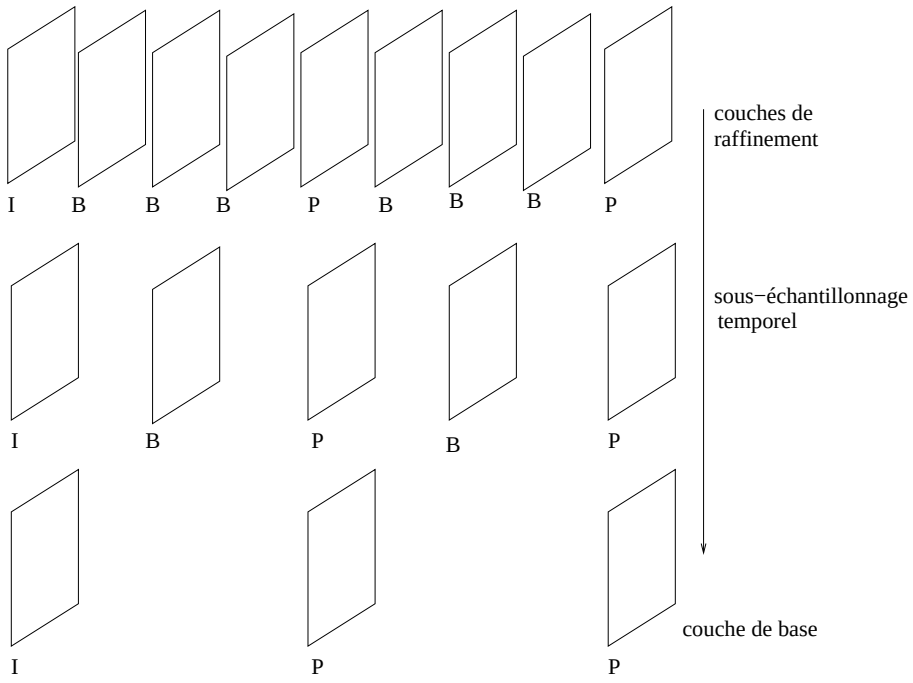


FIG. 1.5 – *Scalabilité temporelle pour un codeur type MPEG*

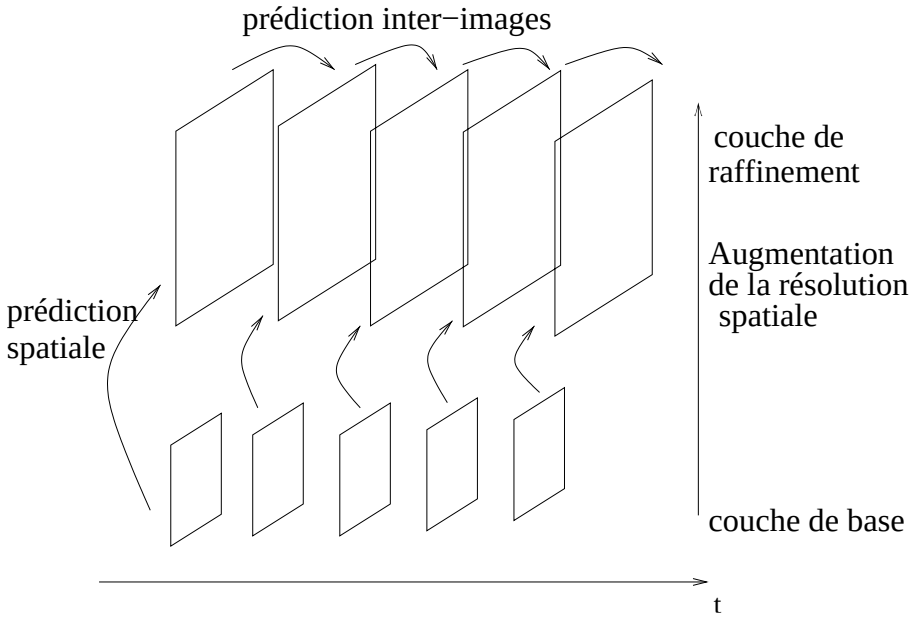


FIG. 1.6 – *Scalabilité spatiale pour un codeur type MPEG*

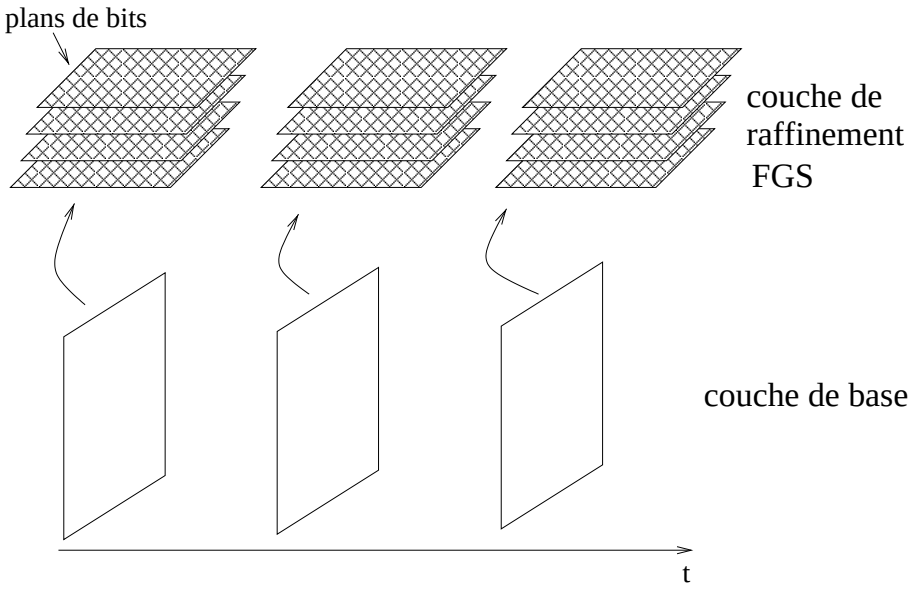


FIG. 1.7 – *Scalabilité en qualité du codeur MPEG-4 FGS*

la variation du PSNR (Peak Signal to Noise Ratio) au cours de la séquence. La séquence utilisée pour les tests est la séquence CREW encodée aux différents formats et débits donnés par le tableau ci-dessous pour le codeur AVC. Le codeur FGS n'a généré qu'un flux au moment de l'encodage, le flux généré a ensuite été décodé aux différents formats et débits voulus pour ce test. On remarque sur les courbes de PSNR que pour les formats haute résolution et CIF, le codeur scalable est 2dB moins performant que le codeur non scalable. Pour atteindre le même niveau de performances en terme de PSNR, le codeur scalable nécessite deux fois plus de débit.
Au niveau de la qualité subjective, les figures 1.9 et 1.10 montrent encore une fois la supériorité du codage non scalable sur le codage scalable. Les séquences reconstruites utilisées pour cette comparaison sont issues de [N5559 03].
Nous voyons que pour pouvoir être compétitif au niveau compression, un codeur scalable doit mettre en œuvre de nouvelles technologies de codage efficaces qui permettent de dépasser ces limitations. L'idée est d'utiliser des technologies naturellement scalables pour construire de nouveaux schémas de codage. Les nouveaux schémas scalables proposés utilisent une représentation multirésolution des données à l'aide d'un codage par ondelettes. Ces schémas bénéficient de la scalabilité naturelle des ondelettes et des performances de décorrélation qu'elles ont montrées dans le cadre du codage d'images fixes avec des codeurs tels que EZW [Shapiro 93], SPIHT [Said 96] ou encore le standard JPEG-2000 [Chrysafis 99]. Ces nouveaux schémas de codage d'images fixes et de codage vidéo seront présentés dans le prochain chapitre. Dans la section suivante, nous allons étudier les outils offrant une scalabilité naturelle et présenter les grands axes de

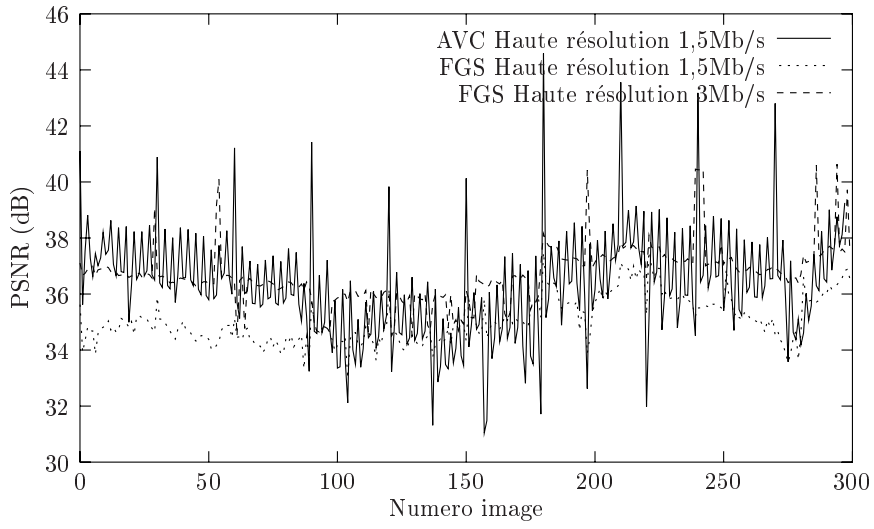
l'approche que nous proposons.

Résolution spatiale	Fréquence temporelle	débits
720x480	30Hz	1500kbs
360x240	30Hz	768kbs
180x120	15Hz	128kbs

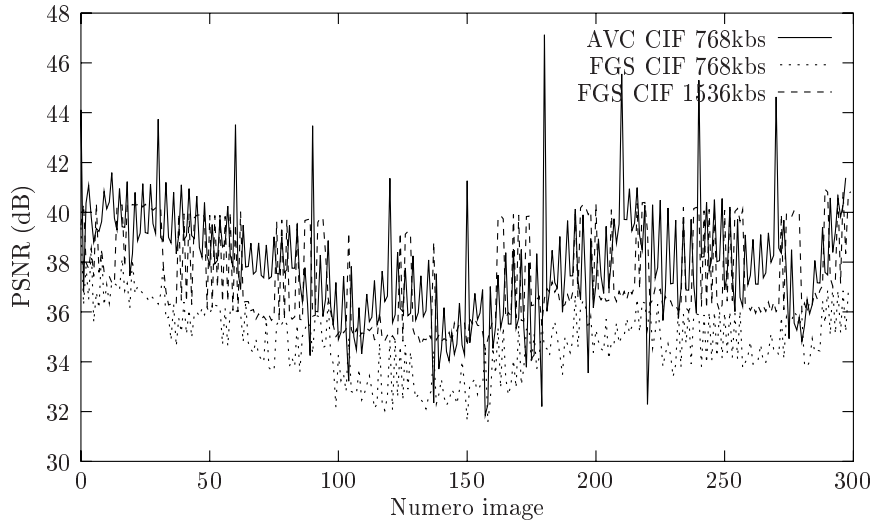
1.3 L'approche proposée

1.3.1 Rupture avec les schémas classiques

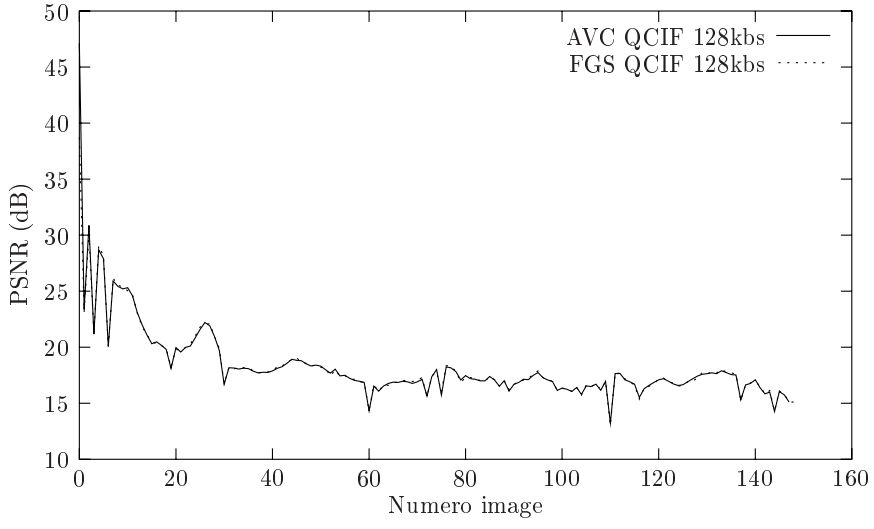
Depuis les prémices du codage vidéo numérique, les schémas de codage par prédiction ont montré un fort potentiel et des améliorations successives de ces schémas ont permis de toujours augmenter leurs performances en terme de codage. Dernièrement, le codeur H264 proposé par l'ITU a été retenu pour la nouvelle génération de codage par le groupe MPEG. Ce codeur améliore les performances des codeurs actuellement sur le marché (MPEG-2, MPEG-4) mais au détriment de la complexité du processus de codage-décodage. En considérant la relation des performances en compression et de la complexité des codeurs, nous pensons que les codeurs vidéo actuels ont atteint un seuil de saturation. Si l'on veut augmenter les performances en compression vidéo, il est nécessaire de construire de nouveaux schémas en rupture avec les approches classiques. De même si l'on s'intéresse à la relation performances et fonctionnalités, on voit là aussi que les codeurs actuels ne sont pas assez performants. Pour offrir une fonctionnalité comme la scalabilité, on doit développer de nouvelles technologies de codage naturellement scalables.
Pour ces raisons, nous proposons une approche en rupture avec les schémas classiques par prédiction. Nous utilisons une technique d'analyse-synthèse inspirée du domaine de la 3D où nous disposons de la représentation d'un objet ou d'une scène et de sa texture. Ces informations sont codées et transmises. Après décodage, la texture est plaquée sur la représentation de l'objet ou de la scène pour reconstruire la séquence vidéo.
La phase d'analyse estime le mouvement des objets dans la séquence vidéo et décorrèle les informations de mouvement, de texture et de formes pour un fonctionnement en mode objet. Le mouvement est représenté et estimé à l'aide d'un puissant outil d'analyse issu des maillages 2D déformables et la texture est représentée par une mosaïque dynamique. Après l'analyse, le mouvement, la texture et la forme sont codés séparément et indépendamment à l'aide d'un codage efficace par ondelettes. A la synthèse, la texture est plaquée sur le maillage et la forme afin de reconstruire les objets ou la séquence vidéo.
Le schéma d'analyse-synthèse est en rupture avec les approches de codage classiques. En effet, les codeurs vidéo actuels sont basés pixels, c'est-à-dire qu'ils visent à reconstruire la séquence vidéo en minimisant l'erreur de reconstruction calculée entre la séquence reconstruite et la séquence originale. Notre codeur, lui, est basé texture, c'est-à-dire qu'on ne cherche plus à minimiser une erreur calculée sur des pixels mais à modéliser au mieux les informations nécessaire à la reconstruction de la séquence:



a)



b)



c)

FIG. 1.8 – *Comparaison AVC-FGS, a) 720x480, 30Hz, 1,5Mbs, b) 360x240, 30Hz, 768kbs, c) 180x120, 15Hz, 128kbs*

FIG. 1.9 – *Comparaison AVC-FGS, format 180x120,15Hz, à 128kbs, a1) FGS image 75, b1) AVC image 75, a2) FGS image 25, b2) AVC image 25, a3) FGS image 102, b3) AVC image 102*

a1)

b1)

FIG. 1.10 – *Comparaison AVC-FGS, format 720x480, 30Hz, à 1500kbs, a1) FGS image 1, b1) AVC image 1*

mouvement, texture, forme. Quand on dispose de la représentation indépendante de chaque information, celles-ci sont codées de manière scalable et la synthèse vise à reconstruire la séquence avec le moins de distorsion visuelle possible, sans s'occuper de savoir si la séquence reconstruite est la reproduction exacte de la séquence originale. Des différences peuvent apparaître au niveau pixellique mais restent invisibles pour l'utilisateur. Par exemple, si le mouvement est codé avec pertes, nous supposons qu'une légère imprécision n'est pas perceptible et n'altère pas la qualité visuelle de la séquence rendue.

1.3.2 Les maillages pour l'analyse

La majeure partie des codeurs actuels utilise une représentation du mouvement par blocs. La méthode d'estimation du mouvement par blocs consiste pour chaque bloc de l'image à prédire à trouver un bloc correspondant dans l'image de référence. Cette technique d'estimation est simple et rapide mais souffre d'un problème de continuité. Le champ de mouvement est discontinu aux frontières de blocs adjacents et cette discontinuité crée des effets de blocs et implique un surcoût de codage. De plus, le mouvement par blocs est translationnel et les contractions, expansions et rotations ne peuvent pas être modélisées.
L'estimation du mouvement par maillage constitue une alternative à la représentation par blocs. Le maillage permet de mieux représenter le mouvement dans la scène et fournit un champ dense continu et inversible du mouvement. Le mouvement est porté par les nœuds du maillage, les mailles se déforment sous l'influence du mouvement, ceci permet d'assurer une continuité de la texture au cours du temps. La figure 1.11 montre la comparaison entre une compensation en mouvement par blocs et une compensation en mouvement par maillage. La continuité de la texture offerte par les maillages constitue un avantage important par rapport à la technique de codage que l'on va utiliser pour la texture. Dans le chapitre suivant, nous verrons plus en détail pourquoi cette continuité est importante, tandis que l'outil de codage sera présenté dans la prochaine section.
Dans notre approche, l'analyse du mouvement est faite à l'aide d'un maillage 2D déformable. L'estimation du mouvement consiste en la minimisation d'une fonctionnelle:

$$\sum_{p\in\Omega} \rho(I(p,t) - I(p - dp, t-1)),$$

avec Ω le support d'estimation, ρ la métrique d'erreur, la plus utilisée est $\rho(r) = r^2$ mais on peut aussi utiliser des estimateurs robustes tels que les M-estimateurs [Odobez 94]. $I(p,t)$ est la valeur de l'image I au point p et à l'instant t, et dp le champ de mouvement dense. Le champ de mouvement dense est exprimé en fonction du mouvement des nœuds du maillage:

$$dp = \sum_i w_i(p) dp_i$$

où $w_i(p)$ représente les coordonnées barycentriques de p par rapport aux nœuds i de la maille à laquelle le point appartient et dp_i représente le déplacement associé au nœud

i. Le déplacement dp en un point est obtenu par interpolation Lagrangienne. La minimisation est effectuée sur les nœuds du maillage, c'est-à-dire sur les dp_i des nœuds de t-1 à t. L'énergie est minimisée par une descente de gradient (type Gauss-Seidel), de manière itérative.
Afin de gagner en robustesse et en qualité du mouvement estimé, un maillage hiérarchique associé à une approche multirésolution sont utilisés pour l'estimation du mouvement, [Marquant 00]. Le maillage hiérarchique comporte plusieurs niveaux de représentation. Le nombre de nœuds dans chaque niveau croît du niveau grossier vers le niveau fin. Le mouvement estimé est de plus en plus précis à mesure que l'on tend vers un niveau fin de maillage. L'approche multirésolution consiste à adapter la résolution des images entre lesquelles est faite l'estimation à la précision du maillage. Ainsi, un maillage grossier estime le mouvement global de la scène entre des images basse résolution tandis qu'un maillage fin estime des mouvements locaux sur des images pleine résolution.
Une technique multi-grille est utilisée pour la propagation des valeurs des nœuds grossiers vers les nœuds fins. Si dp_i^l est le déplacement d'un nœud i au niveau de hiérarchie l, le déplacement dp_j^l d'un nœud j du niveau $l+1$ est donné par:
dp_i^l si j est le fils direct de i
$\frac{1}{2}(dp_i^l + dp_k^l)$ si j est milieu d'un arc au contact des nœuds i et k.
Cette technique de propagation des valeurs convient dans le cas de maillages réguliers et permet de garder la structure régulière du maillage. Cependant, lors de l'estimation de mouvement, le maillage se déforme et la pondération utilisée ci-dessus ne garantit pas la conservation de la structure des maillages inférieurs. Afin de conserver la structure du maillage déformé en passant d'un niveau grossier à un niveau fin, la pondération utilisée pour la répercussion des valeurs utilise le fait qu'un nœud peut s'exprimer comme une combinaison linéaire des nœuds du niveau de hiérarchie inférieur. Pour les nœuds fils direct d'un nœud du niveau inférieur la valeur est propagée directement. Pour les autres nœuds, on calcule les poids barycentriques par rapport aux nœuds des deux triangles du niveau inférieur pouvant contenir le nœud du niveau courant.
Associées à des reconditionnements spécifiques (limitation du déplacement des nœuds, gradient minimal de texture, support d'estimation, problème d'ouverture), ces techniques permettent d'assurer une meilleure convergence du système. Pour de plus amples informations sur les techniques de reconditionnement utilisées, nous redirigeons le lecteur vers [Cammas 03].

1.3.3 Les ondelettes pour le codage

La décomposition d'un signal par une transformation ondelette permet d'analyser un signal à différentes résolutions. Soit un signal f à analyser, la décomposition se fait par projection de f sur des bases de fonctions analysantes. Le signal f est projeté sur une base de fonctions d'échelle $\Phi_m = \{\phi_{m,n}(x)\}_{(m,n)\in Z^2}$ donnant une approximation du signal ou signal basse fréquence et sur une base de fonctions ondelettes $\Psi_m = \{\psi_{m,n}(x)\}_{(m,n)\in Z^2}$ donnant un signal haute fréquence. Dans [Mallat 89], Mallat montre que l'analyse ondelette est semblable à une analyse multirésolution. Le signal f est projeté sur un ensemble de sous-espaces vectoriels $\{V_m\}_{m\in Z}$ emboîtés

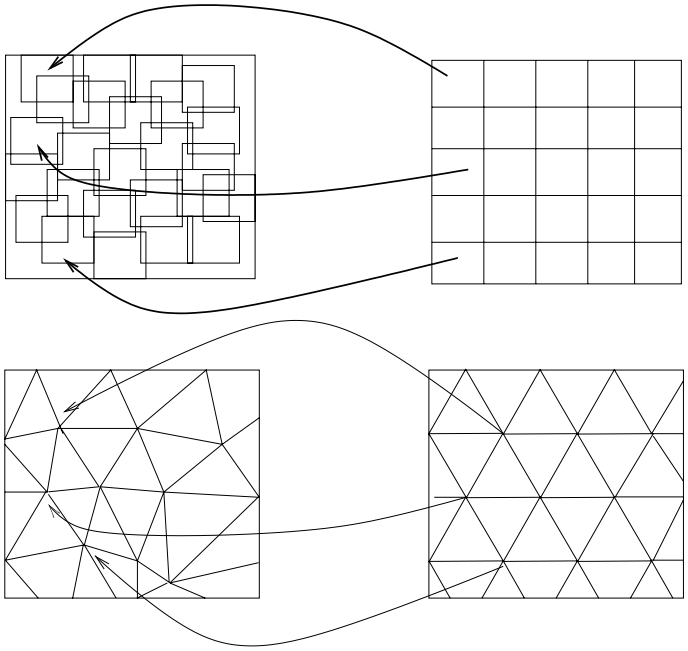


FIG. 1.11 – *Compensation en mouvement par blocs et par maillage*

où $\Phi_m = \{\phi_{m,n}(x)\}_{(m,n)\in Z^2}$ est une base orthogonale de V_m. Soit W_m le sous-espace vectoriel complémentaire de V_m dans V_{m+1}, une base orthogonale de W_m est $\Psi_m = \{\psi_{m,n}(x)\}_{(m,n)\in Z^2}$. Soit $P_{V_m}(f)$ l'approximation du signal au niveau m, et $P_{V_{m-1}}(f)$ l'approximation du signal au niveau m-1, les coefficients de détail ou coefficients *ondelette* pour passer du niveau m au niveau m-1 sont donnés par le produit scalaire: $\{< f, \Psi_{m,n} >\} n \in Z$. L'équation de décomposition s'écrit:

$$P_{V_{m-1}}(f) = P_{V_m}(f) + \sum_{n\in Z} < f, \Psi_{m,n} > \Psi_{m,n}.$$

En pratique, la transformée en ondelette discrète est vue comme une décomposition en sous-bandes réalisée par filtrage linéaire à l'aide d'une paire de filtres d'analyse: passe-bas h et passe-haut g. La figure 1.12 montre la relation entre les filtres d'analyse h et g et les fonctions de base ϕ et ψ. Les filtres de synthèse passe-bas et passe-haut sont respectivement les duaux des filtres d'analyse $\tilde{h}$ et $\tilde{g}$ et leur fonction de base associée sont $\tilde{\phi}$ et $\tilde{\psi}$.

L'algorithme de décomposition sur différentes résolutions est:

$$s_m(n) = \sum_k h(2n-k)s_{m-1}(k)$$
$$c_m(n) = \sum_k g(2n-k)s_{m-1}(k),$$

où s_m est le signal à la résolution m, c_m les coefficients d'ondelettes, h le filtre passe-bas et g le filtre passe-haut. Les filtres h et $\tilde{h}$ et g et $\tilde{g}$ sont associés à une base biorthogonale, la reconstruction du signal est exacte et s'écrit:

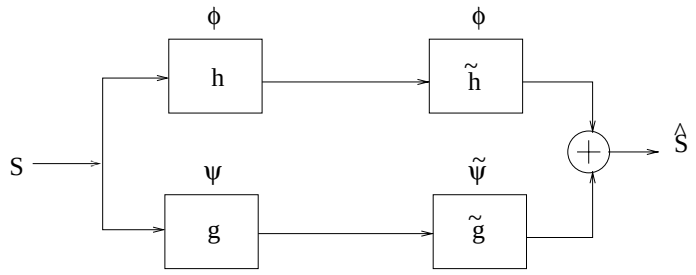


FIG. 1.12 – *Décomposition et synthèse d'un signal par ondelettes*

$$s_{m-1}(k) = \sum_n \tilde{h}(2n-k)s_m(n) + \sum_n \tilde{g}(2n-k)c_m(n).$$

L'analyse multirésolution consiste à décomposer un signal $s_0(n)$ en deux sous-bandes $s_1(n)$ et $c_1(n)$. Cette opération peut être itérée sur le signal $s_1(n)$ jusqu'à la résolution voulue.

L'analyse en ondelette a été étendue au cas bidimensionnel à l'aide d'une transformation par filtres séparables. Dans le cas d'une image, le signal est filtré horizontalement, puis verticalement, résultant en trois sous-bandes de détails et une sous-bande basse fréquence sur laquelle le filtrage peut être itéré. La figure 1.13 montre une telle décomposition.

La décomposition en ondelettes est une transformation spatio-fréquentielle. Elle permet la localisation en fréquence et dans l'espace. Elle assure une très bonne décorrélation et offre une représentation hiérarchique de l'information, permettant un gain en codage et une scalabilité naturelle. La transformée en ondelettes est utilisée dans différents codeurs d'images fixes tels que EZW [Shapiro 93], SPIHT [Said 96], JPEG-2000 [Chrysafis 99] ou EZBC [Hsiang 02]. L'orthogonalité des sous-bandes offerte par la décomposition en ondelettes permet de décomposer l'optimisation débit/distorsion global des coefficients sur chaque sous-bande:

$$R + \lambda D = \sum_i R_i + \lambda D_i,$$

où R et D représentent le débit et la distorsion globaux et R_i et D_i le débit et la distorsion de la sous-bande i. Dans le codeur JPEG-2000, elle a permis d'utiliser des techniques de codage efficaces telles que le codage arithmétique contextuel (CABAC) et un codage progressif par plans de bits offrant une granularité fine.

Pour ces raisons, l'objectif est d'étendre la transformée en ondelettes au cas tridimensionnel de la vidéo. La vidéo est considérée comme un cube d'information à décorréler. Les trois dimensions du cube sont représentées par les deux dimensions spatiales et la dimension temporelle. La figure 1.14 montre une telle décomposition. La décomposition est obtenue par filtrage linéaire le long de l'axe du temps (transformée temporelle), suivi d'une décomposition en ondelettes 2D classique. Le résultat de la transformée ondelettes t+2D est un cube formé de sous-bandes spatio-temporelles. Grâce à la transformée ondelettes, les informations de la séquence sont décorrélées et représentées de manière hiérarchique, offrant une scalabilité naturelle. La représentation multirésolution

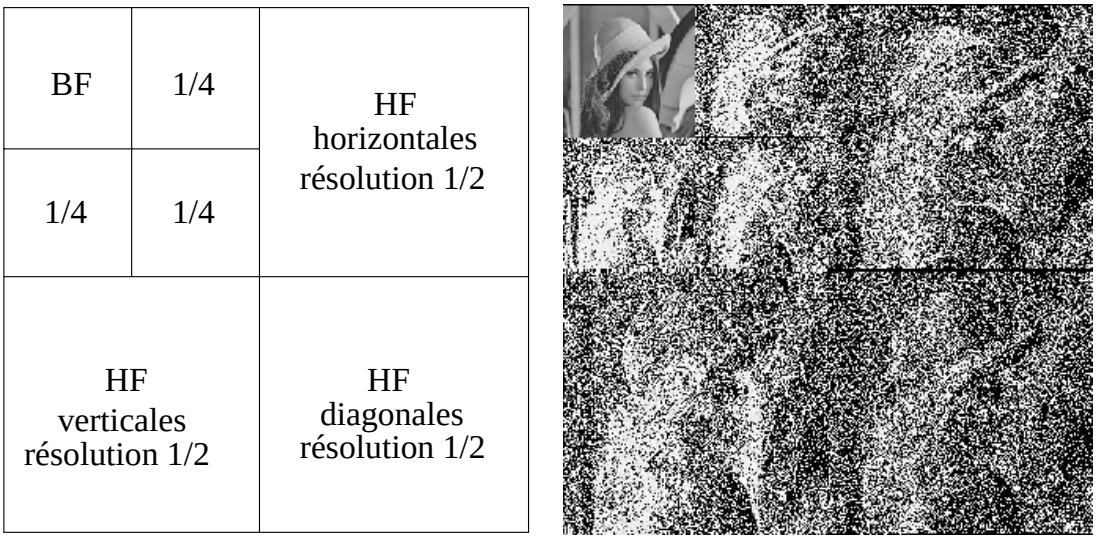


FIG. 1.13 – *Décomposition en ondelette d'une image*

associée à un codage progressif des données (par plans de bits) permettra d'offrir un flux d'information hautement scalable (spatialement, temporellement, en qualité) et aisé à manipuler.
La décomposition en ondelettes est d'autant plus efficace qu'elle s'applique sur un signal continu ou qui varie peu. Ceci est vrai pour le cas des images 2D où la corrélation spatiale entre des pixels voisins est très forte, c'est cette corrélation qui est utilisée dans les méthodes de codage par prédiction. Dans le cas d'une séquence vidéo, la corrélation temporelle est forte le long de l'axe des trajectoires de mouvement. La transformation temporelle de la décomposition t+2D doit donc se faire le long de ces trajectoires afin d'exploiter au mieux la redondance temporelle dans la vidéo. La transformée temporelle nécessite de prendre en compte un modèle de compensation en mouvement. Le chapitre suivant présentera les différents schémas exploitant la transformée temporelle ondelettes proposés dans la littérature et en fera une analyse critique. Dans la section suivante, nous présentons les buts de l'étude dans le cadre que nous venons d'exposer.

1.3.4 Objectifs de l'étude

L'objectif principal de l'étude est d'étudier des technologies innovantes permettant d'offrir un codage vidéo hautement scalable répondant au maximum au cahier des charges défini par le comité MPEG [N6025 03]. Pour ce faire, nous avons montré qu'il était nécessaire de construire un nouveau schéma de codage en rupture avec les approches de codage classiques. Le schéma par analyse-synthèse semble une base innovante pour une telle approche. De plus, la philosophie analyse-synthèse est en adéquation avec les outils d'analyse et de codage que nous désirons utiliser. En effet, les maillages présentés dans la section 1.3.2 représentent un puissant outil d'analyse

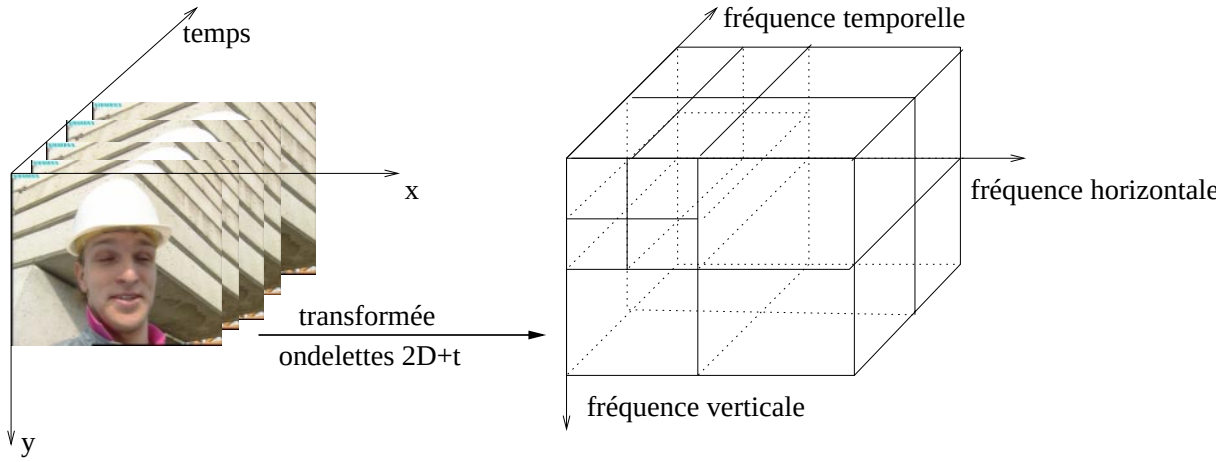


FIG. 1.14 – *Décomposition en ondelettes d'une séquence vidéo*

pour la représentation du mouvement. Ils offrent une modélisation plus réaliste du mouvement que les représentations par blocs et une continuité temporelle de la texture intéressante pour le codage par ondelettes. Dans la section 1.3.3, nous avons vu que la technique de codage par ondelettes est très efficace sur un signal continu, il paraît donc judicieux d'exploiter la continuité offerte par les maillages à l'aide d'une transformation par ondelettes.
Au niveau de la scalabilité, nous avons vu également qu'il était nécessaire pour un codage scalable performant d'utiliser des outils offrant une scalabilité naturelle. La représentation du mouvement par maillage hiérarchique permet aisément un codage scalable du mouvement et le codage par ondelettes t+2D offre une scalabilité naturelle à la représentation des informations à coder.
Pour ces raisons, nous utilisons un codeur vidéo basé maillage pour la représentation et l'estimation du mouvement et basé ondelettes t+2D pour le codage des informations de la séquence. Dans ce cadre, nous voyons apparaître des points essentiels à étudier pour obtenir un codage vidéo scalable performant en terme de compression pure et en terme de scalabilité.
Le premier point a été évoqué dans la section 1.3.3 et concerne l'exploitation du mouvement dans la transformée ondelettes t+2D. Nous avons vu qu'il était nécessaire d'exploiter ce mouvement dans la transformée temporelle afin de décorréler aux mieux les informations à coder. Nous allons donc étudier de quelle manière exploiter ce mouvement dans la transformée temporelle.
Le deuxième point à étudier concerne la représentation du mouvement. Quelque soit la représentation choisie pour le mouvement (maillage, blocs, ...), le problème des occlusions dans la séquence vidéo est un problème non résolu. Le phénomène d'occlusions apparaît lorsque des zones apparaissent ou disparaissent dans la séquence vidéo. Il devient difficile d'effectuer une bonne prédiction et une estimation correcte du mouvement dans ces zones et ceci implique un surcoût de codage. Dans la deuxième partie de ce

manuscrit, nous adresserons ce problème en étudiant les solutions proposées dans la littérature et en proposant une solution basée sur les maillages et les lignes de rupture introduites dans [Marquant 00].

Chapitre 2

Codage vidéo par ondelettes: un état de l'art

Nous avons vu dans le chapitre précédent qu'il était nécessaire de développer de nouveaux schémas de codage vidéo si l'on désirait d'une part améliorer encore les performances en terme de compression et d'autre part offrir des fonctionnalités telles que la scalabilité sans pertes de performances. Nous avons également étudié quelques outils nous permettant de mettre en œuvre de tels schémas avec notamment l'utilisation de la décomposition en ondelettes sur les trois axes de la vidéo. Nous allons étudier dans ce chapitre les nouveaux schémas de codage proposés dans la littérature mettant en œuvre cette décomposition. Les techniques de codage vidéo par ondelettes 3D se composent de quatre briques principales: la transformation temporelle, la transformation spatiale, le mouvement et le codage des sous-bandes spatio-temporelles. Nous avons consacré une section de ce chapitre à chacune de ces briques. La première section abordera l'exploitation du mouvement dans la transformée ondelettes sur l'axe temporel, puis nous étudierons l'influence du mouvement sur la qualité de la décorrélation temporelle. Ensuite, nous étudierons l'utilisation de la décomposition spatiale et enfin la dernière section traitera des outils utilisés pour le codage de sous-bandes ondelettes issues d'une décomposition spatiale et leur extension au codage de sous-bandes spatio-temporelles (3D ou 2D+t ou t+2D).

2.1 Transformée en ondelettes temporelles

2.1.1 Nécessité d'exploiter le mouvement

Les premiers schémas proposés dans la littérature appliquent une transformée ondelettes 3D séparable sur la séquence vidéo en considérant de la même manière les trois axes de la vidéo. Dans [Podilchuk 95] et [Karlsson 88], une décomposition en sous-bandes à un niveau sur l'axe temporel est appliquée sur la séquence à l'aide d'un filtre de Haar. Les sous-bandes résultantes sont ensuite transformées spatialement par un filtre Daubechies 9/7 sur 1 ou 2 niveaux de décomposition. Dans [Kim 97], les auteurs

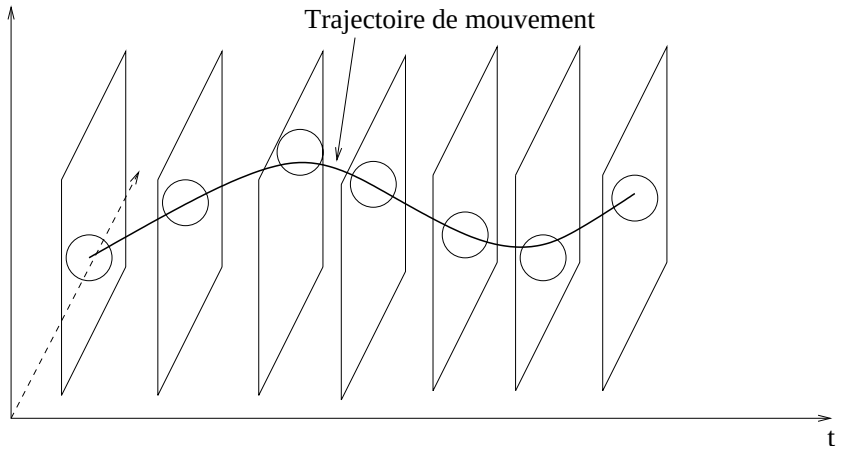


FIG. 2.1 – *Filtrage temporel le long des trajectoires de mouvement*

utilisent le filtre de Daubechies 9/7 dans les trois dimensions. Dans ces schémas, la transformation sur l'axe temporel ne prend pas en compte le mouvement et considère un filtrage 1D en chaque pixel. L'équation de filtrage s'écrit:

$$C_t(x,y) = \sum_k I_{t+k}(x,y) * g(k)$$

où $g(k)$ est le coefficient d'indice k du filtre, (x,y) la position du pixel filtré, $C_t(x,y)$ le coefficient ondelettes en (x,y) à la position temporelle t, $I_t(x,y)$ la valeur du pixel (x,y) à l'instant t.
Dans [Kim 00], les auteurs étendent le schéma proposé dans [Kim 97] à l'utilisation d'un modèle de compensation en mouvement dans l'étape de transformée temporelle. Le modèle de compensation utilise une représentation hiérarchique par blocs. L'équation de la transformée temporelle s'écrit alors:

$$C_t(x,y) = \sum_k W_{t+k \to t}(I_{t+k}(x,y)) * g(k)$$

où $W_{t \to t+k}$ est l'opérateur de compensation en mouvement de l'image t+k vers l'image t. Dans [Kim 00], les auteurs comparent les schémas de codage avec et sans compensation en mouvement dans la transformée temporelle. Les résultats montrent que les séquences reconstruites obtenues sans compensation en mouvement présentent un flou en comparaison de celle obtenues avec compensation en mouvement.
De plus, la décorrélation par ondelettes est d'autant plus efficace que le signal sur lequel elle s'applique est continu. Or, pour un signal vidéo, en considérant l'axe temporel, la corrélation du signal est plus forte le long des trajectoires de mouvement que le long de l'axe temporel direct. La décorrélation temporelle des informations d'une séquence vidéo doit être appliquée le long des trajectoires de mouvement afin d'être vraiment efficace, figure 2.1. Nous allons voir dans les parties suivantes comment les schémas proposés dans la littérature prennent en compte ce mouvement.

2.1.2 Compensation de mouvement global

Dans [Taubman 94] et [Wang 99], les auteurs présentent un schéma de codage vidéo par transformée ondelette 2D+t. Les images de la séquence vidéo sont d'abord alignées sur une grille de référence à l'aide d'une compensation en mouvement global. Les images alignées sont ensuite décomposées à l'aide d'une transformée ondelette spatiale suivie d'une transformée ondelette temporelle. L'utilisation d'un mouvement global limite l'efficacité de la décorrelation par transformation ondelette car il ne prend pas en compte les mouvements locaux dans la scène, la transformation temporelle ne décorrèle pas au mieux les informations. De plus, dans le cas de mouvements sous-pixelliques, des problèmes d'échantillonnage apparaissent. La reconstruction parfaite n'est pas assurée. La décorrélation des informations est plus efficace par l'utilisation de modèles de mouvement plus précis qu'un mouvement global.

2.1.3 Compensation de mouvement par blocs, mouvement unidirectionnel

L'utilisation de mouvement basé blocs permet de mieux décorréler l'information lors de la transformée temporelle par rapport aux schémas par compensation de mouvement globale. Les schémas basés blocs effectuent la transformation temporelle sur des blocs déplacés. Au cours du temps, des régions se découvrent et se recouvrent dans la séquence vidéo. Lors de la compensation en mouvement par blocs entre une image A et une image B avec un champ de mouvement unidirectionnel, ceci se traduit par des recouvrements de blocs et des régions non couvertes dans les images. Les pixels doivent alors être filtrés différemment selon qu'ils appartiennent à une zone de mise en correspondance, une zone recouverte ou une zone découverte. A l'aide d'un champ de mouvement avant allant de l'image B vers A, les pixels sont classés de la manière suivante:

- les pixels *connectés* qui appartiennent à une région de mise en correspondance entre A et B,
- les pixels *non connectés* qui appartiennent à une région de A recouverte dans B ou qui appartiennent à une région de B découverte, qui n'est pas présente dans A,
- les pixels *multiplement connectés* qui appartiennent à l'image A et sont mis en correspondance avec plus d'un pixel dans l'image B.

La figure 2.2 montre les schémas de transformation temporelle proposés respectivement par Ohm [Ohm 94] et Choi et Woods [Choi 99]. Ces schémas utilisent un filtre de Haar pour la transformée temporelle entre deux images A et B. Ohm place la basse fréquence sur l'image B et la haute fréquence sur l'image A. Pour le schéma de Ohm, les équations d'analyse entre des pixels connectés sont les suivantes:

$$L(m,n) = \frac{B(m,n)+\widetilde{A}(m+d_m,n+d_n)}{2}$$
$$H(m,n) = \frac{\widetilde{B}(m+\bar{d}_m,n+\bar{d}_n)-A(m,n)}{2}$$

où $L[m,n]$ et $H[m,n]$ sont les basses et hautes fréquences temporelles, $A[m,n]$ et $B[m,n]$ les images à filtrer,(d_m,d_n) est le vecteur mouvement entre A et B, $\widetilde{A}$ et $\widetilde{B}$

sont les valeurs interpolées de A et B dans le cas de mouvements sous-pixelliques et $(\bar{d}_m,\bar{d}_n)$ le vecteur mouvement inverse de (d_m,d_n) arrondi au plus proche entier si la précision est sous-pixellique.
Dans le cas d'un pixel multiplement connecté dans l'image A, son correspondant est choisi par l'ordre de parcours, en général le premier pixel correspondant rencontré est choisi. Les autres correspondants dans l'image B deviennent isolés (non connectés), la basse fréquence en ces pixels est:

$$L(m,n) = B(m,n)$$

Pour les pixels de A non connectés, la haute fréquence devient:

$$H(m,n) = \frac{\widetilde{E}(m-\widehat{d}_m,n-\widehat{d}_n)-A(m,n)}{2}$$

où E est l'image codée-décodée précédente par rapport à A, $\widetilde{E}$ est la valeur interpolée de E dans le cas de mouvements sous-pixelliques, $(\widehat{d}_m,\widehat{d}_n)$ le vecteur mouvement entre E et A.

Le schéma de Choi et Woods [Choi 99] échange la place des hautes et basses fréquences par rapport à Ohm, comme on le voit sur la figure 2.2, b. Ceci permet de limiter la présence de pixels non connectés sur l'image de hautes fréquences et ainsi de limiter l'apparition de fortes énergies dans ces sous-bandes. Les équations d'analyse du schéma de Choi et Woods deviennent alors:

– pour les pixels connectés

$$\begin{aligned} L[m-\bar{d}_m, n-\bar{d}_n] &= \tfrac{1}{\sqrt{2}}\widetilde{B}[m-\bar{d}_m+d_m, n-\bar{d}_n+d_n] + \tfrac{1}{\sqrt{2}}A[m-\bar{d}_m, n-\bar{d}_n] \\ H[m,n] &= \tfrac{1}{\sqrt{2}}B[m,n] - \tfrac{1}{\sqrt{2}}\widetilde{A}[m-d_m, n-d_n] \end{aligned}$$

– pour les pixels non connectés

$$\begin{aligned} L[m,n] &= \tfrac{2A[m,n]}{\sqrt{2}} \\ H[m,n] &= \tfrac{1}{\sqrt{2}}(B[m,n] - \widetilde{A}[m-d_m, n-d_n]) \end{aligned}$$

où $L[m,n]$ et $H[m,n]$ sont les basses et hautes fréquences temporelles, $A[m,n]$ et $B[m,n]$ les images à filtrer, (d_m,d_n) est le vecteur mouvement entre A et B et $\bar{d}_m$ et $\bar{d}_n$ sont les arrondis à l'entier le plus proche de d_m et d_n. $\widetilde{A}$ et $\widetilde{B}$ sont les valeurs interpolées si les vecteurs mouvement ont une précision sous-pixellique. Dans la méthode de Choi et Woods, les coefficients du filtre par rapport à Ohm sont normalisés et les pixels non connectés dans la basse fréquence sont mis à échelle de la résolution par $\frac{1}{\sqrt{2}}$.

Le filtrage est ainsi appliqué récursivement sur plusieurs niveaux de résolution temporelle. Les deux schémas peuvent être étendus à des filtres plus longs que le filtre de Haar, mais la méthode devient alors très complexe à cause des pixels non connectés. De plus, dans le cas de mouvements sous-pixelliques, ces schémas n'assurent pas la reconstruction parfaite.

Dans la suite des schémas que nous allons présenter, ces problèmes de réversibilité

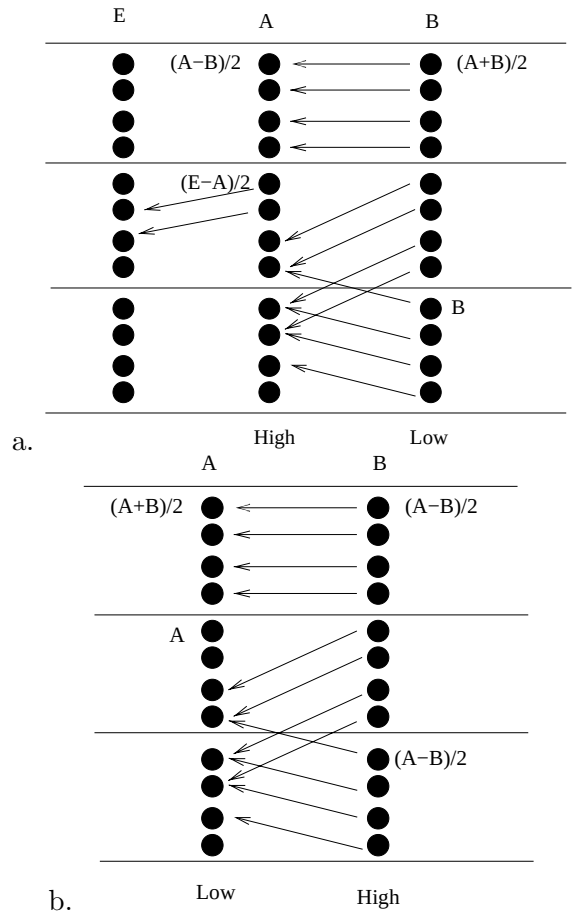


FIG. 2.2 – *Transformée temporelle avec mouvement par blocs: a) schéma de Ohm, b) schéma de Choi et Woods*

et de pixels déconnectés ont été résolus grâce d'une part à l'introduction du lifting dans le schéma de la transformée temporelle, et d'autre part par l'utilisation d'un mouvement bidirectionnel.

2.1.4 Introduction du schéma lifting

2.1.4.1 Le principe du lifting

Le schéma lifting a été introduit par Sweldens [Sweldens 95]. Il permet une transformée en ondelettes par un procédé simple, rapide et réversible. Le principe est le même que pour la transformée en ondelettes classiques, exploiter les corrélations du signal pour aboutir à un ensemble d'informations compactes et diminuer l'entropie du signal. Le schéma se compose de trois étapes, figure 2.3. L'opérateur SPLIT décompose le signal initial x en deux sous-signaux x_e (échantillons d'indice pair) et x_o (échantillons d'indice impair). L'opérateur de prédiction P est appliqué au signal x_e, on obtient l'équation suivante:

$$d = x_o - P(x_e),$$

où d est le signal résiduel de la prédiction représentant les hautes fréquences issues de la transformation. L'opérateur de mise à jour U est ensuite appliqué au signal d pour former le signal basse fréquence s:

$$s = x_e + U(d).$$

Les étapes du lifting peuvent être appliquées récursivement sur le signal s afin d'obtenir différents niveaux de résolutions fréquentielles. La transformation inverse du lifting est simple.
Le schéma lifting peut être obtenu à partir du schéma général de filtrage classique. Les correspondants des filtres classiques en version lifting sont obtenus par factorisation de la matrice polyphase. Le détail de la factorisation et les filtres correspondants sont donnés en annexe.
Dans le cadre d'une transformée temporelle sur un signal vidéo, nous avons vu qu'il était nécessaire d'exploiter le mouvement dans la transformée. Dans le cas de mouvements sous-pixelliques, l'utilisation d'une compensation en mouvement dans la transformée n'assure plus la reconstruction parfaite. L'utilisation du lifting permet de résoudre ce problème car le schéma est réversible quelle que soit la précision du mouvement utilisée pour la compensation. Cependant, dans [Andre 04], les auteurs montrent que la transformée lifting et la transformée classique par des filtres correspondants ne sont pas toujours équivalentes. Les équations d'analyse de la transformée temporelle par un filtre 5/3 sont respectivement pour le filtrage classique et le filtrage lifting:

– Transformée classique:

$$\begin{aligned} H_k[m,n] &= x_{2k+1}[m,n] - \tfrac{1}{2}(W_{2k\to 2k+1}(x_{2k}[m,n]) + W_{2k+2\to 2k+1}(x_{2k+2}[m,n])) \\ L_k[m,n] &= \tfrac{3}{4}x_{2k}[m,n] + \tfrac{1}{4}(W_{2k-1\to 2k}(x_{2k-1}[m,n]) + W_{2k+1\to 2k}(x_{2k+1}[m,n])) \\ &\quad + \tfrac{1}{8}(W_{2k-2\to 2k}(x_{2k-2}[m,n]) + W_{2k+2\to 2k}(x_{2k+2}[m,n])) \end{aligned}$$

– Lifting:

$$\begin{aligned} H_k[m,n] &= x_{2k+1}[m,n] - \tfrac{1}{2}(W_{2k\to 2k+1}(x_{2k}[m,n]) + W_{2k+2\to 2k+1}(x_{2k+2}[m,n])) \\ L_k[m,n] &= x_{2k}[m,n] + +\tfrac{1}{4}(W_{2k-1\to 2k}(H_{k-1}[m,n]) + W_{2k+1\to 2k}(H_k[m,n])) \end{aligned}$$

L_k et H_k représentent respectivement les basses et hautes fréquences, $W_{i\to j}$ est l'opérateur de compensation en mouvement de i vers j, x est le signal original à filtrer.
On remarque que les hautes fréquences sont obtenues par le même filtrage dans les deux techniques. L'équivalence entre les deux techniques est obtenue si les basses fréquences obtenues par des filtrages différents sont les mêmes.

$$\begin{aligned} L_k[m,n] = {} & x_{2k}[m,n] \\ & +\tfrac{1}{4}(W_{2k-1\to 2k}(x_{2k-1}[m,n]) \\ & \quad + W_{2k+1\to 2k}(x_{2k+1}[m,n])) \\ & -\tfrac{1}{8}(W_{2k-1\to 2k}(W_{2k-2\to 2k-1}(x_{2k-2}[m,n])) \\ & \quad + W_{2k-1\to 2k}(W_{2k\to 2k-1}(x_{2k}[m,n])) \\ & \quad + W_{2k+1\to 2k}(W_{2k\to 2k+1}(x_{2k}[m,n])) \\ & \quad + W_{2k+1\to 2k}(W_{2k+2\to 2k+1}(x_{2k+2}[m,n]))) \end{aligned}$$

Equivalence si:

$$W_{2k-1\to 2k}(W_{2k\to 2k-1}(x[m,n])) = x[m,n] \quad (2.1)$$

$$W_{2k-1\to 2k}(W_{2k-2\to 2k-1}(x[m,n])) = W_{2k-2\to 2k}(x[m,n]) \quad (2.2)$$

L'équivalence est obtenue si le champ de mouvement est réversible (équation 2.1) et additif (équation 2.2). Ces deux conditions sont rarement vérifiées par les champs de mouvement utilisés pour la compensation dans la transformée. Par exemple, c'est le cas du mouvement par blocs. Le calcul d'un mouvement vérifiant ces deux propriétés est toutefois assez difficile, de plus ce mouvement contraint serait encore plus éloigné du mouvement réel de la séquence vidéo que le mouvement non contraint. Le filtrage avec mouvement contraint résulterait en de hautes fréquences avec de fortes amplitudes. Dans le cas où l'on désire utiliser la transformée lifting exactement similaire à la transformée classique transverse, dans [Andre 04], les auteurs proposent un schéma général de transformée lifting qui met en œuvre l'exact transverse de la transformée classique. Les hautes fréquences sont obtenues comme dans le schéma classique, les basses fréquences sont représentées par les échantillons d'indice pair du signal original. Ce schéma utilise des filtres de transformée tronqués, le schéma ne met en œuvre que l'étape de prédiction du schéma lifting. Ce schéma lifting est utilisé dans [Luo 01] que nous présentons plus loin.

$$\begin{aligned} H_k[m,n] &= x_{2k+1}[m,n] - \tfrac{1}{2}(W_{2k\to 2k+1}(x_{2k}[m,n]) + W_{2k+2\to 2k+1}(x_{2k+2}[m,n])) \\ L_k[m,n] &= x_{2k}[m,n] \end{aligned}$$

Dans les sections suivantes, nous allons étudier des schémas de transformée temporelle utilisant le schéma lifting que nous venons de présenter (avec étapes P et U du schéma lifting).

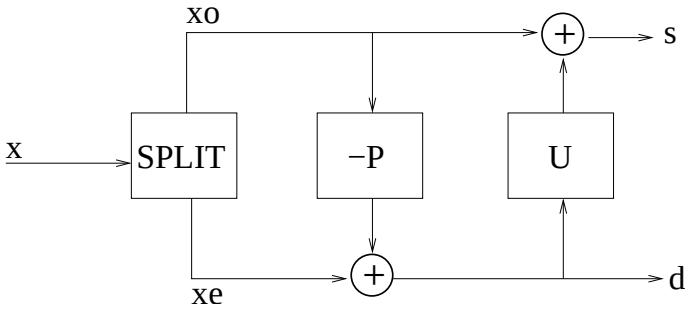


FIG. 2.3 – *Principe du schéma lifting*

2.1.4.2 Lifting et tranformée temporelle

Dans [Secker 01], Secker et Taubman ont introduit le schéma lifting dans la transformée temporelle. Dans leur schéma, la compensation en mouvement est appliquée simultanément à la transformation temporelle. Dans la première version de ce schéma, le mouvement est représenté par un mouvement par blocs et le champ de mouvement est bidirectionnel, c'est-à-dire que l'on dispose du mouvement de A vers B et du mouvement de B vers A. Les équations d'analyse pour un filtre de Haar sont les suivantes:

$$\begin{aligned} H_k[m,n] &= \tfrac{1}{2}(x_{2k+1}[m,n] - (W_{2k\to 2k+1}(x_{2k}))[m,n]) \\ L_k[m,n] &= x_{2k}[m,n] + (W_{2k+1\to 2k}(H_k))[m,n] \end{aligned}$$

où x_{2k} et x_{2k+1} sont les images à filtrer, H_k et L_k sont les images de haute et basse fréquences respectivement, $W_{2k\to 2k+1}$ l'opération de compensation en mouvement de x_{2k} vers x_{2k+1} et $W_{2k+1\to 2k}$ l'opération de compensation en mouvement de x_{2k+1} vers x_{2k}. Ces étapes de lifting sont effectuées récursivement sur plusieurs niveaux de résolution temporelle. On remarque que la haute fréquence correspond à un résidu de prédiction de x_{2k+1} par x_{2k}, cette haute fréquence ne dépend que de la précision du mouvement estimé de x_{2k} vers x_{2k+1}. La haute fréquence contiendra d'autant peu d'énergie que le champ de mouvement sera bon. On remarque alors que dans ce cas, si la haute fréquence tend vers 0, la basse fréquence n'est rien d'autre que l'image x_{2k} filtrée le long de la trajectoire. Ainsi, ce schéma permet d'assurer deux propriétés essentielles lors d'une transformation ondelette: des hautes fréquences avec peu d'énergie et des basses fréquences proches du signal original. La première propriété impacte sur le coût de codage des sous-bandes, la deuxième propriété permet d'éviter les artefacts lors de la reconstruction à une résolution temporelle inférieure à la résolution d'origine. Par l'utilisation de deux champs de mouvement, cette méthode permet d'éviter l'apparition de pixels non connectés dans le schéma de la transformée. Cependant, elle implique alors un coût de codage important pour les deux champs de mouvement. Ce surcoût est d'autant plus important que le filtre utilisé pour la transformée est long. Les équations d'analyse suivantes montrent la méthode de transformée temporelle à l'aide d'un filtre 5/3, les notations sont les mêmes que pour les équations précédentes,

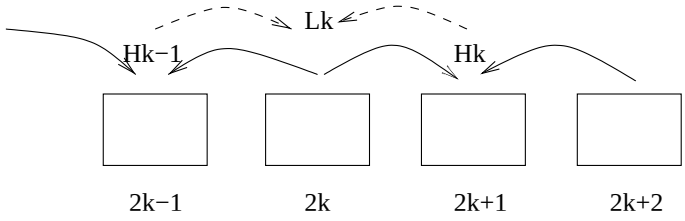


FIG. 2.4 – *Transformée temporelle avec filtre 5/3: schéma de Secker et Taubman*

la figure 2.4 illustre ces équations:

$$\begin{aligned} H_k[m,n] &= x_{2k+1}[m,n] - \tfrac{1}{2}(W_{2k\to 2k+1}(x_{2k})[m,n] + W_{2k+2\to 2k+1}(x_{2k+2})[m,n]) \\ L_k[m,n] &= x_{2k}[m,n] + \tfrac{1}{4}(W_{2k-1\to 2k}(H_{k-1})[m,n] + W_{2k+1\to 2k}(H_k)[m,n]) \end{aligned}$$

La décorrélation est d'autant plus forte que la transformée utilise un filtre long, faisant ainsi intervenir plus de redondance, mais la complexité opératoire et le nombre de champ de mouvement nécessaire augmentent d'autant plus. Pour une transformation sur un groupe de huit images, le nombre de champ de mouvement nécessaire pour un niveau de résolution avec un filtre de Haar est de 8 alors que ce nombre est de 14 pour un filtre 5/3.

Une amélioration de cette méthode a été proposée par les mêmes auteurs [Secker 02]. Cette fois, le mouvement est représenté par un maillage déformable. L'utilisation d'un maillage pour la représentation du mouvement permet de réduire le coût de codage du mouvement qui constituait une des principales limitations pour ce schéma. En effet, un champ de mouvement par maillage est inversible, en ce sens qu'il est possible à partir d'un champ de mouvement entre deux images A et B, $W_{A\to B}$ d'obtenir le champ de mouvement de B vers A par simple inversion: $W_{B\to A} = -W_{A\to B}$. Le coût du mouvement est divisé par deux grâce à cette méthode et il est alors semblable à une méthode de transformée temporelle avec mouvement unidirectionnel.
Cependant, les auteurs montrent que le coût de mouvement peut encore être diminué grâce à l'exploitation de la continuité temporelle de la représentation du mouvement par maillage. En effet, la transformée ondelette temporelle est appliquée sur plusieurs niveaux de résolution temporelle et nécessite des champs de mouvement entre des images non immédiatement successives, figure 2.5. Sur la figure 2.5, on voit que les champs de mouvement $W_{1\to 2}$, $W_{2\to 1}$, $W_{2\to 3}$ et $W_{3\to 1}$ peuvent être déduit des champs de mouvement $W_{3\to 2}$ et $W_{1\to 3}$. D'une part, $W_{i\to j} = -W_{j\to i}$ et d'autre part: $W_{i\to k} = W_{i\to j} + W_{j\to k}$. Finalement, grâce à l'utilisation des maillages pour la représentation du mouvement, le nombre de champ de mouvement nécessaire pour une transformation ondelette temporelle sur 8 images, avec un filtre 5/3 sur trois niveaux de décomposition est passé de 22 $(2 \times (7 + 3 + 1))$ champs de mouvement à 7.
L'utilisation de champ de mouvement représenté par un maillage assure une continuité de la texture lors de la compensation en mouvement, cette continuité n'est pas assurée si le champ de mouvement est un mouvement par blocs. La technique de compensation

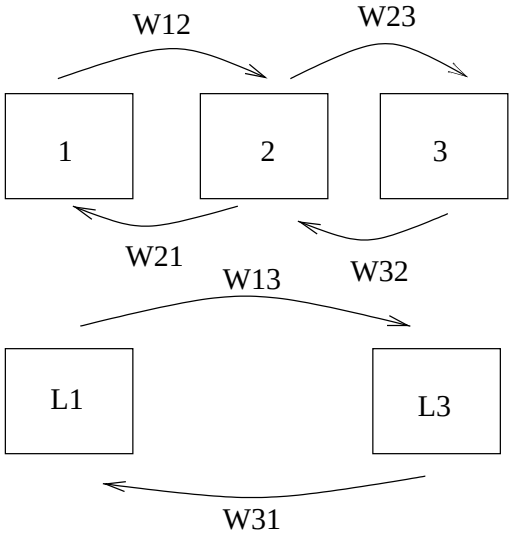


FIG. 2.5 – *Représentation des champs de mouvement pour une décomposition ondelette temporelle sur deux niveaux de résolution*

par maillage effectue un plaquage de texture sur une grille d'échantillonnage avec des étirements et des contractions de la texture dans les zones d'expansion et de contraction du mouvement. Dans les schémas présentés précédemment, ceci se traduisait par l'apparition de pixels non connectés. Nous avons vu les avantages que pouvait apporter la continuité de la texture grâce aux maillages, cependant il est à noter que cette continuité peut engendrer quelques inconvénients. En effet, lors de la synthèse temporelle, il peut arriver que des erreurs de quantification locales soient plaquées sur une zone d'expansion du mouvement, ces erreurs qui avaient un impact local avant la compensation en mouvement sont alors étalées sur la texture et peuvent avoir un effet néfaste sur la reconstruction globale de la séquence ainsi que sur la mesure objective de sa qualité.

Une autre approche a été proposée dans [Luo 01] utilisant une transformation temporelle basée lifting et un champ de mouvement bidirectionnel. La transformation ondelette temporelle est effectuée sur un groupe de neuf images à l'aide d'un filtre 5/3 tronqué, c'est-à-dire que seule l'étape de prédiction du lifting des hautes fréquences est appliquée. La figure 2.6 montre les étapes de prédiction et de mise à jour avec un filtre 5/3 lifting et l'étape de prédiction du lifting avec le filtre 5/3 tronqué. L'équation d'analyse de la transformée est:

$$H_i = I_{2i+1} - \tfrac{1}{2}(W_{2i\rightarrow 2i+}(I_{2i}) + W_{2i+2\rightarrow 2i+1}(I_{2i+2}))$$

avec I_i l'image originale, H_i la sous-bande haute fréquence et $W_{i\rightarrow j}$ l'opération de compensation en mouvement de i vers j. La figure 2.7 illustre le schéma de lifting. Les

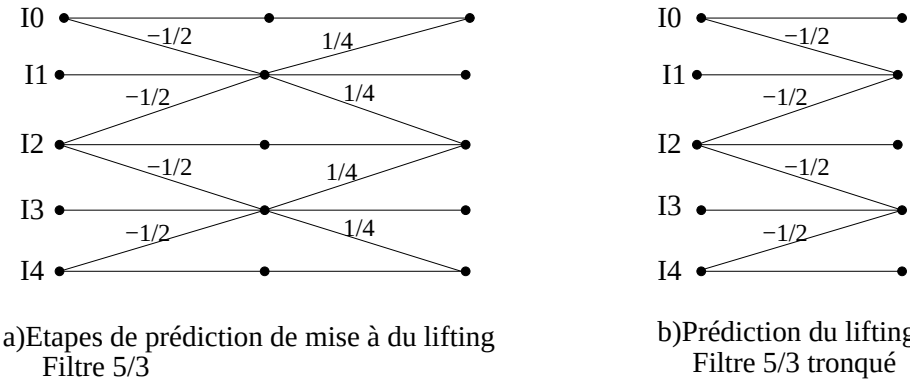


FIG. 2.6 – *Etapes du lifting avec filtre 5/3 et filtre 5/3 tronqué*

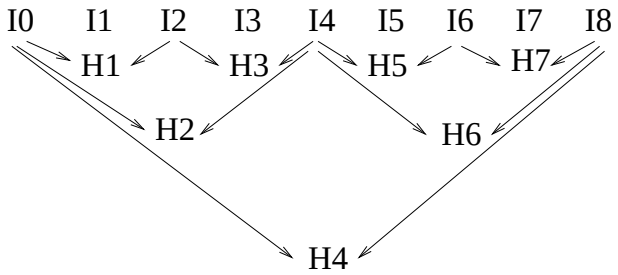


FIG. 2.7 – *Transformation ondelette temporelle sur neuf images pour le schéma MCLIFT*

sous-bandes hautes fréquences de chaque niveau de résolution sont obtenues par filtrage des images originales. Les basses fréquences sont représentées par les deux images extrêmes du groupe. Le champ de mouvement est basé bloc et bidirectionnel. Le schéma d'obtention des sous-bandes est proche d'un schéma prédictif MPEG type IBBBPBBB. En effet, les sous-bandes aux extrémités sont codées en mode Intra et les sous-bandes hautes fréquences correspondent à des résidus de prédiction bidirectionnelle. L'avantage de ce schéma par rapport au schéma MPEG est que les références pour la prédiction des images sont plus proches que dans le schéma MPEG IBBBPBBB. Avec MPEG, les images B sont prédites par rapport à des images I et P, pour I_1 la distance est de 3 intervalles de temps entre I_1 et une de ses références I_4. Avec le schéma MCLIFT, la distance de I_1 à ses références I_0 et I_2 est de un intervalle de temps. La prédiction par des références plus proches permet d'améliorer l'efficacité de la décorrelation temporelle.

Le schéma proposé dans [Bottreau 02] effectue une transformée temporelle sur un groupe de huit images. Les images sont filtrées par paire à l'aide d'un filtre de Haar sur trois niveaux de résolutions. Une première version du schéma utilise un mouvement avant pour la compensation, les pixels déconnectés ou multiplement connectés sont gérés comme dans [Choi 99]. La transformée temporelle avec précision sous-pixellique

Schéma	Filtres	Convolutif	Lifting
Ohm	Haar	0.875	0.875
	5/3	1.875	1.375
	9/7	3.5	2.375
Secker et Taubman	Haar	1.75	1.75
	5/3	3.75	2.75
	9/7	7	4.75
MCLIFT	5/3 tronqué		1.55

TAB. 2.1 – *Nombre de compensations en mouvement (par images) par méthodes et par filtres pour une transformée temporelle sur trois niveaux*

requiert l'utilisation du schéma lifting afin d'assurer la reconstruction parfaite. Les sous-bandes temporelles obtenues sont ensuite décomposées spatialement à l'aide d'un filtre de Daubechies 9/7, puis les sous-bandes spatio-temporelles sont codées à l'aide d'un codeur SPIHT.
Afin de réduire la distance temporelle entre les images filtrées à des niveaux de résolution basse, les auteurs choisissent de modifier le sens de la compensation en mouvement et d'alterner entre une compensation avant et une compensation arrière. La comparaison entre les deux modes de compensation en mouvement est montrée sur la figure 2.8. On voit que pour les deux plus bas niveaux de résolution temporelle, la distance entre images filtrées est réduite dans le mode alterné par rapport au mode compensation avant uniquement. La réduction de la distance entre images filtrées permet d'améliorer la décorrélation temporelle car la redondance temporelle entre deux images est d'autant plus importante que les images sont proches. Cependant l'utilisation de filtres courts comme le filtre de Haar limite l'intérêt des ondelettes pour la décorrélation temporelle.

2.1.4.3 Influence de la taille des filtres sur la complexité opératoire

La taille des filtres influe sur la complexité opératoire de la transformée temporelle. Le tableau 2.1 montre le nombre de compensations en mouvement nécessaires pour les schémas de Ohm [Ohm 94], Secker et Taubman [Secker 01] et du MCLIFT [Luo 01] en version classique et lifting avec les filtres suivants: Haar, 5/3 et 9/7. On remarque que pour les filtres 5/3 et 9/7, la version lifting est moins complexe que la version classique, tout en assurant une reconstruction parfaite du signal et la gestion des erreurs d'arrondis. La méthode de Secker et Taubman est plus complexe en terme de nombre de compensation en mouvement que celle de Ohm car elle utilise un champ bidirectionnel, cependant la technique de Ohm doit gérer le problème des pixels déconnectés ou multiplement connectés.

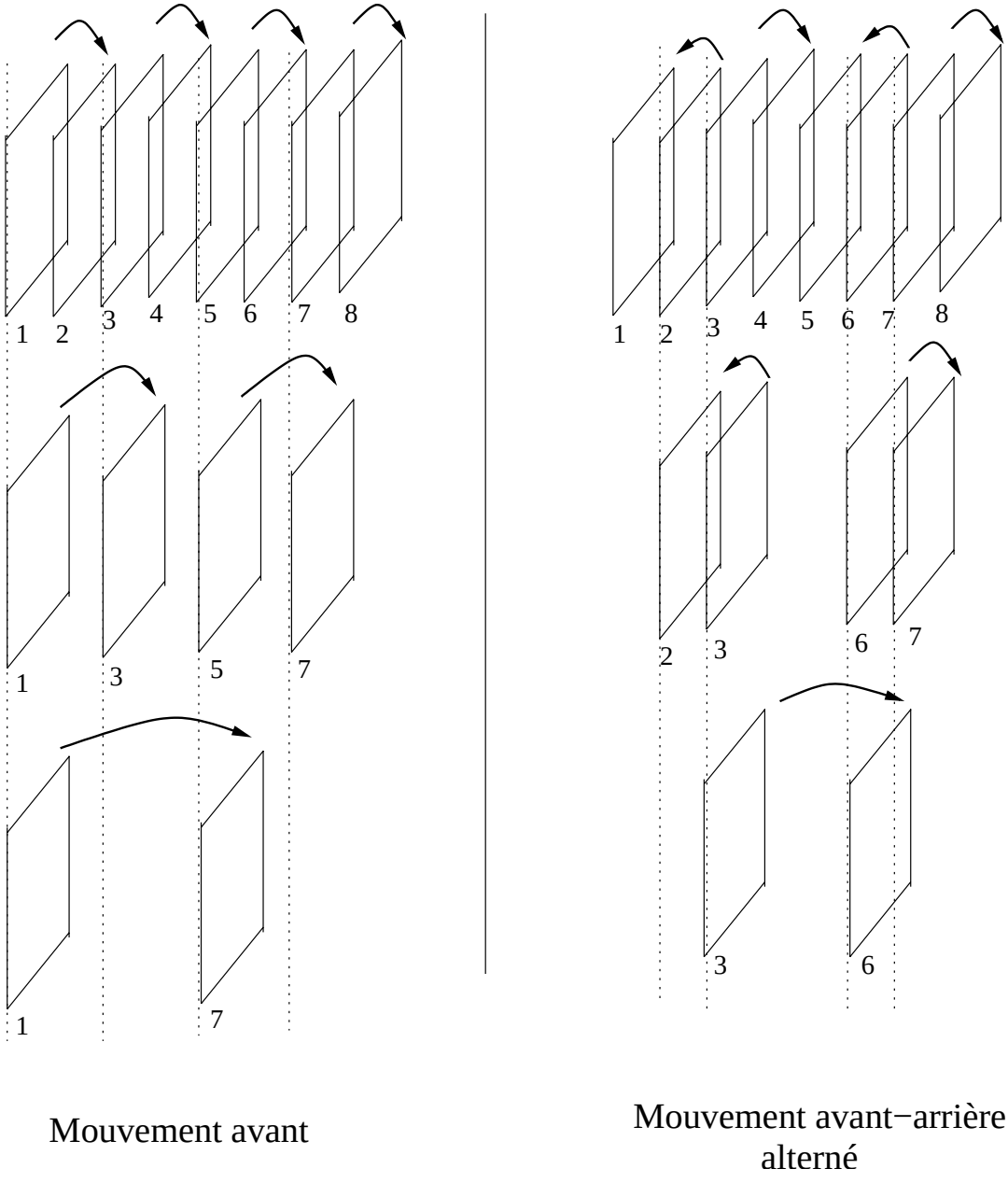


FIG. 2.8 – *Transformation ondelette temporelle sur huit images par filtre de Haar: mouvement avant et mouvement avant-arrière alterné*

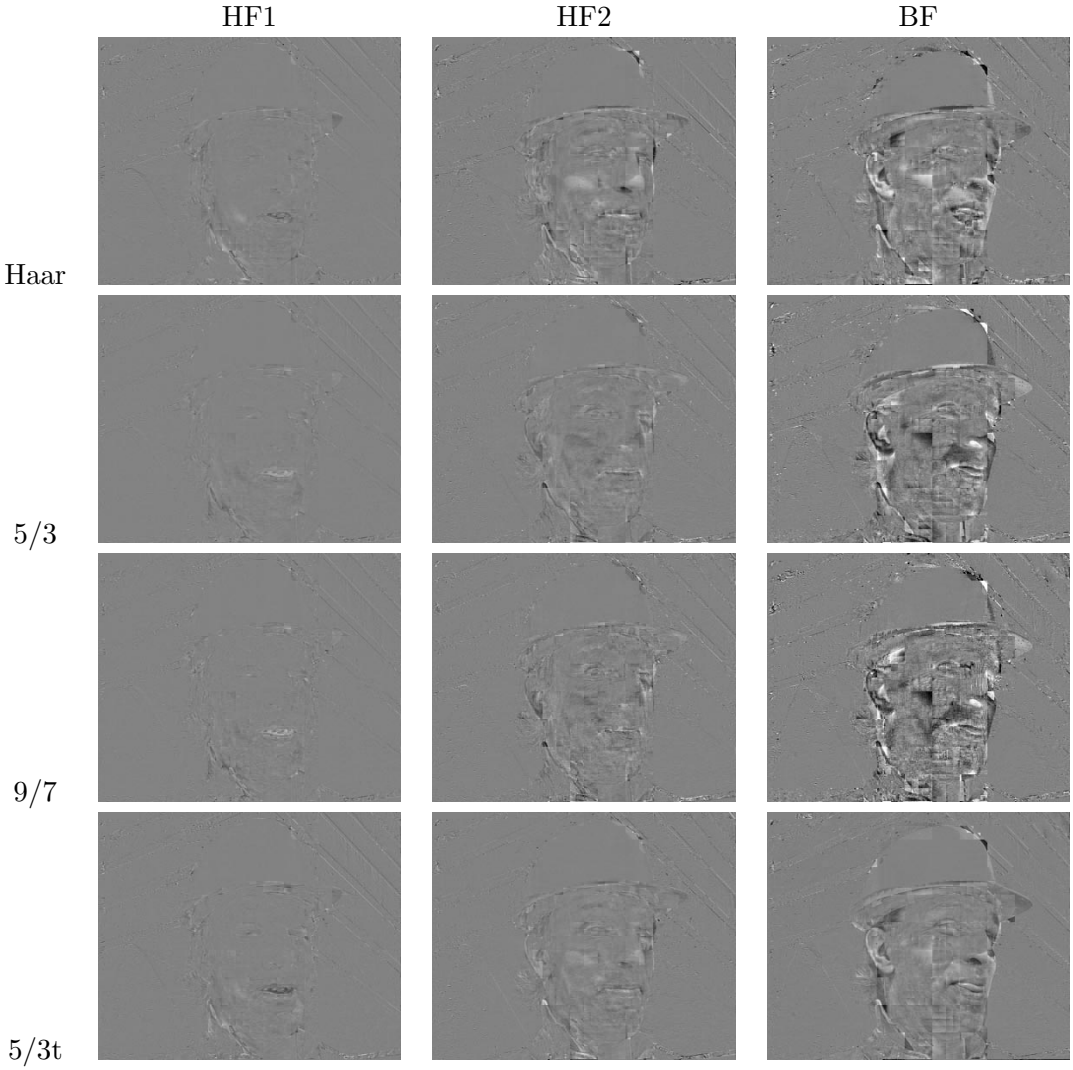


FIG. 2.9 – *Sous-bandes obtenues par transformation ondelettes par les filtres Haar, 5/3, 9/7 et 5/3 tronqué (5/3t)*

2.1.4.4 Influence de la taille des filtres sur la qualité des sous-bandes temporelles

La figure 2.9 montre les sous-bandes obtenues par transformation ondelettes avec les filtres de Haar, 5/3, 5/3 tronqué et 9/7 par la méthode de Secker et Taubman (mouvement bidirectionnel) et un mouvement par blocs. La décorrélation dépend de la longueur du filtre utilisé pour la transformée temporelle, en théorie plus le filtre est long, plus l'information est décorrélée. Cependant, dans le cas de la transformée temporelle avec compensation en mouvement, la qualité du mouvement estimé influe fortement sur le pouvoir de décorrélation du filtre. En effet, le filtre le plus long (ici 9/7) qui permettrait une meilleure décorrélation de l'information que les autres filtres ne donnent pas les résultats attendus. Au niveau des hautes fréquences, les filtres 5/3 et 5/3 tronqué donnent les sous-bandes hautes fréquences de plus faibles énergies, pour les images basses fréquences, le 5/3 tronqué fournit les sous-bandes les plus faibles en énergie.

2.1.5 Transformée temporelle et discontinuité du mouvement

Cette section présente des schémas de transformée temporelle qui exploitent le mouvement mais en tenant compte des discontinuités temporelles du mouvement (occlusion, ...) et des aspects perceptuels.

2.1.5.1 Lignes de mouvement et transformée temporelle basée objet

Dans [Xu 00], les auteurs proposent un schéma de codage vidéo scalable basé objet. Une transformée temporelle est appliquée le long de la trajectoire de mouvement des objets vidéo, figure 2.10. Un mouvement par blocs est utilisé, et chaque pixel de la première image est suivi sur sa trajectoire. Un pixel de l'objet à l'instant t est classé selon qu'il: commence, continue, finit une trajectoire ou entre en collision avec une autre trajectoire. Cette classification est présentée sur la figure 2.11. Elle permet de définir les lignes de mouvement sur lesquelles va être appliquée la transformée temporelle. La transformée est appliquée sur chaque ligne de mouvement considérée comme un signal 1D. Les sous-bandes temporelles obtenues sont ensuite décomposées à l'aide d'une transformation ondelettes 2D basée objet et les sous-bandes résultantes sont codées par un codeur 3D-ESCOT. Le codeur 3D-ESCOT est une extension du codeur EBCOT utilisé pour le codage de sous-bandes spatiales dans la norme JPEG-2000. Le codeur a été étendu au codage de sous-bandes spatio-temporelles en prenant en compte le mouvement et en considérant un contexte de codage 3D.
La technique de transformation temporelle le long des lignes de mouvement a été étendue à un schéma non objet dans [Luo 03]. La transformée temporelle est appliquée à l'aide d'un schéma lifting et d'un filtre 5/3 et la compensation en mouvement est bidirectionnelle. L'intérêt ici des lignes de mouvement est que chaque pixel est connecté de chaque côté. Les pixels multiplement connectés sont filtrés suivant la première ligne de mouvement rencontrée et les pixels connectant un pixel multiplement connecté sont filtrés avec ce pixel que ce pixel ait été filtré avec lui ou non. Ceci permet de ne pas

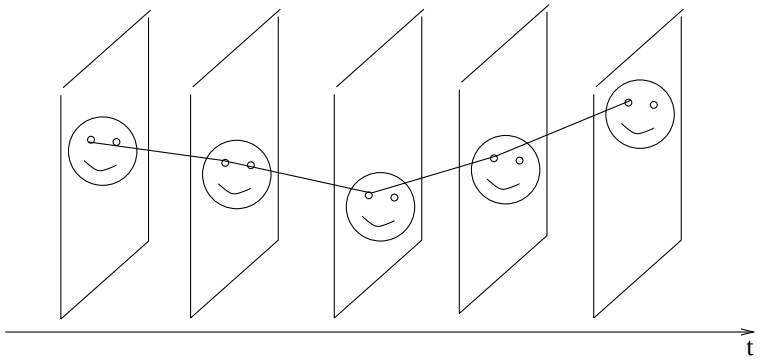


FIG. 2.10 – *Exemple de ligne de mouvement sur un objet vidéo*

créer de petites trajectoires lors de collision de pixels. De même, les pixels isolés sont filtrés en utilisant le mouvement d'une ligne adjacente. Cependant, le fait de toujours connecter les pixels peut parfois nuire à la décorrélation par ondelettes. En effet, parfois, les déconnections sont dues à l'apparition d'occlusion dans la scène, le fait de connecter des pixels qui ne devraient pas l'être, introduit de l'énergie dans les hautes fréquences et impacte sur le coût de codage.

La notion de lignes de mouvement est également présentée dans [Golwelkar 03]. Un mouvement unidirectionnel arrière est estimé, puis à partir du mouvement estimé un mouvement bidirectionnel est généré par inversion du mouvement arrière. Le long des trajectoires de mouvement, les pixels sont classés selon qu'ils marquent le début, la fin ou la continuation d'une trajectoire ou bien qu'ils sont isolés. Cette technique permet d'utiliser des filtres longs comme le filtre 5/3 et l'inversion du champ de mouvement arrière permet d'utiliser un champ bidirectionnel sans augmenter le coût de mouvement. La transformée temporelle est appliquée sous forme lifting afin de préserver la réversibilité du schéma dans le cas de mouvements sous-pixelliques. La transformation est effectuée sur les lignes de mouvement en appliquant un miroir pour les pixels aux bords des trajectoires. Les pixels isolés ne sont présents que sur la dernière image du groupe traité, cette image doit être une basse fréquence et les pixels isolés sont filtrés par leur propre valeur. Si des pixels déconnectés apparaissent sur une autre image que la dernière, ils sont classés comme des pixels débutant une nouvelle trajectoire.

2.1.5.2 Optimisation des paramètres de filtrage

Les techniques présentées sont proches des schémas de transformée temporelle classique, elles tentent de les améliorer en prenant en compte l'aspect perceptuel de la vidéo et notamment les discontinuités du mouvement dues aux occlusions. Pour cela, elles adaptent les techniques mises en œuvre dans les schémas de codage prédictif permettant d'améliorer la qualité de la prédiction (blocs de taille variable, références

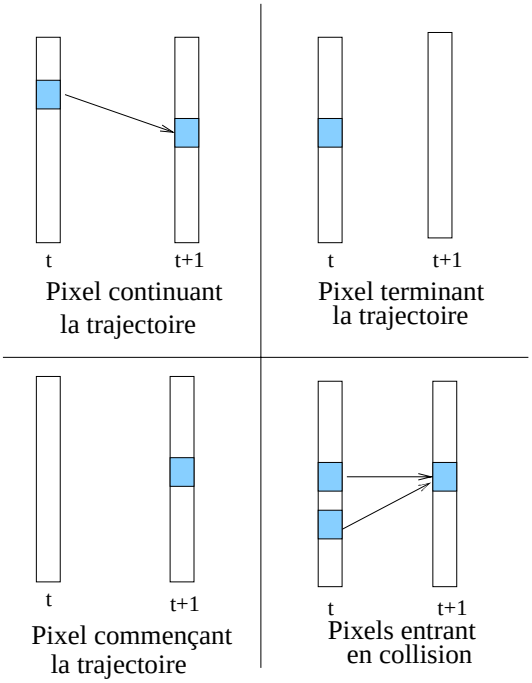


FIG. 2.11 – *Classification des pixels sur les lignes de mouvement*

multiples, mode Intra).
Dans [Xiong 04], les auteurs améliorent le schéma proposé dans [Luo 03] en utilisant des blocs de taille variable pour la représentation du mouvement. L'utilisation de blocs de taille variable permet de diminuer l'apparition de pixels non connectés mais augmente l'effet de blocs dans les sous-bandes hautes fréquences. Les auteurs proposent alors l'utilisation d'une compensation en mouvement recouvrante (OBMC: overlapped block-motion compensation), un pixel n'est plus prédit par un seul pixel de l'image référence mais par une combinaison pondérée du pixel de l'image référence et de ses voisins.
Dans [Turaga 02], les auteurs proposent une amélioration des schémas classiques par transformée temporelle avec compensation en mouvement. L'amélioration porte sur l'adaptation de la prédiction par multiples références introduites dans le codeur H264 au cas du filtrage temporel. Le mouvement est représenté comme dans [Xiong 04] par des blocs de taille variable qui sont en correspondance avec des blocs pouvant appartenir à plusieurs images références, figure 2.12. Cette technique permet de diminuer l'apparition de pixels non connectés et de diminuer l'amplitude des hautes fréquences en filtrant un signal plus corrélé.
L'obtention des hautes fréquences se fait par filtrage des blocs avec leur correspondant dans l'image référence, les basses fréquences sont représentées par les images références originales inchangées, figure 2.13. Le filtrage temporel sur plusieurs niveaux de résolution est appliqué classiquement sur les images non filtrées. L'utilisation d'un

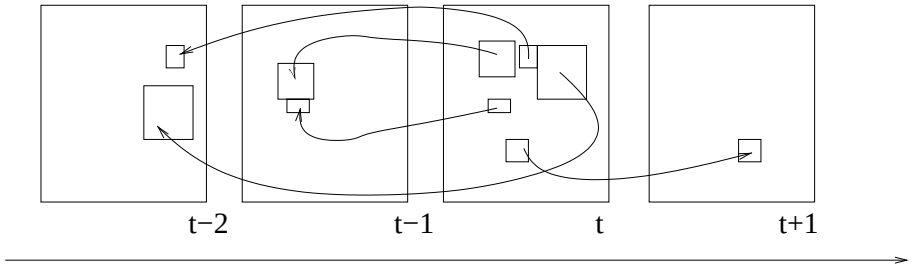


FIG. 2.12 – *Estimation de mouvement avec références multiples*

mouvement bidirectionnel permet d'utiliser des images références passées et futures pour la prédiction et ainsi de prendre en compte les objets entrant et sortant de la vidéo.
Les images références représentant les basses fréquences sont positionnées de manière arbitraire dans le temps, ceci permet une scalabilité temporelle arbitraire.

Dans [Chen 03a], la transformée temporelle avec compensation en mouvement est appliquée selon le schéma de [Choi 99]. En fonction de l'état de connexion d'un bloc, celui-ci peut-être filtré temporellement si la connexion à son correspondant permet une bonne prédiction, ou codé en mode Intra si le bloc appartient à une région découverte (échec de l'estimation mouvement) ou si la connection à son correspondant produit une mauvaise prédiction et donc une forte amplitude en hautes fréquences.

2.1.6 Discontinuités aux frontières du signal

Cette section traite du problème engendré par la nature finie du signal à transformer. Les filtres utilisés dans la transformée temporelle sont à support infini, or le signal filtré est lui à support fini. Des problèmes interviennent lors du filtrage de coefficients situés aux frontières du signal. Ce problème apparaît dans tous les schémas de transformée traitant les images par groupe et également dans les schémas exploitant les lignes de mouvement définies plus haut, où le problème se pose à chaque extrémité de la ligne.
Dans le cas des lignes de mouvement, une solution simple consiste à ne pas filtrer le coefficient et à le laisser tel quel dans la sous-bande. Cette solution est acceptable dans le cas d'un coefficient situé sur une basse fréquence car le signal basse fréquence est proche du signal original, une mise à échelle du coefficient est cependant souhaitable selon la résolution de la basse fréquence. Mais cette solution est inenvisageable dans le cas où le coefficient est sur une haute fréquence. Le signal haute fréquence doit avoir une faible amplitude, laisser la valeur du coefficient dans la haute fréquence entraînerait une trop forte amplitude difficile à coder. Mettre le coefficient à zéro dans la haute fréquence n'est pas non plus envisageable car la transformée ne serait alors plus réversible.
Les solutions le plus souvent utilisées étendent le signal afin de simuler un signal infini. Le signal peut être étendu par répétition périodique de lui-même ou par symétrie par

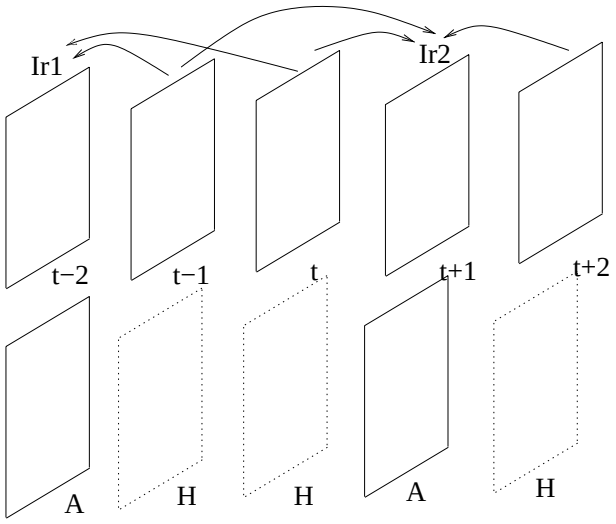


FIG. 2.13 – *Transformée temporelle avec références multiples*

rapport aux bords. Les coefficients aux frontières du signal peuvent alors être filtrés de la même manière que les autres coefficients. Une autre approche est de modifier le filtre appliqué aux bord du signal, cependant cette technique est assez coûteuse et complexe car le filtre modifié doit être orthogonal et normalisé.
La méthode des lignes de mouvement adresse le problème des pixels multiplement connectés ou non connectés. On remarque que le problème de connection des pixels est analogue au problème des frontières du signal. En effet, un pixel non connecté d'un côté est considéré comme un pixel en frontière d'un signal (ligne de mouvement). Dans [Zhan 02], ce problème est adressé sans utiliser la notion de lignes de mouvement et donc en évitant les inconvénients liés à cette technique. Les auteurs proposent un schéma de transformée temporelle lifting avec mouvement par blocs et filtre de transformée long. Le mouvement est bidirectionnel ainsi les pixels des hautes fréquences sont toujours connectés. Pour le calcul des basses fréquences, la phase de mise à jour du lifting est adaptative, les coefficients du filtre s'adaptent en fonction de la connexion du pixel. Si le pixel est isolé, le coefficient prend une valeur filtrée du pixel. Si le pixel n'est connecté que d'un côté, une symétrisation du signal est effectuée avec pour centre de symétrie le pixel filtré (symétrie miroir). Dans le cas où le pixel est connecté à plusieurs pixels, le meilleur pixel est choisi pour le filtrage à partir de la minimisation d'un critère qui peut être basé sur la minimisation de la DFD, ou la minimisation de la norme du vecteur mouvement, ou autre. Cette technique permet d'adapter le filtrage aux endroits où l'estimation mouvement a échoué, typiquement aux pixels non connectés

ou multiplement connectés.
Enfin, nous adressons dans ce paragraphe le problème des bords de GOFs (Group Of Frames). Dans la plupart des schémas, les images sont traitées par groupes et le problème des bords se posent comme pour les frontières d'un signal. La solution la plus retenue est de symétriser le signal aux bords afin de traiter les images extrémités. Certains schémas traitent les images au flot [Zhan 02, Chen 03b] pour des raisons de délai au décodage et afin d'éviter les problèmes aux bords. Le traitement des images au flot réduit le délai d'attente au décodage mais interdit la possibilité d'accès aléatoire dans le flux.

2.2 Le mouvement

2.2.1 Influence de la qualité du mouvement

Nous avons vu dans les sections précédentes que selon le type de compensation en mouvement choisi, la taille des filtres de transformée temporelle variait d'un schéma à l'autre. En théorie, la transformation ondelettes d'un signal est à support infini et la décorrélation des informations est d'autant plus importante que le support du filtre de transformée est long. Cependant, en pratique, de par la nature finie des données à traiter, la transformée temporelle appliquée dans les schémas de codage vidéo est faite sur un support fini. De plus, la présence de mouvement dans la séquence vidéo et la qualité de ce mouvement utilisé pour la transformée temporelle influent sur la taille des filtres de la transformée. Dans [Vieron 02], les auteurs montrent que plus le mouvement utilisé dans la transformée temporelle est bon, moindre est l'énergie contenue dans les sous-bandes hautes fréquences. La figure 2.14 montre les sous-bandes obtenues avec une transformation temporelle ondelettes par la méthode de Secker et Taubman et les filtres de transformée Haar, 5/3, 5/3 tronqué et 9/7 et un mouvement contraint en débit à 35% du débit total. On remarque que lorsque le mouvement est contraint en débit, les filtres courts permettent de réduire l'énergie des sous-bandes par rapport aux filtres longs. En effet, ce constat a été introduit dans la section 2.1.4.4 où les sous-bandes montrées sont obtenues avec un mouvement non contraint. La qualité du mouvement influe d'autant plus sur la qualité des sous-bandes temporelles que le mouvement est contraint en débit.

2.2.2 Influence du type de mouvement

La qualité des sous-bandes est aussi liée au type de mouvement utilisé. Les sous-bandes de la figure 2.14 ont été obtenues avec l'utilisation d'une représentation du mouvement par blocs, et on peut remarquer sur les sous-bandes des effets de blocs. Le mouvement par blocs est discontinu aux frontières des blocs et entraîne l'apparition de hautes fréquences artificielles dans les sous-bandes. Ces hautes fréquences sont dites artificielles dans le sens où elles ne résultent pas des discontinuités du mouvement propre dans la séquence mais de la représentation même du mouvement. Elles sont introduites par la représentation par blocs. La représentation du mouvement par maillage assure

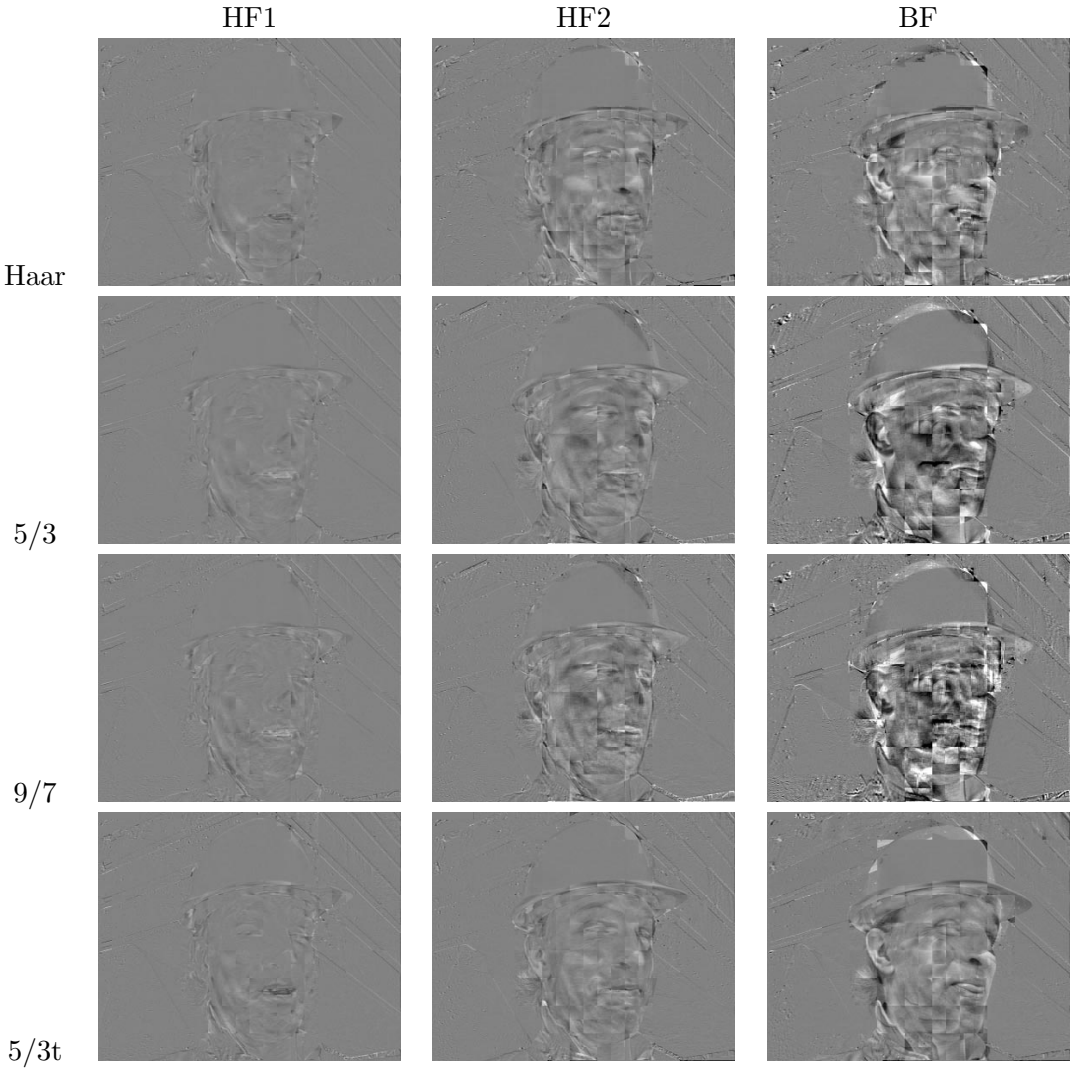


FIG. 2.14 – *Sous-bandes obtenues par transformation ondelettes sur trois niveaux de décomposition avec mouvement contraint à 35% du débit total pour les filtres Haar, 5/3, 9/7 et 5/3 tronqué (5/3t)*

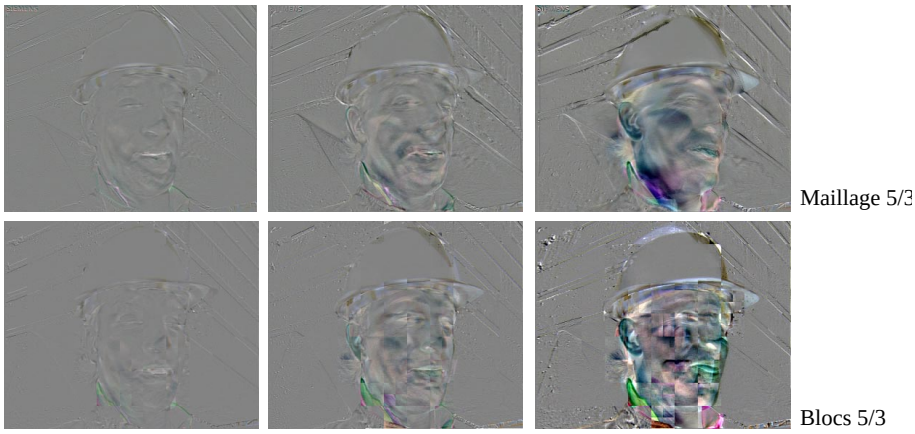


FIG. 2.15 – *Sous-bandes obtenues par transformation ondelettes avec mouvement par maillage et par blocs (mouvement contraint à 35% du débit total), filtre 5/3*

une continuité de la texture au cours du temps. Les sous-bandes temporelles résultantes ne présentent pas de hautes fréquences introduites par cette représentation. La figure 2.15 compare les sous-bandes obtenues par transformation temporelle avec filtre 5/3 pour un mouvement par maillage et un mouvement par blocs. On observe que les sous-bandes du maillage sont plus lisses et présentent moins de pics hautes fréquences que les sous-bandes par blocs. Les sous-bandes par maillage sont a priori plus faciles à coder que les sous-bandes issues d'un mouvement par blocs.

La supériorité des maillages sur les blocs pour un codage vidéo par ondelettes est aussi montrée dans [Bozinovic 04]. Les auteurs comparent les performances d'un codage vidéo par ondelettes 2D+t avec mouvement par blocs et mouvement par maillages. Les images sont transformées à l'aide d'un filtre 5/3 lifting sur deux niveaux de décomposition, puis les sous-bandes temporelles sont codées par un codeur JPEG-2000. Le mouvement est codé sans pertes par un codeur JPEG-LS. Bien que dans leur méthode la technique avec mouvement par maillages impliquent un coût de mouvement plus important que le mouvement par blocs, les performances en PSNR sont meilleures pour la technique par maillages que pour la technique par blocs. Dans [Secker 02] présenté précédemment, les résultats vont également en faveur du mouvement par maillages. Dans ce schéma, les maillages permettent deux améliorations importantes: la première concerne la meilleure qualité des sous-bandes temporelles à coder (figure 2.15), la deuxième concerne la réduction du nombre de champ de mouvement à transmettre par rapport au schéma par blocs.

2.2.3 Réversibilité du champ de mouvement

La réversibilité du mouvement dans le schéma de transformée temporelle n'est pas automatique selon la direction et le type de mouvement utilisés. Le mouvement

modélisé par un maillage est directement inversible. En effet, le champ de mouvement par maillage est bijectif et le déplacement d'un point est calculé par rapport aux déplacements des nœuds du maillage.
Pour une représentation du mouvement par blocs, le champ de mouvement n'est pas bijectif, d'où l'apparition des pixels non connectés ou multiplement connectés dans les schémas de transformée avec un seul champ de mouvement. Dans le cas de deux champs de mouvement (un champ de mouvement avant et un champ de mouvement arrière), les pixels déconnectés ou multiplement connectés n'apparaissent pas.
Dans les schémas de [Ohm 94], [Chen 03b] ou [Golwelkar 03], un seul champ de mouvement est utilisé. Dans ces schémas, l'inversion du mouvement est nécessaire pour gérer les pixels non connectés et/ou si une précision sous-pixellique est utilisée. L'inversion d'un vecteur mouvement b est donné par: $f = -\overline{b}$, où $\overline{b}$ est l'arrondi au plus proche entier de b. Dans le schéma MCLIFT [Luo 01], l'inversion du mouvement n'est pas nécessaire car seule l'étape de prédiction du schéma lifting est appliquée et le schéma ne nécessite pas le mouvement inverse.
Dans [Xu 04], le champ de mouvement n'est pas inversé bien qu'un seul champ de mouvement soit utilisé pour la transformée temporelle lifting. L'étape de prédiction du lifting s'effectue de manière classique, pour l'étape de mise à jour, chaque pixel est mis à jour à l'aide de tous les pixels dont il a contribué à la prédiction, cette technique est expliquée par la figure 2.16.

2.2.4 Précision du mouvement et réversibilité de la transformée

La précision du mouvement influe sur la qualité de la décorrélation de la transformée temporelle ainsi que sur la réversibilité du schéma. Dans les premiers schémas de Ohm [Ohm 94] et Choi et Woods [Choi 99], la transformée n'est pas réversible dans le cas d'une précision du mouvement sous-pixellique à cause de l'étape d'interpolation qui n'est pas inversible. Dans [Ohm 94], l'auteur montre qu'une interpolation bilinéaire dans la transformée temporelle provoque un flou du au filtrage passe-bas et préfère utiliser une interpolation plus précise dans le domaine DCT, cette dernière est cependant assez coûteuse. Dans [Hsiang 02], l'auteur propose une transformée temporelle réversible avec précision au demi-pixel du mouvement. Le mouvement est représenté par blocs et pour deux blocs connectés, un bloc composite est construit par fusion des deux blocs résultant en un bloc de résolution deux fois supérieure aux blocs originaux. La technique est similaire à un entrelacement de deux signaux. Le filtrage temporel est appliqué comme un filtrage 1D spatial dans le bloc composite, les basses et hautes fréquences sont placées dans les blocs originaux correspondants. Cette technique permet une transformée temporelle réversible avec précision sous-pixellique car l'interpolation est la même à l'analyse et à la synthèse, cependant cette technique devient d'autant plus coûteuse que la précision augmente.
Une technique de transformation temporelle permettant d'assurer la réversibilité de la transformée quelle que soit la technique d'interpolation est l'utilisation d'une transformée lifting. Dans [Chen 03b], une amélioration de [Choi 99] est proposée par l'introduction du lifting. Le lifting assure la réversibilité de la transformée dans le cas d'une

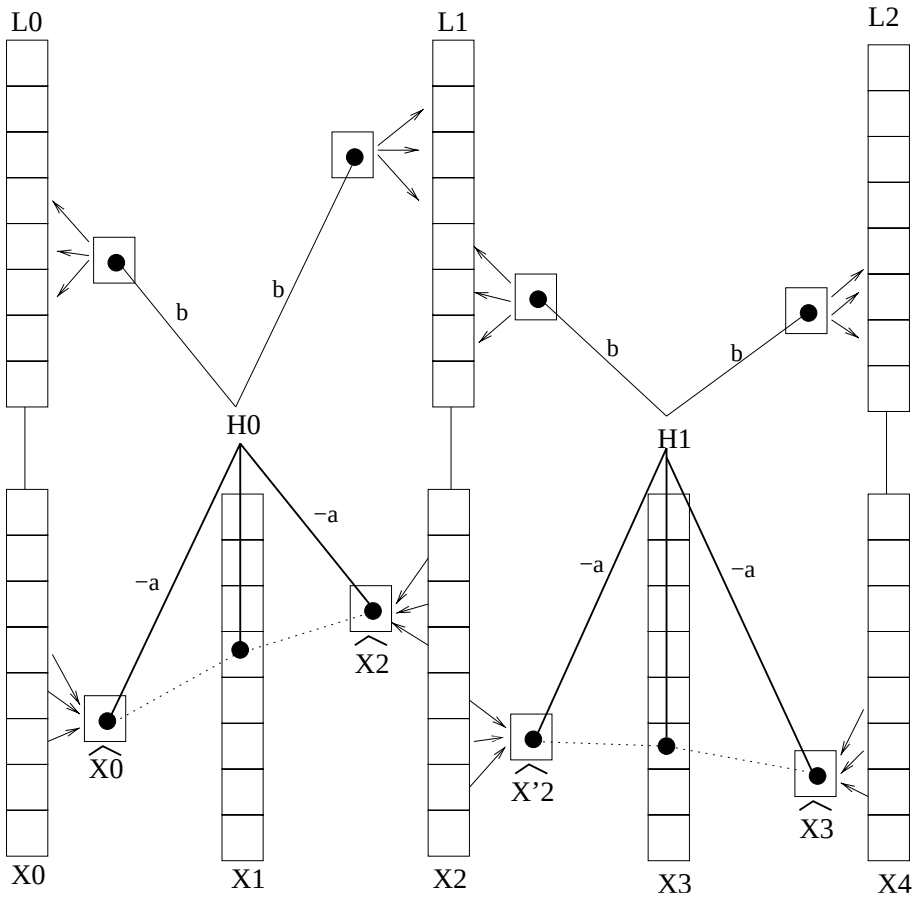


FIG. 2.16 – *Prédiction et mise à jour du schéma lifting sans inversion du champ de mouvement*

précision sous-pixellique car l'interpolation est la même au codeur et au décodeur, de plus le schéma fonctionne quelle que soit la précision et l'interpolation utilisées. Le lifting est utilisé dans la plupart des schémas par transformée temporelle [Secker 01, Chen 03b, Luo 01, Golwelkar 03, Vieron 02, Bottreau 01, Chen 02].
Comme dans les shémas de codage prédictif classique, dans les schémas par transformée, la précision du mouvement influe aussi sur les performances en compression du codeur. Dans [Chen 03b], les auteurs montrent une amélioration des performances en compression de 2dB à 2Mps entre le demi-pixel et le quart-pixel. Le gain en performance grâce à l'augmentation de la précision du mouvement est confirmé par les résultats de [Bottreau 02]. La précision au demi-pixel est meilleure que la précision au pixel pour les hauts débits, à bas débits, la précision demi-pixel n'apporte pas de gain à cause du coût plus élevé du mouvement sous-pixellique. Cependant, on remarque que dans [Golwelkar 03], dans le cas d'un mouvement sous-pixellique la construction des lignes de mouvement le long des trajectoires est approximatif, alors qu'il est exact dans le cas de mouvement pixellique.

2.2.5 Scalabilité du mouvement

Dans le cadre d'un codage vidéo complètement scalable, toutes les informations doivent être codées de manière scalable. La qualité et la précision du mouvement doivent être adaptés aux caractéristiques de décodage au niveau fréquence temporelle, résolution spatiale et distorsion visuelle. Par exemple pour un décodage à une basse fréquence temporelle, si les hautes fréquences de texture ne sont pas décodées, il est inutile de décoder les informations de mouvement relatives à ces hautes fréquences. De même, pour un décodage à une basse résolution spatiale, la précision du mouvement doit être appropriée à la résolution spatiale. Pour cela, un codage scalable du mouvement est nécessaire.
Un tel codage est présenté dans [Bottreau 01]. Le mouvement est représenté de manière hiérarchique comme pour la texture. La scalabilité temporelle est obtenue en plaçant dans le flux binaire les informations codées de mouvement concernant les hautes fréquences de texture avec ces dernières. L'adaptation à la résolution spatiale des vecteurs mouvement se fait par simple division par deux de ces vecteurs pour obtenir une résolution plus basse, jusqu'au niveau de base. Les vecteurs résultants du niveau de base sont codés par une technique DPCM et un codage entropique par VLC. Pour les niveaux supérieurs, un raffinement est codé par rapport au niveau juste inférieur par un codage arithmétique contextuel.
Une approche différente pour le codage scalable des vecteurs mouvement est proposée dans [Taubman 03]. Le schéma de codage vidéo est semblable à celui proposé dans [Secker 02]. Dans le schéma proposé, le champ de mouvement par maillage est codé de manière scalable à l'aide d'une transformation ondelette temporelle et spatiale. La transformée temporelle est appliquée à l'aide d'un filtre de Haar, puis la transformée spatiale est appliquée avec un filtre 5/3 sur deux niveaux de décomposition. Les sous-bandes de mouvement sont ensuite codées par plan de bits en couches de qualité par

une technique similaire à celle employée dans le codeur JPEG-2000 [Chrysafis 99].
Dans le schéma proposé, les images de la séquence sont transformées à l'aide du mouvement original non compressé, mais au décodage, les images sont synthétisées à l'aide du mouvement décodé au débit approprié. Le codage scalable du mouvement peut entraîner une distorsion sur le signal reconstruit. Cette distorsion est due au fait que la reconstruction du signal utilise un mouvement différent de celui utilisé au codage. Cependant, le support de la transformée temporelle étant fini, la distorsion introduite par le mouvement est elle aussi finie. Dans le cas de faibles erreurs de mouvement, les auteurs montrent que la distorsion du signal reconstruit varie linérairement avec la distorsion du mouvement, ceci permet une optimisation dans la répartition des débits entre les informations de mouvement et de texture après compression.
Les auteurs étudient l'impact de la transformation ondelette sur la qualité de la séquence vidéo après décodage du mouvement à différents débits. Les images de la séquence vidéo sont compressées sans pertes, la distorsion de la séquence reconstruite est seulement due au mouvement décodé. Les auteurs comparent la distorsion du signal reconstruit avec un mouvement codé avec pertes sans transformation ondelettes et celle obtenue avec un mouvement codé avec pertes avec transformation ondelettes spatio-temporelles. Ils montrent que la distorsion du signal reconstruit est moindre dans le cas où les informations de mouvement ont été transformées en ondelettes préalablement au codage progressif, cette différence est accentuée pour les bas débits.

2.3 Transformation spatiale

Dans la plupart des schémas de codage vidéo par ondelettes 2D+t, la transformation temporelle est suivie de la transformation spatiale des sous-bandes temporelles. Dans ces shémas, le filtre de décomposition le plus utilisé est le Daubechies 9/7.
Cependant, quelques schémas effectuent la transformation spatiale préalablement à la transformée temporelle. Dans ce cas, la transformée temporelle ne s'applique plus sur les images de la séquence vidéo mais sur des sous-bandes ondelettes spatiales. Dans cette section, nous allons étudier les schémas appliquant d'abord la transformée spatiale, puis nous verrons une application de l'utilisation de la transformée spatiale préalablement à une décomposition t+2D et nous terminerons par l'étude de la décomposition spatiale dans le cadre d'un codage vidéo basé objet.

2.3.1 Transformée spatiale préalable

Dans [Reichel 03], les auteurs proposent un schéma de codage vidéo mettant en œuvre une transformée spatiale ondelette des images de la séquence vidéo. Les images de sous-bandes spatiales sont ensuite décomposées par une transformée temporelle avec un filtre de Haar. Le mouvement n'est pas pris en compte dans la transformée temporelle. Ce schéma permet de bons résultats dans le cadre particulier d'applications bien définies. Une application type est la vidéo-surveillance, dans ce cas, la séquence vidéo

est pratiquement fixe. Avec un mouvement global de faible amplitude et peu de mouvements dans la séquence vidéo, la technique présentée est intéressante. Cependant, le fait de ne pas utiliser le mouvement dans la transformée temporelle nuit à la généricité du schéma.
La plupart des schémas mettant en œuvre la transformée spatiale sur les images originales de la séquence et utilisant le mouvement pour exploiter la redondance temporelle se rapprochent des approches classiques des schémas prédictifs. Dans [Andreopoulos 02], les auteurs décomposent les images de la séquence vidéo par une transformation ondelette spatiale. L'estimation de mouvement est effectuée dans le domaine de la transformée spatiale, puis les sous-bandes spatiales sont prédites et l'erreur de prédiction est codée par un codeur entropique. L'estimation de mouvement étant effectuée pour chaque sous-bande spatiale, le coût de codage du mouvement est assez élévé. Le schéma n'est vraiment intéressant que pour les hauts débits.
Dans [Andreopoulos 03], les auteurs proposent un schéma de codage vidéo appliquant la transformée temporelle dans le domaine de la transformée ondelette spatiale. Afin d'avoir une estimation précise du mouvement, les sous-bandes spatiales obtenues par décomposition par un filtre Daubechies 9/7 sont utilisées pour construire une transformée spatiale ondelette redondante pour chaque résolution spatiale. L'estimation et la compensation de mouvement sont effectuées dans ce domaine ainsi que la transformée temporelle qui est appliquée à toutes les sous-bandes spatiales. L'estimation de mouvement est similaire aux approches classiques par blocs avec taille des blocs variable et multiples images références. La comparaison de ce schéma et du schéma avec transformée temporelle classique montre de meilleures performances pour le schéma avec transformée spatiale redondante. Cependant, la transformée spatiale redondante implique une plus grande complexité et l'estimation du mouvement pour chaque niveau spatial implique un surcoût de codage des vecteurs mouvement.

2.3.2 Débruitage de la séquence vidéo

Dans [Chen 02], une transformation spatiale sur un niveau de résolution est appliquée à la séquence vidéo afin d'éliminer le bruit présent dans celle-ci. La décomposition spatiale permet d'améliorer l'estimation du mouvement dans la séquence vidéo et de réduire le nombre de pixels non connectés dus au bruit d'acquisition de la vidéo. Le schéma de transformée temporelle tel que présenté dans [Hsiang 02] est ensuite appliqué sur les sous-bandes basses fréquences. Les sous-bandes temporelles et les sous-bandes spatiales issues de la première décomposition sont ensuite transformées spatialement et codées par un codeur de sous-bandes EZBC. Afin de bien représenter le mouvement dans la séquence, l'estimation du mouvement se fait par blocs de taille variable et les images sont traitées par groupes de taille variable en fonction de l'activité dans le groupe d'images, un rafraîchissement intra est possible dans le cas où l'estimation du mouvement a échoué. Ce schéma présente de bonnes performances à très hauts débits et vise principalement des applications de vidéo pour le cinéma numérique où la qualité de la vidéo doit être excellente et où les débits autorisés sont aussi très élevés.

2.3.3 Transformation spatiale basée objet

Des schémas de transformation ondelettes t+2D basés objet ont été présentés dans la section précédente. Ces schémas montraient l'adaptation des schémas de transformée temporelle au mode objet mais fournissait également des solutions adaptables aux schémas non objet permettant de gérer le problème des discontinuités de mouvement (transformée temporelle et lignes de mouvement).
En mode objet, les sous-bandes issues de la décomposition temporelle sont incomplètes, les coefficients ondelettes n'ont été calculés que sur la trajectoire de l'objet. Une première solution est de compléter ces sous-bandes temporelles par une technique de *padding* qui consiste à remplir les zones non définies du signal. L'extension du signal doit être la plus proche possible du signal original afin de ne pas introduire trop de hautes fréquences impliquant un surcoût de codage. Les sous-bandes temporelles complètes peuvent ensuite être transformées par une décomposition spatiale 2D classique. la phase délicate de cette solution est l'étape de padding et le choix de la technique d'extension du signal.
Une deuxième solution est d'adapter la transformation spatiale au mode objet comme la transformée temporelle [Han 98]. Cependant, l'adaptation de la transformée spatiale au mode objet repose le problème des discontinuités aux bords dans la décomposition d'un signal à support fini, ce problème doit aussi être géré dans le cas du codage non objet mais la résolution en mode non objet reste moins complexe que pour une transformée sur des objets de forme arbitraire.
De plus, le problème de la validité du support de définition des objets défini par la segmentation peut entraîner un surcoût de codage par rapport à une approche non objet. La segmentation de la séquence vidéo en objets vidéo reste un problème ouvert et il n'existe pas de solutions génériques efficaces qui offrent une segmentation exacte de tous les objets d'une séquence. Ce manque d'exactitude dans la définition d'un objet peut nuire à la bonne décorrélation spatiale des informations de texture de cet objet.

2.4 Codage des sous-bandes spatio-temporelles

Dans cette section, nous nous intéressons aux outils de codage de sous-bandes issues d'une décomposition spatiale ou d'une décomposition spatio-temporelle.

2.4.1 Intérêts d'un flux emboîté ou progressif

Un flux progressif est un flux composé de sous-flux emboîtés les uns dans les autres, un flux de base permet de décoder un signal avec une qualité minimale, chaque sous-flux successif apportant un raffinement au flux qu'il emboîte. Un tel flux permet de décoder un signal à un débit inférieur à celui pour lequel il a été généré avec une distorsion optimale pour le débit de décodage. Un flux progressif est un flux scalable.
La décomposition en ondelettes permet de représenter les informations de manière hiérarchique, cette hiérarchisation doit être conservée lors du codage des coefficients ondelettes. Le flux binaire doit être codé de manière hiérarchique ou encore progressive afin de permettre la scalabilité.

Le flux progressif permet une représentation des informations du niveau grossier vers le niveau fin, offrant une représentation scalable comme nous venons de voir mais cette représentation présente aussi d'autres avantages. La protection des données prioritaires peut être assurée par la construction du flux à partir de plusieurs couches de quantification donnant plusieurs niveaux de priorité aux données contenues dans les couches. Le débit est contrôlé à chaque étape de quantification. Le débit ou la qualité de décodage n'ont pas à être connus au moment du codage, un seul flux est généré au débit maximum et est ensuite tronqué selon les demandes.
Une première façon de construire un flux progressif est d'effectuer successivement des étapes de seuillage des coefficients et à chaque étape de coder l'information de seuillage des coefficients, cette technique est utilisée dans [Shapiro 93] et [Said 96] qui seront présentés dans la sous-section suivante.
La progressivité est également mise en œuvre par le codage par plan de bits. Le principe est le même que la technique de seuillage, les seuillages se faisant à chaque plan de bits des coefficients. Le nombre de plan de bits dépend de la précision des coefficients à coder. Cette technique permet non seulement de générer un flux progressif mais aussi d'augmenter les performances en compression.

2.4.2 Codage inter sous-bandes

Les premiers codeurs de sous-bandes issues d'une décomposition spatiale par ondelettes utilisent la corrélation existant entre des coefficients de sous-bandes successives et de même orientation, figure 2.17. Le coefficient à la résolution n est appelé *parent* et les coefficients similairement orientés et localisés spatialement à la résolution n+1 sont appelés *enfants*. Un coefficient parent et l'ensemble de ses enfants constituent un arbre.

Un des premiers algorithmes de codage de sous-bandes a été présenté dans [Shapiro 93], sous le nom de EZW (Embedded Zerotree Wavelet). Il utilise l'hypothèse que si un coefficient ondelette d'une résolution n est insignifiant par rapport à un seuil T alors les coefficients de même orientation et localisation spatiale à des résolutions plus fines que n sont susceptibles d'être insignifiants par rapport à T. Cette hypothèse, lorsqu'elle est vérifiée, permet de coder l'ensemble des coefficients (parent et enfants) à l'aide d'un seul symbole indiquant que tous les coefficients de l'arbre issu du coefficient parent sont insignifiants par rapport à T. Cette technique de *zerotree* permet de réduire l'entropie des coefficients à coder.
Shapiro utilise en parallèle un codage par approximations successives. Le codeur procède itérativement en appliquant une succession de seuillage aux coefficients et déterminant leur signification par rapport à chaque seuil. A chaque seuil est associé une carte de signification qui est codée par la méthode du zerotree. Le seuil initial T_0 est choisi tel que $T_0 > \frac{x_j}{2}$, avec x_j les coefficients ondelettes. Les seuils successifs sont ensuite obtenus par $T_i = \frac{T_{i-1}}{2}$. Cette technique permet de coder les coefficients ondelettes de manière progressive.
Une amélioration de l'algorithme EZW est donnée dans [Said 96]. L'algorithme SPIHT (Set Partitioning in Hierarchical Trees) considère les coefficients par groupes. Comme

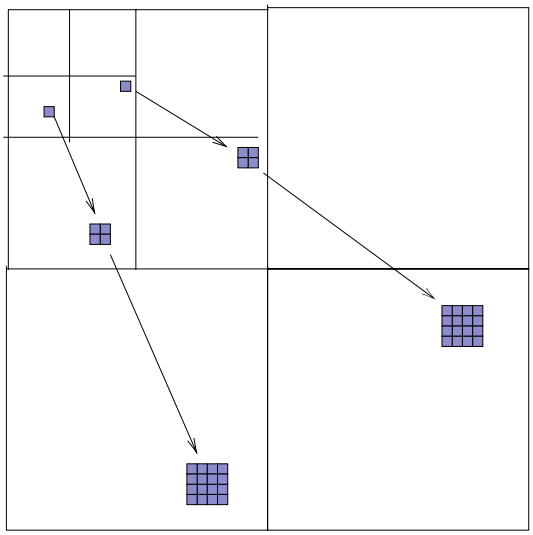


FIG. 2.17 – *Relation entre pixels inter sous-bandes*

EZW, il procède de manière itérative en considérant une succession de seuils. Les seuillages successifs sont appliqués sur les groupes de coefficients et si un des coefficients du groupe est significatif par rapport au seuil, le groupe est divisé de manière récursive et par rapport au seuil jusqu'à ce que les sous-groupes ne contiennent qu'un coefficient significatif. Le signe des coefficients significatifs est codé, puis une passe raffine l'amplitude des coefficients qui étaient significatifs au seuil précédent. Le seuil est diminué et le processus de division itère sur les groupes qui étaient non significatifs. L'algorithme utilise un codage arithmétique pour coder l'information de signification des coefficients. L'algorithme SPIHT améliore les performances de l'algorithme EZW tout en en diminuant la complexité.
L'algorithme SPIHT a été étendu au codage 3D de sous-bandes spatio-temporelles [Kim 00]. Les sous-bandes spatio-temporelles sont obtenues par une transformée temporelle par filtre de Haar avec compensation en mouvement suivi d'une décomposition en ondelettes 2D par filtre biorthogonal 9/7. Les coefficients sont considérés par cube, l'extension des groupes en 2D au cas 3D, l'algorithme de codage fonctionne de manière similaire au cas 2D sur les cubes, cependant l'information de mouvement n'est pas prise en compte pour la formation des cubes sur l'axe temporel.

2.4.3 Codage intra sous-bandes

L'algorithme EBCOT [Taubman 99](Embedded Block Coding with Optimized Truncation) n'utilise pas la corrélation inter sous-bandes comme les algorithmes présentés précédemment, mais utilise la corrélation spatiale des coefficients dans une même sous-bande. Les sous-bandes sont partitionnées en blocs et un flux hautement scalable est généré pour chaque bloc de chaque sous-bande. Chaque flux codé dispose de points de

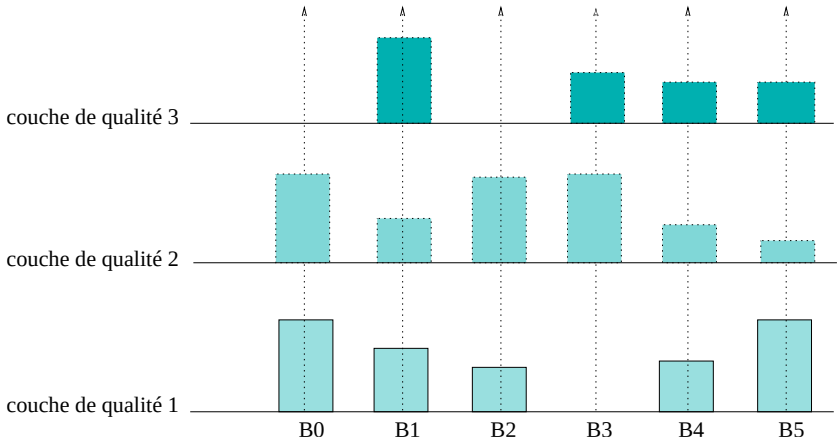


FIG. 2.18 – *EBCOT: contribution de chaque flux aux couches de qualité*

troncature correspondant à un débit et une distorsion donnés. Ces points sont situés sur la courbe de débit-distorsion du bloc. Le train binaire final est organisé en couche de qualité pour obtenir une représentation progressive. Tous les flux précédemment codés sont répartis dans les couches de qualité, la contribution de chaque flux à chaque couche est déterminée par un algorithme PCRD(Post Compression Rate-Distortion), une structuration en couches de qualité est montrée sur la figure 2.18.

Les coefficients d'un bloc sont codés par plan de bits. La technique est la même que pour les algorithmes inter sous-bandes avec l'application de seuillages successifs. Pour chaque plan de bits, tous les coefficients sont examinés, si le coefficient n'est pas significatif pour le plan de bits courant, cette information est codée à l'aide des primitives *zero coding* et *run-length coding*, qui permettent de coder cette information pour un ensemble de coefficients non significatifs adjacents. Si le coefficient devient significatif au plan de bits courant, son signe est codé, s'il était déjà significatif, un raffinement de son amplitude est codé. Les primitives de codage des plans de bits sont codées par codage arithmétique contextuel. Les contextes associés à ces primitives prennent en compte les voisins spatiaux du coefficient (horizontaux, verticaux, diagonaux). L'algorithme de codage progressif EBCOT a été retenu pour le standard de codage d'images fixes JPEG2000 [Chrysafis 99].
Une extension du codage EBCOT au codage de sous-bandes spatio-temporelles est présentée dans [Xu 01]. Les auteurs présentent un schéma de codage vidéo scalable par transformée ondelettes spatio-temporelles. Le schéma de transformée temporelle est le même que dans [Xu 00]. La transformée temporelle est effectuée sur les lignes de mouvement obtenue après une estimation du mouvement par blocs. Le filtre utilisé est le Daubechies 9/7 sous forme lifting. Les sous-bandes temporelles sont ensuite transformées spatialement par le même filtre Daubechies 9/7.
Les sous-bandes spatio-temporelles sont codées par le codeur 3-D ESCOT, extension 3D du codeur EBCOT. Le codeur 3-D ESCOT code chaque sous-bande indépendamment

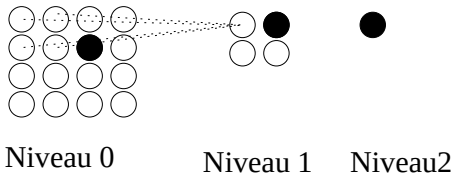


FIG. 2.19 – *EZBC: représentation des sous-bandes en quadtree*

afin d'offrir une grande flexibilité au niveau des scalabilités spatiales et temporelles. De plus, l'optimisation débit-distorsion peut être pré-calculée indépendamment sur chaque sous-bande pour améliorer l'efficacité en compression. Le codage 3-D ESCOT étend le codage arithmétique contextuel 2D du codeur EBCOT à un codage arithmétique contextuel 3D en prenant en compte les voisins temporels le long des trajectoires de mouvement.

2.4.4 Codage hybride: intra et inter sous-bandes

Le codeur de sous-bandes ondelettes EZBC (Embedded ZeroBlocks Coding of subband/wavelet coefficients) est présenté dans [Hsiang 00]. Il associe les deux corrélations utilisées dans les codeurs présentés précédemment. Il utilise d'une part la corrélation inter sous-bandes prise en compte dans les codeurs type EZW ou SPIHT et d'autre part la corrélation spatiale intra sous-bandes utilisée dans EBCOT.
Le codeur EZBC représente chaque sous-bande par un arbre de type *quadtree*. Les nœuds du plus bas niveau de l'arbre correspondent aux coefficients de la sous-bande basse fréquence. Quatre nœuds adjacents deviennent les fils d'un nœud d'un niveau supérieur, les niveaux sont ainsi construits successivement jusqu'à la racine de l'arbre. La représentation en quadtree est montrée sur la figure 2.19. Pour chaque sous-bande, les valeurs des nœuds et des coefficients sont ensuite codées par plan de bits avec des passes successives pour déterminer si les valeurs sont significatives et coder leur valeur le cas échéant. La structure en quadtree est codée efficacement à l'aide d'un codage contextuel. Pour un nœud, le contexte prend en compte les huit voisins spatiaux du nœud ainsi que le nœud dans la sous-bande parente du niveau de quadtree inférieur, la figure 2.20 montre les nœuds utilisés dans le contexte intra et inter sous-bandes.
Le codeur EZBC a lui aussi sa version étendue au codage 3D de sous-bandes spatio-temporelles. Dans [Hsiang 01], les auteurs proposent un schéma de codage vidéo par transformée ondelettes spatio-temporelle. Les sous-bandes résultantes de la transformation sont codées séparément par le codeur EZBC. La représentation par contexte est la même que dans le cas 2D (intra et inter sous-bandes spatiales). L'information de contexte sur l'axe temporel n'est pas utilisée, l'extension au cas 3D n'est ici rien d'autre que le codage de sous-bandes temporelles par le codeur EZBC 2D, les sous-bandes temporelles sont codées indépendamment.
Une comparaison des performances du codeur EZBC par rapport au codeur EBCOT

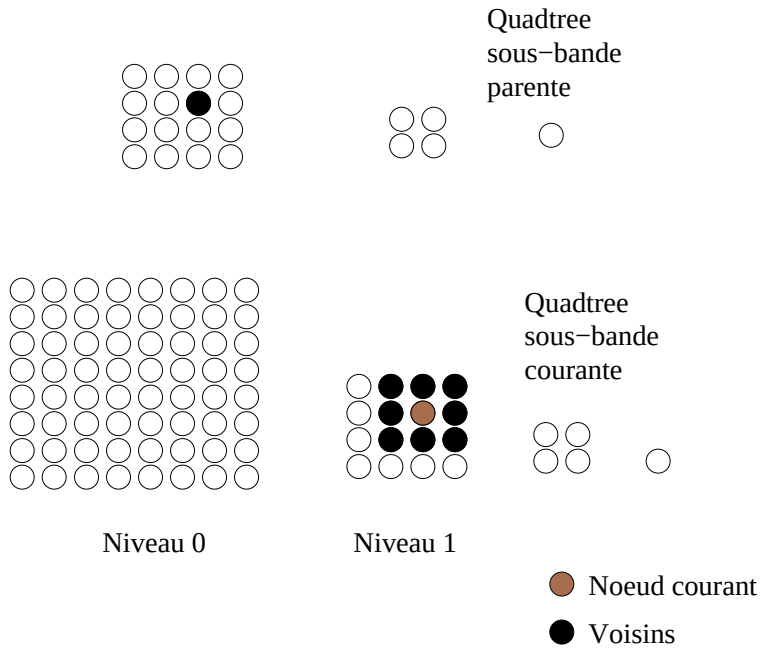


FIG. 2.20 – *EZBC: contexte intra et inter sous-bandes*

utilisé dans JPEG2000 est donnée dans la section suivante.

2.4.5 Discussion sur le codage intra et inter sous-bandes

Nous avons présenté dans la sous-section précédente un certain nombre d'algorithmes de codage de sous-bandes ondelettes. Trois types de schémas se distinguent selon la corrélation entre coefficients qu'ils utilisent. Les schémas utilisant la corrélation intra sous-bandes utilisent l'hypothèse que la décomposition spatiale en ondelettes a complètement décorrélée les données et qu'il n'y a pas de redondance entre les coefficients inter sous-bandes. Les techniques intra sous-bandes utilisent donc la corrélation spatiale des coefficients à l'intérieur d'une même sous-bande.
Les techniques inter sous-bandes émettent l'hypothèse inverse, elles présument que la décomposition ondelettes n'a pas décorrélée complètement les données et qu'un résidu de corrélation subsiste entre les coefficients inter sous-bandes. En théorie, les filtres de décomposition en ondelettes sont orthogonaux et les mêmes filtres passe-bas et passe-haut sont utilisés à l'analyse et à la synthèse. Il résulte de l'orthogonalité que le résidu de corrélation inter sous-bandes est nul. Cependant, en pratique, les filtres utilisés ne sont pas tout à fait orthogonaux mais biorthogonaux. Un couple de filtres duaux est utilisé pour l'analyse et la synthèse [Cohen 89]. L'utilisation de filtres biorthogonaux justifient le fait qu'il subsiste un résidu de corrélation entre les coefficients inter sous-bandes.
Sur le plan de la progressivité, l'utilisation d'un codage inter sous-bandes peut être

pénalisant. En effet, dans ce type de techniques, quand le codeur rencontre un coefficient non significatif d'une sous-bande n et que ce coefficient est parent d'un arbre dont tous les coefficients sont non significatifs, cette information est codée au niveau du coefficient parent. Si le décodage ne doit être fait que pour les sous-bandes basses jusqu'à la sous-bande n, le décodeur aura cependant décodé des informations relatives à des sous-bandes de plus hautes fréquences. L'utilisation du débit autorisé n'est pas optimale puisque l'on décode des informations dont on n'a pas besoin. C'est une des raisons pour lesquelles le codeur EBCOT qui utilise la corrélation intra sous-bandes offre de meilleures perlormances au niveau de l'utilisation du débit.
Le codeur EZBC utilise les deux corrélations inter et intra sous-bandes. On peut penser que le codage inter sous-bandes le pénalise en terme d'utilisation du débit par rapport à un codage uniquement intra sous-bandes. Cependant, si l'on étudie de près comment le codeur traite la corrélation inter sous-bandes, on s'aperçoit que pour une sous-bande donnée, le codeur ne code pas d'informations relatives à des sous-bandes de plus hautes fréquences, mais qu'il utilise la corrélation existante entre des sous-bandes plus basses, informations déjà décodées.
Nous allons maintenant nous intéresser aux performances de codage offertes par les deux codeurs: JPEG2000 et EZBC. Nous avons comparé ces deux codeurs en codant la première image des séquences Mobile&Calendar et Foreman au format CIF et décodé les flux à différents débits allant de 40Kb à 100Kb. Les versions utilisées pour ces codeurs sont respectivement pour JPEG2000 et EZBC, le VM8.0 et la version issue du codeur vidéo MC-EZBC [Woods 02]. La qualité objective a été mesurée à l'aide du PSNR et les courbes de résultats sont données sur la figure 2.21. Les images de la figure 2.22 donnent un aperçu de la qualité subjective de ces codeurs. En terme de PSNR, on remarque que le codeur EZBC offre des performances supérieures au codeur JPEG2000. Au niveau de la qualité subjective, cette différence est moins visible à mesure que l'on augmente le débit.

Conclusion

Dans ce chapitre, nous avons présenté un état de l'art des schémas de codage vidéo par ondelettes 3D en nous intéressant aux quatre principales briques de ces schémas: la transformée en ondelettes temporelles, le mouvement, la transformation spatiale et le codage des sous-bandes spatio-temporelles.
Nous avons vu l'importance de l'utilisation d'un modèle de compensation en mouvement dans la transformée temporelle ainsi que l'influence de ce mouvement sur l'efficacité de la transformée. L'exploitation du mouvement dans la transformée temporelle pose plusieurs problèmes qui ont été solutionné de différentes manières selon les schémas. Le problème de la réversiblité de la transformée temporelle dans le cas de mouvements sous-pixelliques a été résolu grâce à l'introduction du schéma lifting. L'utilisation d'un double champ de mouvement permet de gérer les discontinuités du mouvement, se traduisant par l'apparition de pixels déconnectés dans les méthodes de représentation du

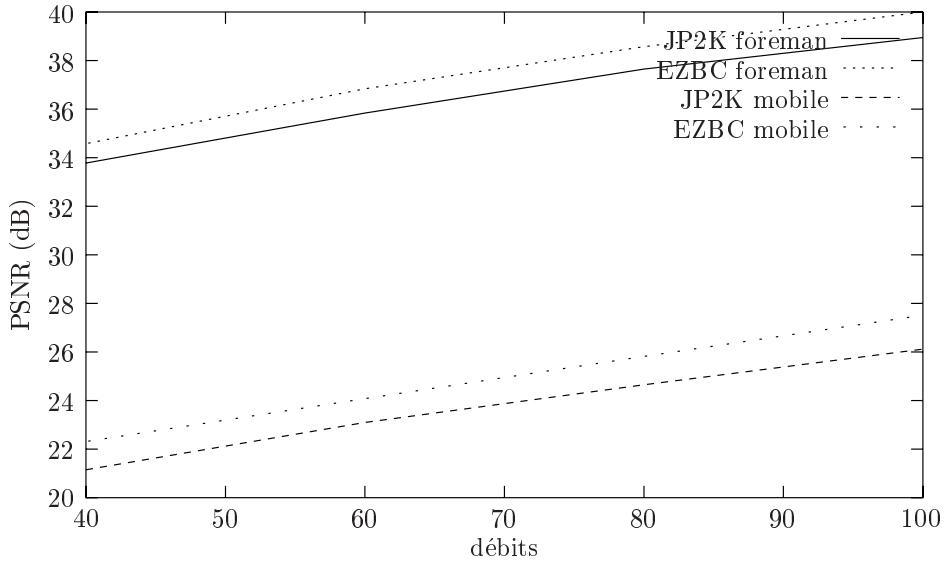


FIG. 2.21 – *Comparaison EZBC et JPEG200*

mouvement par blocs. Cependant, cette technique est assez coûteuse en terme de codage du mouvement. Ceci peut être résolu par l'utilisation de mouvements par maillages qui ont un coût de codage plus faible que les méthodes par blocs. L'utilisation des maillages offre également une solution au problème de réversiblité du champ de mouvement dans le cas où un seul champ de mouvement est utilisé. En effet, un champ de mouvement par blocs n'est pas inversible mais beaucoup de techniques tentent de l'inverser directement à cause du coût élevé d'un double champ et ceci au détriment de l'apparition de hautes fréquences que cette inversion peut produire. De plus, la continuité de la texture qu'offre la représentation par maillages permet une décorrélation efficace des informations à l'aide d'une transformée ondelette le long de l'axe temporel.

Dans ce chapitre, nous avons également étudié l'étape de transformation spatiale et sa place dans le schéma de décorrélation. Comme la plupart des schémas utilisent des filtres de transformée séparables, l'ordre dans lequel les transformations sont effectuées n'a pas d'importance. Cependant, ceci n'est plus vrai lorsque la compensation en mouvement est introduite dans la transformée temporelle. Lorsque la transformée spatiale est effectuée avant la transformée temporelle, la compensation en mouvement doit alors être appliquée dans le domaine transformée. Or la compensation en mouvement appliquée dans le domaine transformée n'est pas équivalente à celle appliquée dans le domaine image.

Enfin, nous avons présenté des codeurs progressifs de sous-bandes ondelettes et leur extension au codage des sous-bandes spatio-temporelles. Ces codeurs se distinguent d'une part par la corrélation intra et/ou inter des coefficients qu'ils utilisent et d'autre part par les techniques de codage progressif utilisées (codage par plans de bits pour EZBC et par couches de qualité pour JPEG-2000 par exemple).

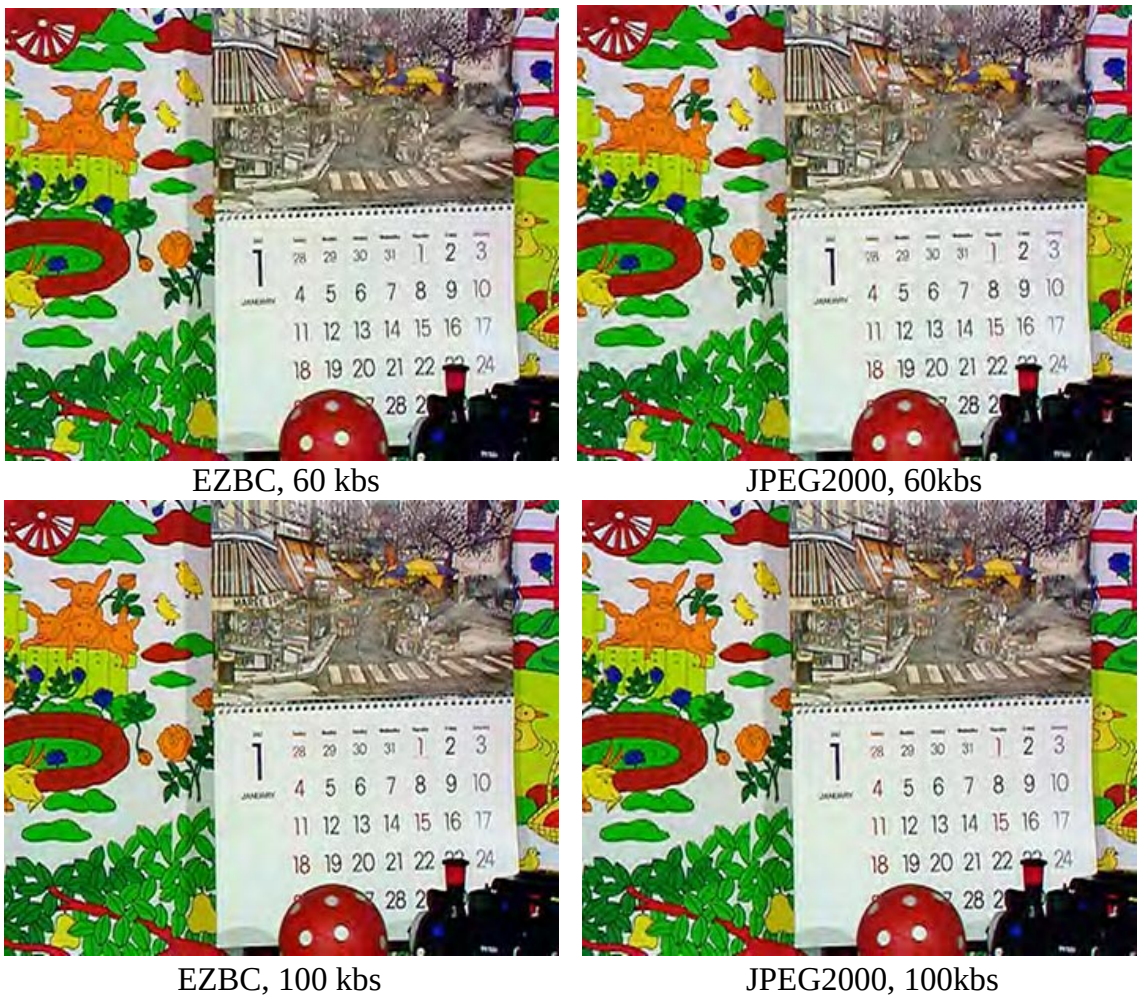


FIG. 2.22 – *Comparaison EZBC et JPEG200 sur la première image de la séquence Mobile&Calendar, CIF*

Dans le prochain chapitre, nous présenterons la contribution que nous avons apporté dans le cadre des schémas de codage vidéo scalable par ondelettes t+2D.

Chapitre 3

Codage vidéo par ondelettes: schéma global de codage

Dans ce chapitre, nous proposons un schéma de codage vidéo basé sur une approche analyse-synthèse utilisant les ondelettes t+2D pour le codage et les maillages pour la représentation du mouvement. Nous avons vu dans le chapitre précédent que le problème crucial dans les schémas de transformation temporelle est la définition et l'exploitation des trajectoires de mouvement dans la transformée. Après avoir présenté les principales caractéristiques du codeur vidéo par analyse-synthèse que nous avons utilisé, nous proposerons des solutions pour la définition et l'exploitation du mouvement dans la transformée. Ensuite, nous verrons les choix que nous avons faits pour la phase de codage et enfin nous expliquerons comment nous avons mis en œuvre la scalabilité dans le codeur.

3.1 Schéma de codage vidéo existant

3.1.1 Schéma de principe

Le codeur vidéo utilise une approche analyse-synthèse. Cette notion est analogue à celle du domaine de la 3D où l'on dispose de la représentation d'une scène (VRML par exemple) et d'un ensemble de textures qui sont plaquées sur les composants de la scène. Dans notre codeur vidéo, la représentation de la scène et les textures sont extraites de la séquence vidéo. Le mouvement représenté par un maillage offre une représentation maillée de la scène, les textures sont obtenues par projection des images sur des grilles d'échantillonnage de référence. La reconstruction est faite par reprojection des images de texture sur leur grille d'échantillonnage à l'aide du maillage. Cette approche offre une représentation totalement indépendante du champ de mouvement et de la texture de la séquence vidéo. Le principe de l'analyse-synthèse est illustré par la figure 3.1.
Représentées indépendamment, les informations de mouvement et de texture sont codées séparément à l'aide d'une transformée ondelettes t+2D et d'un codage progressif. Le schéma de codage est présenté sur la figure 3.2.

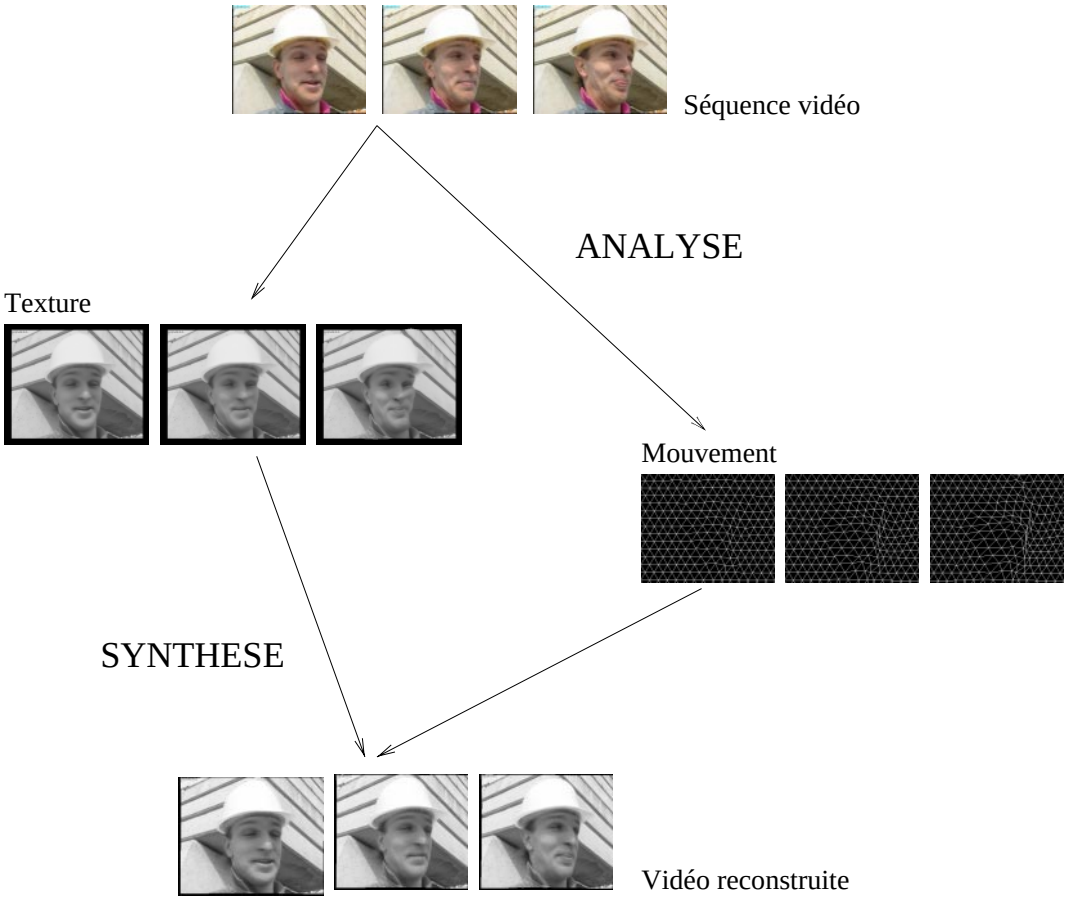


FIG. 3.1 – *Principe de l'analyse-synthèse*

Le schéma par analyse-synthèse offre une totale décorrélation du mouvement et de la texture. Une transformée ondelette temporelle peut alors être appliquée sur la texture redressée sans être influencée par le mouvement.
De plus, la représentation du mouvement par un maillage permet un suivi continu de la texture au cours du temps qui peut ensuite être exploité par une transformation temporelle le long des trajectoires.
Contrairement aux approches par blocs, le champ de mouvement par maillage est continu, il ne présente pas de discontinuités artificielles. De plus, le mouvement par maillages est inversible, ce qui réduit le coût de codage du mouvement dans le cas de codage par transformée ondelette temporelle compensée en mouvement.

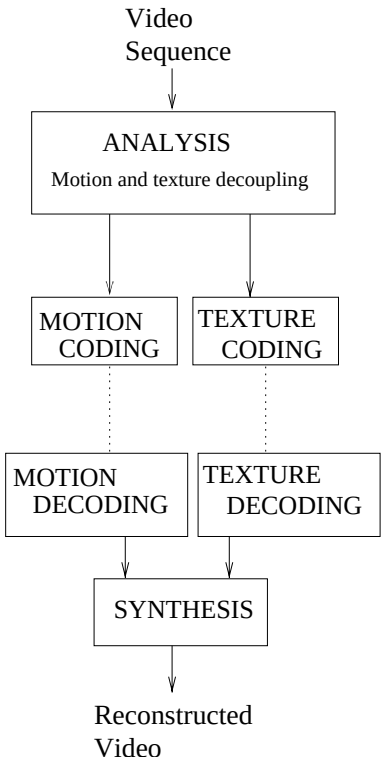


FIG. 3.2 – *Codeur vidéo par analyse-synthèse*

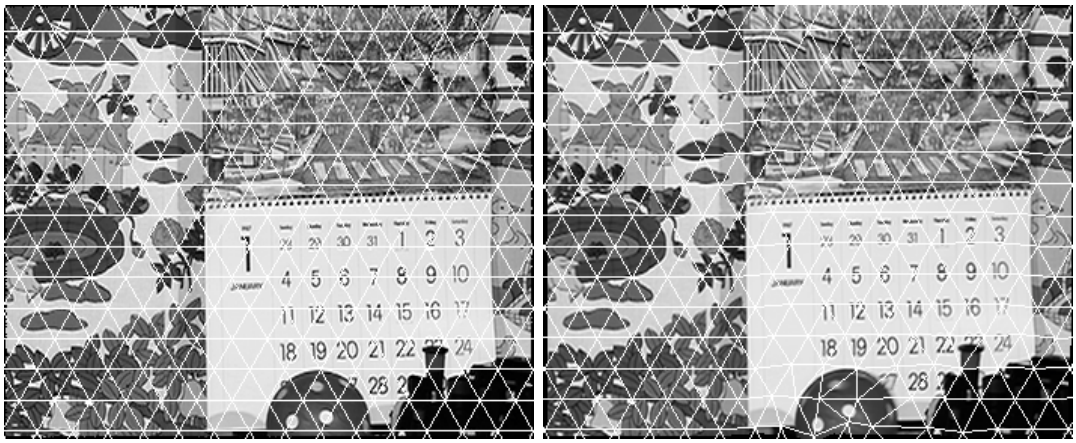


FIG. 3.3 – *Estimation du mouvement par maillages*

3.1.2 Représentation du mouvement et de la texture

Le mouvement est représenté par un maillage régulier hiérarchique. L'estimation du mouvement est effectuée entre la première image du GOF (Group Of Frames) et chaque image du groupe. Ce suivi sur tout le GOF (figure 3.3) permet la compensation en mouvement entre deux images i et k quelconques du GOF. L'estimation du mouvement est hiérarchique et multi-résolution [Marquant 00] et utilise également une technique multi-grille pour la propagation du mouvement à travers les niveaux de hiérarchie [Cammas 03].
La texture est extraite des images par projection sur une grille d'échantillonnage de référence. Cette projection permet de redresser les images par rapport à une référence et de décorréler le mouvement de la texture. Les images sont projetées sur un support plus grand que leur support d'origine afin de construire une mosaïque dynamique représentant les variations d'illumination dans les images du GOF courant [Pateux 01].

3.1.3 Schéma de codage

3.1.3.1 Structure générale du schéma de codage

Le codage des informations de texture et celui des informations de mouvement se font indépendamment l'un de l'autre mais selon le même schéma. Le codage est structuré en deux couches.
Une couche basse code les informations de mouvement et de texture de la première et de la dernière image d'un GOF. Elle permet de reconstruire une version sous-échantillonnée de la séquence originale. Les images intermédiaires entre la première et la dernière image d'un GOF sont interpolées à partir de ces deux images reconstruites, (figure 3.4).
Une couche haute scalable permet de raffiner la séquence vidéo basse qualité reconstruite par la couche de base. Cette couche de raffinement code un résidu de prédiction des informations de texture et de mouvement entre le signal original et le signal inter-

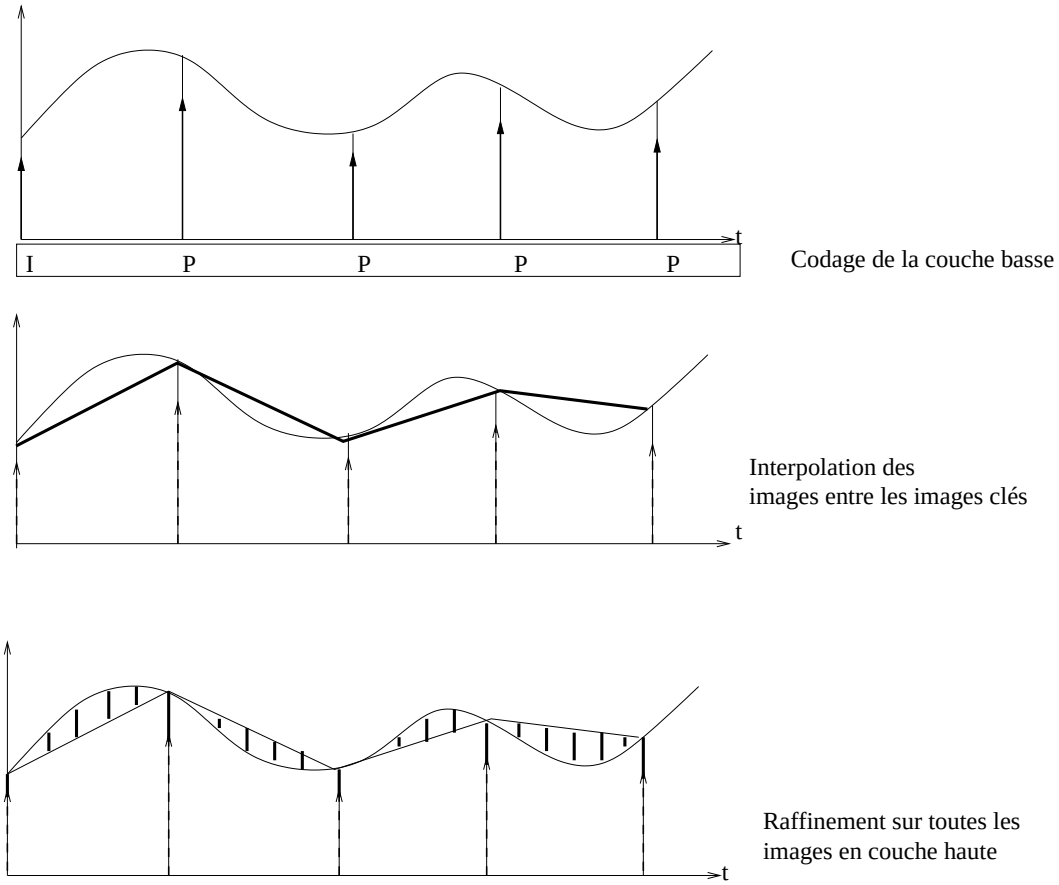


FIG. 3.4 – *Structure en couches du codage*

polé obtenu par la couche de base.

3.1.3.2 Calcul de résidus d'interpolation

Après avoir été projeté sur la première image du GOF, les images de texture sont prédites à l'aide d'une interpolation entre la première et la dernière image du GOF selon l'équation:

$$\widehat{M_t(x,y)} = (1-\alpha) * M_1(x,y) + \alpha * (M_N(x,y))$$

où $\widehat{M_t}$ est l'image de texture prédite au temps t, M_i l'image de texture au temps i et $\alpha = \frac{t-1}{N-1}$.

3.1.3.3 La couche de base

La couche de base contient les informations de mouvement et de texture de la première et de la dernière images du GOF. Un GOF peut être un GOF Intra, la première image du GOF est codée en Intra, ou un GOF Inter, la première image du GOF correspond à la dernière image du GOF précédent, elle n'est pas codée dans le GOF courant. La dernière image d'un GOF est codée par prédiction par rapport à la première image du GOF. La couche basse code l'erreur de prédiction de la dernière image.

3.1.3.4 Le raffinement en texture

La couche de raffinement contient les erreurs de prédiction (mouvement et texture) de toutes les images intermédiaires et un raffinement sur les images aux extrémités du GOF. Une décomposition temporelle est appliquée sur les résidus des images intermédiaires. La transformée temporelle est appliquée dans sa forme classique avec un filtre Daubechies 9/7.
Le signal de résidu est nul aux extrémités, ceci permet d'effectuer une extension antisymétrique sur les bords du signal à transformer et permet une transformation temporelle totalement orthogonale, même aux bords.
Les sous-bandes temporelles ainsi que le raffinement des images extrêmes sont ensuite transformés spatiallement par un filtre Daubechies 9/7 sur cinq niveaux de décomposition.
Les sous-bandes spatio-temporelles sont codées par un codeur progressif, EBCOT.

3.1.3.5 Le codage du mouvement

Le mouvement est codé de la même manière que la texture, sur deux couches. La couche de base contient l'information de mouvement de la dernière image du GOF.
Le mouvement des images intermédiaires est interpolé comme pour la texture, à partir du mouvement de la première et de la dernière image du GOF. Les résidus de prédiction des images intermédiaires sont transformés temporellement à l'aide d'un filtre Daubechies 9/7. Puis, les sous-bandes temporelles sont transformées en ondelettes de mouvement [Marquant 00].
La couche de raffinement code les sous-bandes spatio-temporelles à l'aide d'un codeur arithmétique contextuel couplé à une optimisation débit/distorsion.
La couche de raffinement du mouvement est complètement scalable. Les maillages permettent de représenter le mouvement à l'aide de la position des nœuds du maillage $Pos(x, y, t)$. Cette représentation offre l'avantage d'être adaptée pour l'interpolation des positions du maillage dans les images intermédiaires à partir des positions du maillage des images extrêmes ainsi que pour la transformée en ondelette temporelle.
De plus, la séparation du mouvement et de la texture grâce au schéma analyse-synthèse permet de coder le mouvement avec pertes sans influencer le codage et la qualité visuelle de la texture. Dans une certaine mesure, une erreur sur le mouvement n'est pas perceptible dans la vidéo reconstruite et permet de reporter le gain en débit sur le codage de la texture, améliorant la qualité visuelle de la séquence, surtout à bas débits.

3.2 L'analyse

3.2.1 Choix de la grille de référence

La texture est extraite des images par projection sur des grilles d'échantillonnage de référence. Cette étape correspond à un redressement des images par rapport à une référence, elle permet de décorréler le mouvement des images de texture.
Dans la version initiale du codeur, les images sont projetées sur une seule grille de référence: la première image du GOF. Cette technique est similaire à la méthode de [Taubman 94] bien que le mouvement utilisé ici soit plus complexe qu'un simple mouvement global. Cette technique permet de faire abstraction du mouvement pour toutes les étapes de codage suivantes et notamment lors de la transformée temporelle. Cependant, l'inconvénient est que la qualité de reconstruction des images à l'intérieur du GOF est inégale pour un codage à débit constant.
En effet, la qualité se dégrade de plus en plus que les images s'éloignent de l'image utilisée comme référence pour la compensation. Ceci est dû au fait que la projection d'une image lors de la compensation en mouvement déforme les images, dû aux contractions ou aux étirements de mailles. Plus l'image est loin de la référence, plus le maillage est déformé entre l'image courante et l'image référence et donc plus la texture redressée sera déformée. Les déformations engendrées ne sont pas réversibles, et dans le cas où la déformation est trop importante, la reconstruction de l'image par reprojection sur sa grille d'échantillonnage entraîne des défauts dans les images qui ne pourront être corrigés.
Dans le cas où l'on projette toutes les images sur la première image du GOF, la dernière image du GOF est la moins bien reconstruite.
De plus, la qualité décroissante au cours du GOF est gênante visuellement, la qualité décroît dans le GOF, puis une amélioration brutale de la qualité est visible dans les premières images du GOF suivant, puis la qualité décroît jusqu'au GOF suivant, etc... La séquence reconstruite présente ainsi de brusques sauts de qualité.
Pour améliorer la qualité de reconstruction de toutes les images, l'idéal serait de reconstruire chaque image sur sa propre grille d'échantillonnage, comme dans le schéma de [Secker 01]. Cependant, cette technique n'offre pas une décorrélation complète du mouvement et de la texture car alors un nombre non négligeable de compensations en mouvement sont nécessaires dans la transformation temporelle. Pour une transformation temporelle avec un filtre 5/3 sur trois niveaux et huit images, cette technique nécessite 22 compensations en mouvement pour traiter la luminance, tandis que la première technique nécessite seulement 7 compensations en mouvement initiales pour construire la mosaïque et aucune durant la transformation temporelle.
Un autre inconvénient apparaît dans cette technique, lié au mouvement estimé dans la séquence. Quels que soient la méthode d'estimation et le type de représentation du mouvement utilisés, la trajectoire du mouvement n'est jamais connue de manière idéale. Ceci se traduit lors de la compensation en mouvement par blocs entre deux images par des pixels déconnectés ou multiplement connectés si un champ unidirectionnel est utilisé, introduisant alors une mauvaise prédiction dans ces zones ; avec un champ bi-

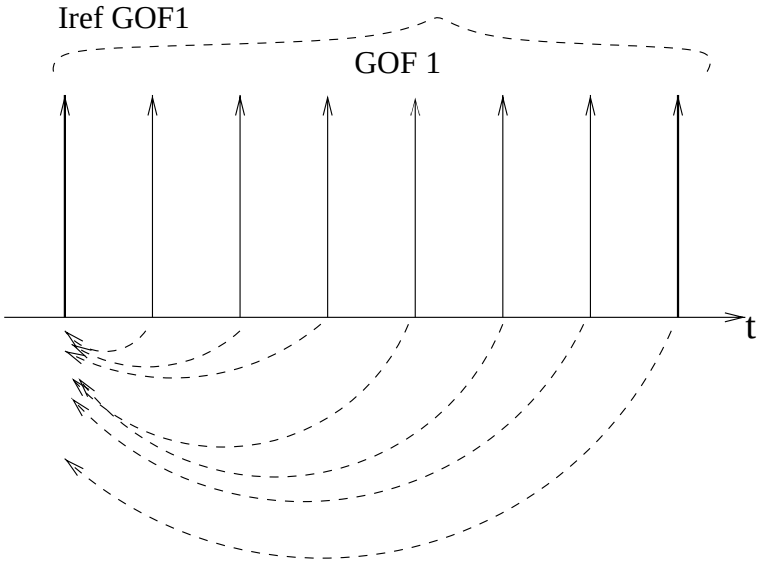
directionnel, ces pixels n'apparaissent plus, mais le coût de mouvement est important et de hautes fréquences persistent aux frontières des blocs. Le problème du mouvement non idéal existe aussi avec une représentation par maillage. Dans les zones de contraction, des hautes fréquences apparaissent dues à l'aliasing, les zones d'étirement présentent un étalement des textures. Les contractions et étirements introduisent dans la texture des motifs difficiles à coder.
Le mouvement non idéal utilisé lors des compensations en mouvement dans la transformée temporelle influence la qualité des sous-bandes et des étapes de filtrage de la transformée. Pour cette raison et de par la complexité opératoire que la technique de projection de chaque image représente, nous avons finalement choisi une troisième technique dans le choix des images de référence.

Cette technique choisit deux images de référence dans chaque GOF, la première et la dernière image du GOF. Ce choix permet de diminuer la distance de projection des images par rapport à leur référence. La première et la dernière image du GOF sont reconstruites parfaitement car elles sont définies sur leur propre grille d'échantillonnage. Les images de la première moitié du GOF sont projetées sur la première image, comme pour la première technique, les images de la deuxième moitié du GOF sont projetées sur la dernière image. Ceci permet d'améliorer la qualité de reconstruction pour toutes les images de la deuxième moitié du GOF par rapport à la première technique présentée.
Cette technique permet également un compromis au niveau coût opératoire entre les deux précédentes techniques. Le nombre de compensations en mouvement pour notre technique à deux références est moindre que pour la technique précédente, en effet outre les compensations initiales pour obtenir les images de texture, peu de compensations en mouvement sont nécessaires dans la transformée temporelle, les compensations concernent le filtrage entre images de moitié de GOF différentes. Pour un GOF de huit images, l'obtention des textures nécessite 6 compensations en mouvement lors de la projection des images sur leur référence, et la transformée temporelle avec un filtre 5/3 lifting sur trois niveaux, 6 compensations en mouvement, ce qui donne un total de 12 compensations en mouvement pour la technique à deux références.
La figure 3.5 illustre les trois techniques de projection que nous venons de présenter.

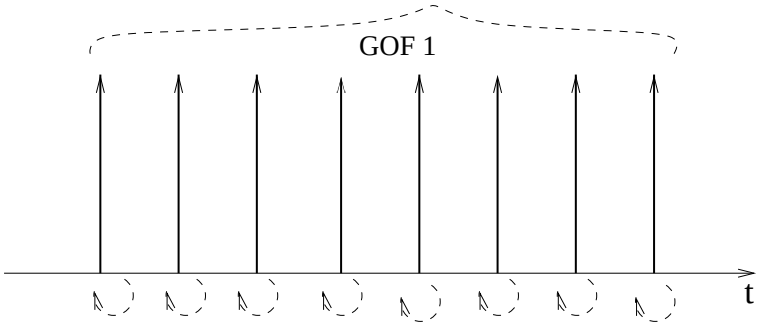
3.2.2 Le padding spatio-temporel

Lors de la projection des images sur une grille de référence, les images compensées en mouvement ne sont pas forcément complètement contenues dans le cadre de la référence et peuvent dépasser des frontières du cadre de définition. Pour la représentation par blocs comme pour celle par maillage, le problème est résolu par l'utilisation d'un cadre de compensation plus large que le cadre de la référence. Cependant, ce cadre plus large n'est alors pas défini partout, figure 3.6. Il existe une frontière brutale entre le domaine où le signal est défini et le domaine où il n'est pas défini. Cette transition brutale crée des hautes fréquences spatio-temporelles artificielles dans le signal transformé et des motifs difficiles à coder. Afin de ne pas pénaliser le coût de codage par l'élargissement

1. Projection sur la première image du GOF:



2. Projection sur chaque image:



3. Projection sur la première et la dernière image du GOF:

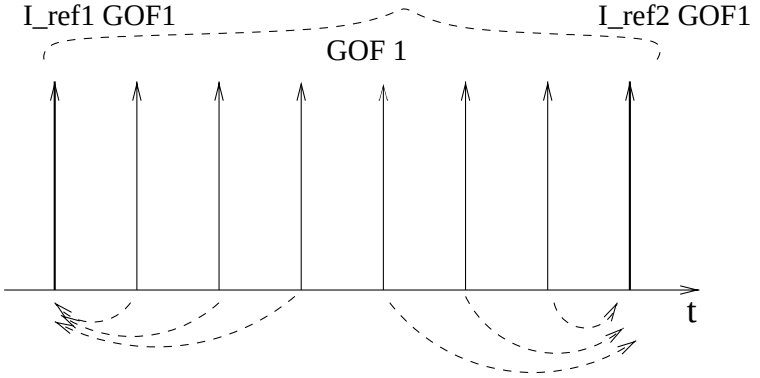


FIG. 3.5 – *Projection des images sur leur grille de référence*

FIG. 3.6 – *Projection d'une image sur un support plus grand que celui de l'image*

du cadre de compensation, il faut soit coder le signal par des techniques de codage basées forme, soit étendre de manière continue le signal dans les zones où il n'est pas défini et coder ainsi le signal et son extension.

Le cas du codage basé forme n'est pas adapté à notre schéma de codage car dans ce cas, le mouvement doit être codé sans pertes. En effet, le mouvement codé avec pertes reconstruit une image dans un cadre de définition où les frontières ne sont pas exactes. Si la texture est codée selon un schéma basé forme, la reprojection de celle-ci dans le domaine image peut alors introduire des zones non définies dans la séquence reconstruite.
Pour cette raison, nous avons choisi d'étendre le signal dans les zones non définies à l'aide d'une extension lisse.

3.2.2.1 Les techniques d'extension existantes

On distingue plusieurs techniques d'extension d'un signal.
La technique utilisée dans le cas de codage vidéo MPEG est une répétition de la valeur des échantillons aux bords du signal. Le dernier échantillon est répété à l'infini. Dans le cas du codage 2D, une telle extension peut créer des hautes fréquences artificielles, elle peut par exemple transformer une petite discontinuité locale sur le bord d'une image en une discontinuité de plus grande dimension dans l'extension, qui coûtera cher à coder et qui de plus ne participera à la reconstruction que d'une petite zone.
Une autre technique utilisée dans le codage ondelette basé forme (SA-DWT: shape-adapted discrete wavelet transform) est la répétition du signal en miroir par rapport au bord du signal, [Minami 01, Mertins 98]. L'extension en miroir doit être faite horizontalement puis verticalement, ou l'inverse. La répétition en miroir du signal n'est pas lisse dans les deux directions horizontale et verticale. De plus, même si cette technique est très utilisée dans le cadre du codage DWT classique, dans le cas du codage basé forme, le support est trop petit pour pouvoir construire une extension lisse et considérée un

signal étendu à support infini.

3.2.2.2 La technique proposée

Nous proposons ici une technique d'extension (padding) spatio-temporelle.
Le padding consiste à remplir les zones non définies par une prolongation continue du signal contenu dans les zones définies. On souhaite définir un signal lisse $\hat{S}$ égal au signal original S aux échantillons connus et qui soit lisse ailleurs. Une décomposition itérative par analyse-synthèse du signal original permet de trouver un tel signal $\hat{S}$. Le signal lisse $\hat{S}$ est représenté de manière hiérarchique, sous forme paramétrique:

$$\begin{array}{l} \hat{S} = \hat{S}^L \\ \hat{S}^l = \sum_{k=1}^{k=l} \Delta\hat{S}^k \\ \Delta\hat{S}^l = \sum_k c_{k,l}\psi_k^l \end{array} \tag{3.1}$$

Le padding procède de manière itérative en minimisant à chaque niveau la différence $S - \hat{S}^l$, avec $\hat{S}^l$ pouvant se réécrire de manière récursive:

$$\hat{S}^l = \hat{S}^{l-1} + \Delta\hat{S}^l.$$

Le signal $\hat{S}$ est raffiné de manière successive par des étapes d'analyse et de synthèse à chaque résolution. A chaque niveau, un résidu Res^l est calculé par:

$$Res^l = S - \hat{S}^l$$

Initialement,

$$\begin{array}{l} Res^0 = S \\ \hat{S}^0 = 0 \end{array} \tag{3.2}$$

La phase d'analyse calcule le raffinement $\Delta\hat{S}^l$ qui approxime au mieux le résidu Res^{l-1}. La synthèse de $\Delta\hat{S}^l$ permet la mise à jour du signal lisse $\hat{S}^l$.
La décomposition du processus de padding est analogue à une décomposition ondelettes où le signal $\hat{S}^l$ est décomposé en un signal moyen $\hat{S}^{l-1}$ et un signal de raffinement $\Delta\hat{S}^l$. Le signal moyen et le signal de raffinement peuvent être réécrit respectivement comme une somme de fonctions d'échelles et de fonctions d'ondelettes:

$$\begin{array}{l} \hat{S}^l = \sum_{k=1}^{k=2^{l+1}} a_k^l \Phi_k^l \\ \Delta\hat{S}^l = \sum_{k=1}^{k=2^{l+1}} c_k^l \Psi_k^l \end{array} \tag{3.3}$$

où Φ et Ψ sont respectivement des fonctions de base d'échelle et d'ondelettes, et c_k^l les coefficients à déterminer par la phase d'analyse.
Ce processus général de padding est appliqué sur les images de texture. Une première technique consiste à compléter de manière séparée chaque image mosaïque. Cette extension du signal fournit des mosaïques avec un prolongement continu spatialement (figure 3.7) mais ne prend pas en compte la continuité temporelle. En effet, des hautes fréquences sont introduites le long de l'axe temporel car le prolongement n'est pas continu entre deux images successives.
Pour prendre en compte l'axe temporel, le signal est complété spatialement pour les deux mosaïques aux extrémités du GOF, puis interpolé pour les mosaïques intérieures,

figure 3.8. Le signal prolongé correspond à la prédiction des informations par les images extrêmes de la couche basse. Ce prolongement n'introduit alors pas d'informations supplémentaires à coder dans les zones où la prédiction est exacte. Cependant, la technique peut introduire des hautes fréquences spatiales à la frontière entre les zones où la prédiction est exacte et les zones où elle ne l'est pas.
Une troisième technique qui permet d'obtenir un prolongement continu dans les trois dimensions (mais avec un coût en complexité plus élevé) est d'effectuer le padding par analyse-synthèse dans les trois dimensions (spatiales et temporelle) comme montré sur la figure 3.9 et selon les équations:

$$\begin{array}{l} \hat{S} = \hat{S}^{L,T} \\ \hat{S}^{l,t} = \sum_{m=1}^{m=t} \sum_{k=1}^{k=l} \Delta\hat{S}^{k,m} \\ \Delta\hat{S}^{l,t} = \sum_k c_{k,l,t}\psi_k^{l,t} \end{array} \tag{3.4}$$

Le signal étendu spatio-temporellement est alors la somme des extensions spatiales de toutes les fréquences temporelles:

$$\begin{array}{l} \hat{S}^{l,t} = \sum_{m=1}^{m=t} \hat{S}^{l,m} \\ \hat{S}^{l,m} = \hat{S}^{l-1,m} + \Delta\hat{S}^{l,m} \end{array} \tag{3.5}$$

L'extension du signal par la décomposition par analyse-synthèse est alors continue et permet de ne pas introduire de hautes fréquences dans le signal à coder.
Cependant, le signal est rempli par GOF et l'extension n'est pas continue entre les GOFs. Ainsi, la dernière image d'un GOF est complétée par rapport aux informations contenues dans ce GOF, ces informations sont des informations passées. Pour le GOF suivant, cette image est la première image du GOF, elle est complétée par les informations connues du GOF qui sont des informations futures. Le remplissage de cette image commune se fait séparément pour chaque GOF et introduit une discontinuité dans la représentation de la texture aux frontières de GOF. De plus, dans le premier GOF, la non-utilisation du futur pour remplir les images du GOF provoque l'apparition de flou dans les zones entrant dans la vidéo. Ces zones sont des zones de découvrement qui ne peuvent être connues qu'en utilisant l'information future de la vidéo.
Pour éliminer ce flou et éviter la discontinuité de texture aux frontières des GOFs, le remplissage est effectué en utilisant l'information connue sur plusieurs GOFs. L'idéal serait de calculer l'extension du signal sur le support de la séquence entière. Cependant, la complexité opératoire du système deviendrait énorme. Le remplissage des textures est alors fait sur le support de deux GOFs successifs, figures 3.10 et 3.11.
L'extension sur le GOF courant est calculé en utilisant les informations de texture du GOF courant et du GOF suivant. Puis, la dernière image du GOF courant, partagée avec le GOF suivant, est figée. Ensuite, on recommence sur le GOF suivant avec en entrée la première image du GOF qui a été étendue dans l'opération de padding du GOF précédent.

La figure 3.12 résume la phase d'analyse du codeur. Quand le mouvement entre les images est connu, les images de la séquences sont redressées par rapport aux grilles

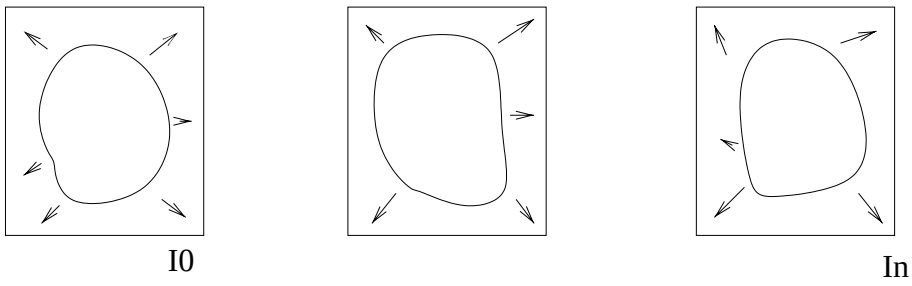


FIG. 3.7 – *Padding spatial de chaque image*

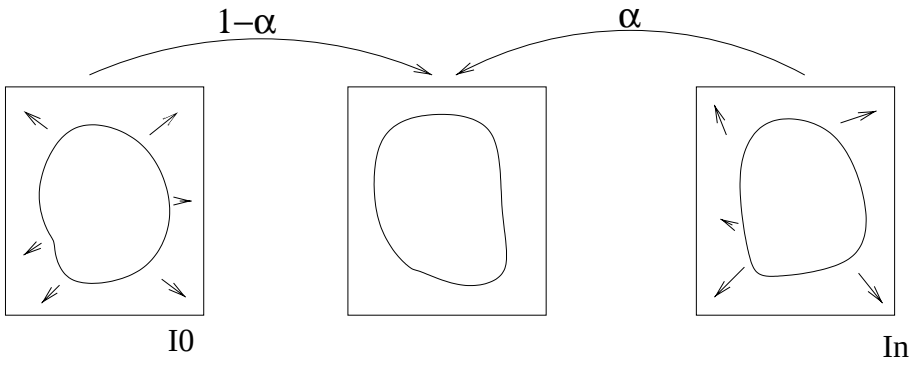


FIG. 3.8 – *Padding spatio-temporel avec interpolation linéaire*

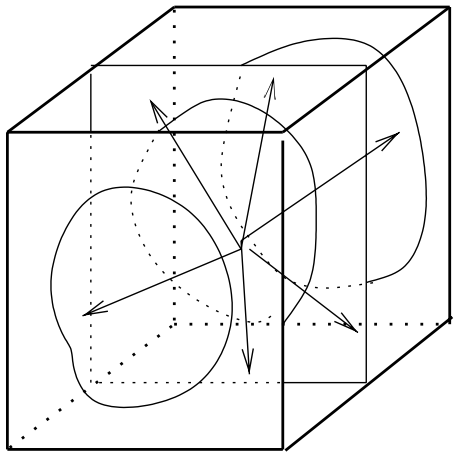


FIG. 3.9 – *Padding spatio-temporel par analyse-synthèse 3D*

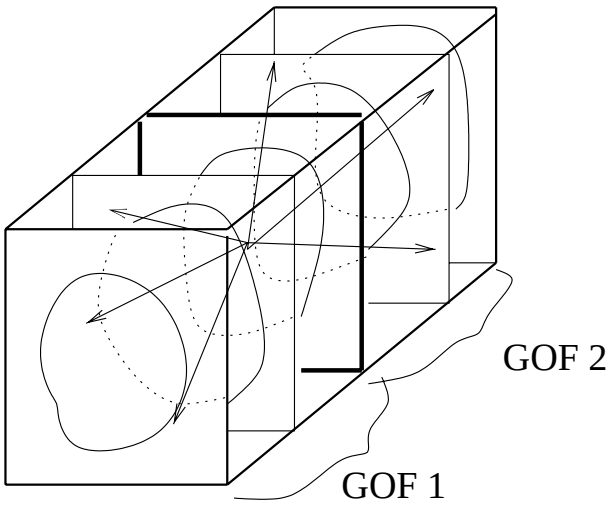


FIG. 3.10 – *Padding spatio-temporel par analyse-synthèse 3D et extension sur deux GOFs*

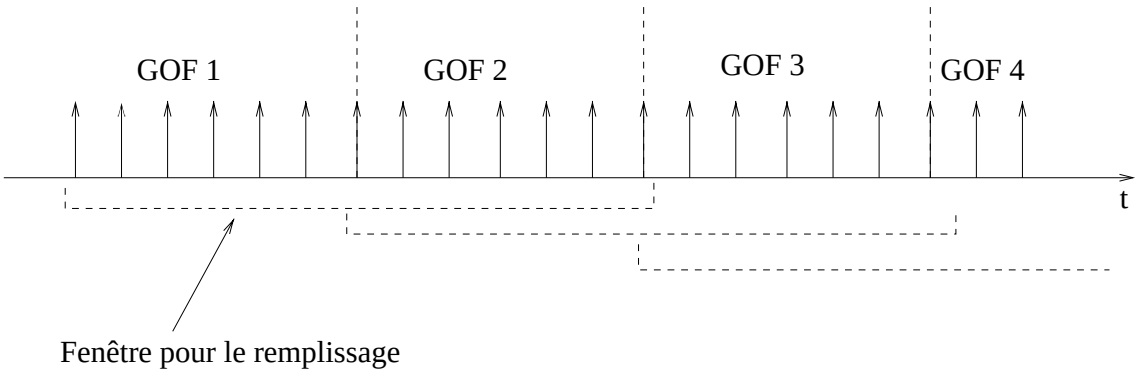


FIG. 3.11 – *Padding sur deux GOFs successifs*

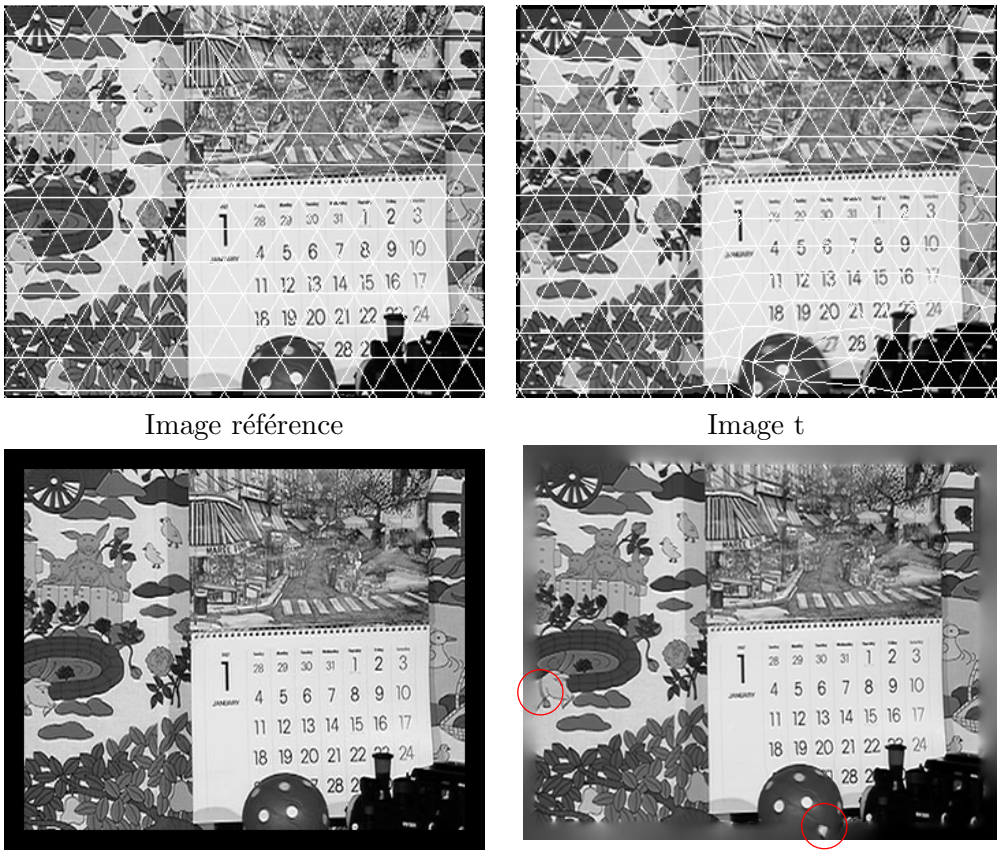


Image référence | Image t

Texture projetée t | Texture paddée t

FIG. 3.12 – *Analyse d'une séquence. Sur la texture paddée, sont entourées les zones d'informations introduites par le padding par rapport à la texture projetée.*

d'échantillonnage de référence. Les images de textures sont ensuite complétées par la technique de padding spatio-temporel par analyse-synthèse.
L'image de texture paddée de la figure 3.12 montre l'intérêt de l'utilisation du futur dans la phase de padding. En effet, de nouvelles zones d'informations apparaissent de manière nette par rapport à la mosaïque originale. Ces zones sont visibles dans le bas du ballon où un point blanc est apparu ainsi que sur le bord gauche de l'image où le cou de la chèvre est visible. Ces zones entrant dans la vidéo apparaîtront nettes dès leur entrée dans les images de la séquence reconstruite.

3.3 Codage spatio-temporel

Après la phase d'analyse, l'étape de codage traite les informations de mouvement et de texture indépendamment. Ces deux types d'information sont codés de la même manière mais séparément.

Le flux binaire est structuré en deux couches. Une couche de base contient les informations relatives à la première et à la dernière image de chaque GOF. A partir de cette couche, une première version basse qualité de la séquence vidéo peut être reconstruite. Le mouvement et la texture des images intermédiaires du GOF sont interpolées entre la première et la dernière image.
Une couche de raffinement scalable code les erreurs de codage des images extrêmes du GOF ainsi que les résidus d'interpolations en mouvement et en texture des images intermédiaires à l'aide d'une transformée ondelette t+2D.

3.3.1 Transformée temporelle lifting

Les informations de texture sont prédites à l'aide des images extrêmes du GOF selon l'équation:

$$M_t(x,y) = (1-\alpha) * (M_0^t(x,y)) + \alpha * (M_{N-1}^t(x,y)),$$

avec M_t la mosaïque à l'instant t, M_i^t la mosaïque de la i-ième image compensée par rapport à l'image t, $\alpha = \frac{i}{N}$ et N le nombre d'images du GOF. Cette prédiction permet de calculer des résidus sur toutes les images du GOF.
Les résidus de texture sont ensuite décomposés à l'aide d'une transformée ondelette temporelle lifting. Le lifting a été présenté dans le chapitre précédent (2.1.4.1). Il permet d'effectuer une transformée ondelettes simple, rapide et réversible. Le signal résidu à transformer est nul aux extrémités. Ceci permet lors de la transformée temporelle d'effectuer une extension anti-symétrique aux bords et d'avoir une transformation temporelle totalement orthogonale, même aux bords du signal.
La décomposition des résidus est faite sur un nombre de niveaux fonction de la taille du GOF traité. Les équations de décomposition pour un filtre 5/3 sont les suivantes:

$$\begin{aligned}
H_k[m,n] &= x_{2k+1}[m,n] - \tfrac{1}{2}(\mathrm{Comp}(2k,2k+1)[x_{2k}](m,n) + \\
&\quad \mathrm{Comp}(2k+2,2k+1)[x_{2k+2}](m,n)) \\
L_k[m,n] &= x_{2k} + \tfrac{1}{4}(\mathrm{Comp}(2k-1,2k)[H_{k-1}](m,n) + \\
&\quad \mathrm{Comp}(2k+1,2k)[H_k](m,n)) \\
\text{avec } \mathrm{Comp}(i,j)[x] &= \begin{cases} W_{i\to j}(x) & \text{si } t_{ref}(i) \neq t_{ref}(j) \\ x & \text{sinon} \end{cases}
\end{aligned}$$

$W_{i\to j}$ est l'opérateur de compensation en mouvement de i vers j. Dans la transformée temporelle, la compensation en mouvement n'est nécessaire que lorsque les images de textures ont des images référence différentes. Dans le cas contraire, la transformée temporelle est un filtrage 1D classique. Explicitons les images références (t_{ref}) pour chaque image i du GOF selon la technique de projection utilisée:

- Si toutes les images sont projetées sur la première image (figure 3.5, 1.):

$$t_{ref}(i) = 0$$

- Si chaque image est projetée sur elle-même (figure 3.5, 2.):

$$t_{ref}(i) = i$$

Projection	5/3	5/3 tronqué	9/7	Qualité de la dernière image
1, une référence	0	0	0	non
2, référence=image	22	12	38	oui
3, deux références	6	3	10	oui

TAB. 3.1 – *Comparaison du nombre de compensations en mouvement nécessaires pour chaque technique de projection pour un groupe de 8 images*

– Si les images sont projetées sur la première et la dernière image (figure 3.5, 3.):

$$t_{ref}(i) = \begin{cases} 0 & \text{si i appartient à la première moitié du GOF} \\ N-1 & \text{si i appartient à la deuxième moitié du GOF} \end{cases}$$

Le tableau 3.1 compare le nombre de compensations en mouvement nécessaires pour chaque technique de projection présentées dans 3.2.1 et pour différents filtres de transformée. Le tableau indique également si la technique permet de bien reconstruire la dernière image du GOF. La troisième technique (projection sur deux images références) offre le meilleur compromis entre complexité opératoire (nombre de compensations en mouvement) et qualité de reconstruction.

Au niveau de la compensation en mouvement, le mouvement est représenté par un maillage déformable. Cette représentation est réversible et permet à partir d'un seul champ de mouvement de compenser une image vers l'avant et vers l'arrière. De plus, le champ de mouvement est dense, le mouvement en chaque point est calculé par interpolation lagrangienne des déplacements des nœuds de la maille contenant le point.
La représentation du mouvement par maillage assure une continuité temporelle de la texture.
Dans une vidéo, le mouvement présente des zones de contraction et d'expansion qui se traduisent dans les schémas basés blocs par l'apparition de pixels déconnectés ou multiplement connectés. Avec la représentation par maillage, ces phénomènes se traduisent par des contractions et des expansions de texture. Lors de la compensation en mouvement, les textures sont projetées sur la grille de l'image vers laquelle elles sont compensées. Dans les zones de contraction, plusieurs pixels sont projetés au même endroit, et dans les zones d'expansion, un pixel est projeté à plusieurs endroits. Au codage des sous-bandes temporelles, ce phénomène crée des motifs difficiles à coder. De plus, à la synthèse temporelle des textures, il entraine un étalement des erreurs de quantification sur les images reconstruites, augmentant leur distorsion.
L'exploitation de la transformée temporelle ondelette compensée en mouvement associée à une représentation du mouvement par maillage permet de bien décorréler l'information de texture à coder. Cependant, l'étalement des erreurs de quantification est la conséquence directe des expansions locales du maillage. Afin de tenir compte de ce phénonème, la distorsion de chaque maille doit être pondérée par un facteur de déformation. La déformation des mailles doit être prise en compte dans la phase de quantification des coefficients et dans le codage des sous-bandes temporelles.

3.3.2 Décomposition en ondelettes spatiales avec nombre de niveaux adaptatif

Les sous-bandes temporelles ainsi que les images de résidus de codage des images aux extrémités sont normalisées puis sont ensuite décomposées par une transformation en ondelettes spatiales. La transformée est appliquée à l'aide d'un filtre Daubechies 9/7 et le nombre de niveaux de décomposition varie selon la nature de la sous-bande.
En effet, les sous-bandes hautes et basses fréquences n'ont pas le même aspect. Les basses fréquences sont proches du signal original, tandis que les hautes fréquences sont semblables à des résidus de prédiction. Pour cette raison, les hautes fréquences sont décomposées sur un faible nombre de niveaux de décomposition (2 ou 3), alors que les basses fréquences sont décomposées sur un plus grand nombre de niveaux (5). Les images de résidus sont décomposées sur un faible nombre de niveaux (3).
Les sous-bandes spatio-temporelles normalisées sont ensuite codées à l'aide d'un codeur progressif EBCOT. L'orthogonalité des sous-bandes permet d'effectuer une optimisation débit/distorsion efficace par EBCOT.

3.3.3 Codage scalable du mouvement-Adaptation de la distorsion à la résolution de décodage

La couche de raffinement du mouvement code les résidus d'interpolation du mouvement des images intermédiaires. Les résidus sont transformés par une ondelette spatio-temporelle et codés de manière scalable à l'aide d'une optimisation débit/distorsion. Au décodage, la distorsion tolérée est fonction de la résolution spatiale de la séquence reconstruite.
En effet, une erreur d'un pixel au niveau du mouvement n'a pas le même effet sur une séquence reconstruite au format SD ou sur une séquence reconstruite en CIF ou en QCIF. Si la même distorsion est utilisée pour tous les formats, une erreur d'un pixel au format SD équivaut à une erreur d'un quart de pixel en CIF et d'un huitième en QCIF. Si l'on veut tolérer la même erreur quelle que soit la résolution de décodage, il est nécessaire d'adapter la distorsion au décodage à la résolution de la séquence reconstruite. Si l'on autorise une distorsion donnée D au niveau SD, la distorsion pour le niveau CIF est alors $4\times$D et $16\times$D pour le QCIF.
Dans les schémas de codage vidéo proposés dans la littérature, et notamment dans ceux proposés en réponse au Call For Proposal for Scalable Video Coding technology ([cfp 03]), pour coder le mouvement de manière scalable, les techniques utilisent des plans de bits. La scalabilité du mouvement est obtenue en faisant varier la précision du mouvement en fonction de la résolution de décodage. Si l'on considère une précision au quart de pixel au format SD, la même précision au format CIF ou QCIF est obtenue pour une précision au demi-pixel ou au pixel au format SD.

3.3.4 Mesure de qualité objective

Le codage du mouvement avec pertes induit un biais dans la mesure objective de la qualité de la séquence reconstruite. La mesure de qualité objective la plus utilisée est le PSNR (peak signal to noise ratio) calculée entre les images de la séquence originale et les images de la séquence reconstruite. Cette mesure est basée sur la différence de luminance d'un pixel de l'image originale et de l'image reconstruite.
Lorsque le mouvement est codé avec pertes, ce calcul est biaisé par la distorsion sur le mouvement. En effet, avec un mouvement codé avec pertes, les textures des images peuvent être déplacées, ce déplacement n'est pas visible à l'œil mais fausse le calcul du PSNR. La figure 3.3.4 montre un exemple de séquence reconstruite avec un mouvement codé avec pertes. La qualité visuelle de la séquence reconstruite est bonne mais le PSNR est biaisé (PSNR de 20dB) à cause du mouvement avec pertes.

L'optimisation débit/distorsion de notre codeur est faite séparément pour chaque information (mouvement et texture). Pour juger de la qualité des séquences reconstruites, il est alors plus judicieux de mesurer la distorsion de la séquence reconstruite dans chaque domaine.
Dans une certaine mesure, la distorsion du mouvement n'est pas détectable par le système visuel humain. La qualité de la texture représente la qualité visuelle de la séquence. Nous considérons donc que l'erreur sur le mouvement est négligeable et nous calculons le PSNR dans le domaine texture entre les images de texture originales et les images de texture décodées. Ce PSNR permet d'avoir une mesure objective de la qualité visuelle de la séquence reconstruite.
On peut aussi mesurer la qualité objective de la séquence reconstruite par rapport à une séquence référence reconstruite comme évoqué dans [N6373 04]. Cette séquence référence est construite à l'aide de notre technique de codage et représente la séquence reconstruite par notre codeur avec un codage sans pertes sur la texture.

3.4 Fonctionnalités

3.4.1 Scalabilité

3.4.1.1 Scalabilité temporelle

Dans les schémas de codage vidéo avec transformée ondelettes temporelle, la scalabilité temporelle peut être aisément obtenue en découpant le volume des informations selon l'axe temporel. Les informations sont codées de manière progressive, de sorte que les hautes fréquences temporelles non souhaitées ne sont pas décodées. Par exemple, pour une séquence codée à 60Hz, si la fréquence désirée au décodage est de 30Hz, il suffit de ne pas décoder les premières hautes fréquences issues de la décomposition temporelle, la séquence décodée et synthétisée temporellement correspond alors aux basses fréquences temporelles issues du premier niveau de décomposition. D'autres fréquences temporelles plus basses peuvent être obtenues en utilisant les basses fréquences des niveaux supérieurs et en écartant les hautes fréquences de ces niveaux. Cette technique

1. originale

2. reconstruite

3.Image de différences entre originale et reconstruite

FIG. 3.13 – *Séquence St Sauveur, séquence reconstruite avec mouvement codé avec pertes*

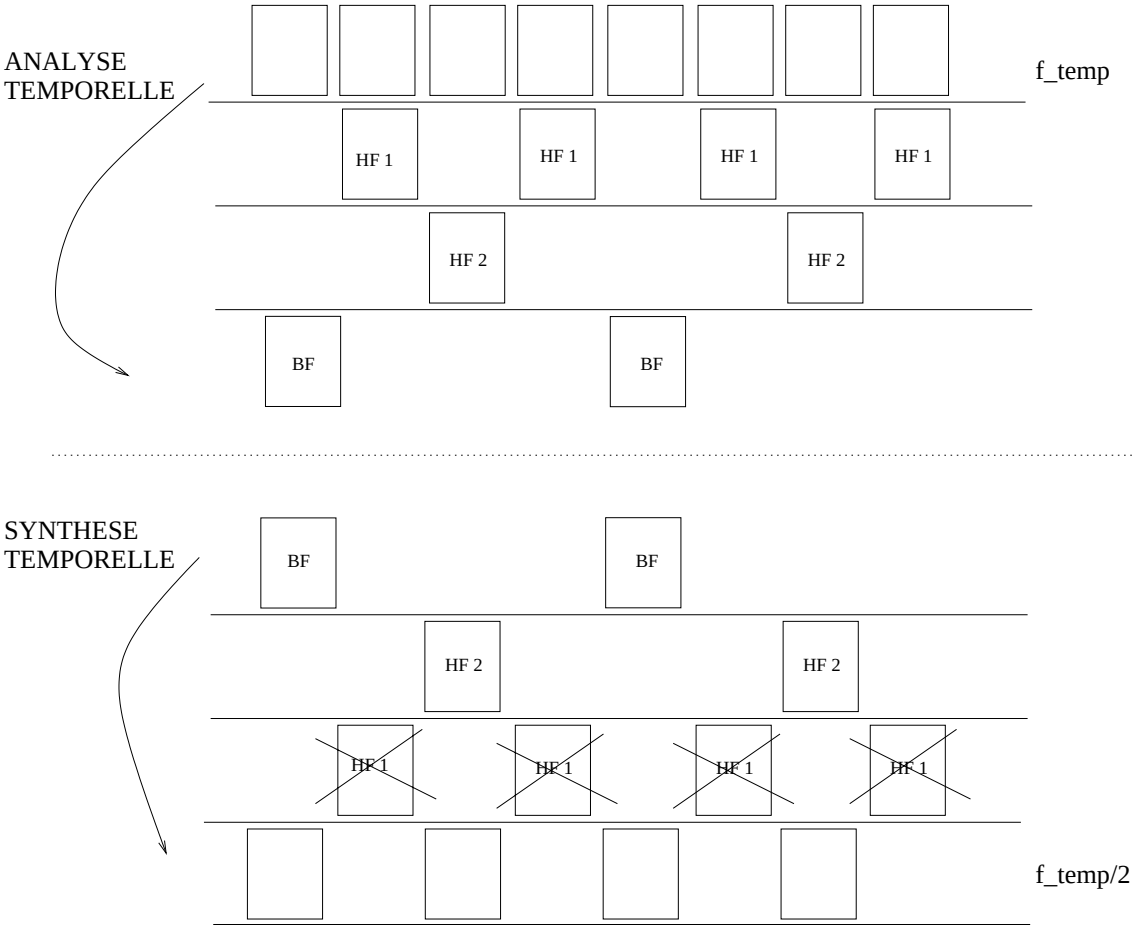


FIG. 3.14 – *Obtention de la scalabilité temporelle classique, à l'étape de synthèse*

est présentée sur la figure 3.14.
Cependant, cette technique provoque des artéfacts visuels sur les séquences reconstruites à plus basse fréquence temporelle, appelés *ghosting*. Ces effets de ghosting se traduisent par des traces, des voiles fantômes apparaissant dans les basses fréquences. Nous pensons que la technique d'écarter les hautes fréquences pour obtenir la scalabilité temporelle nuit trop à la qualité de la séquence reconstruite et nous préférons utiliser toutes les fréquences codées au moment du décodage; par exemple pour une séquence codée à 60Hz et une séquence reconstruite à la fréquence temporelle de 30Hz, nous décodons toutes les fréquences codées à 60Hz, ces fréquences sont synthétisées et la séquence reconstruite est alors à 60Hz. Un sous-échantillonnage temporel est possible si la fréquence temporelle doit être à 30Hz mais nous offrons la possibilité de pouvoir quand même visualiser une séquence à 60Hz. Cette technique est présentée par la figure 3.15.

Les tests que nous avons effectués montrent que les hautes fréquences temporelles

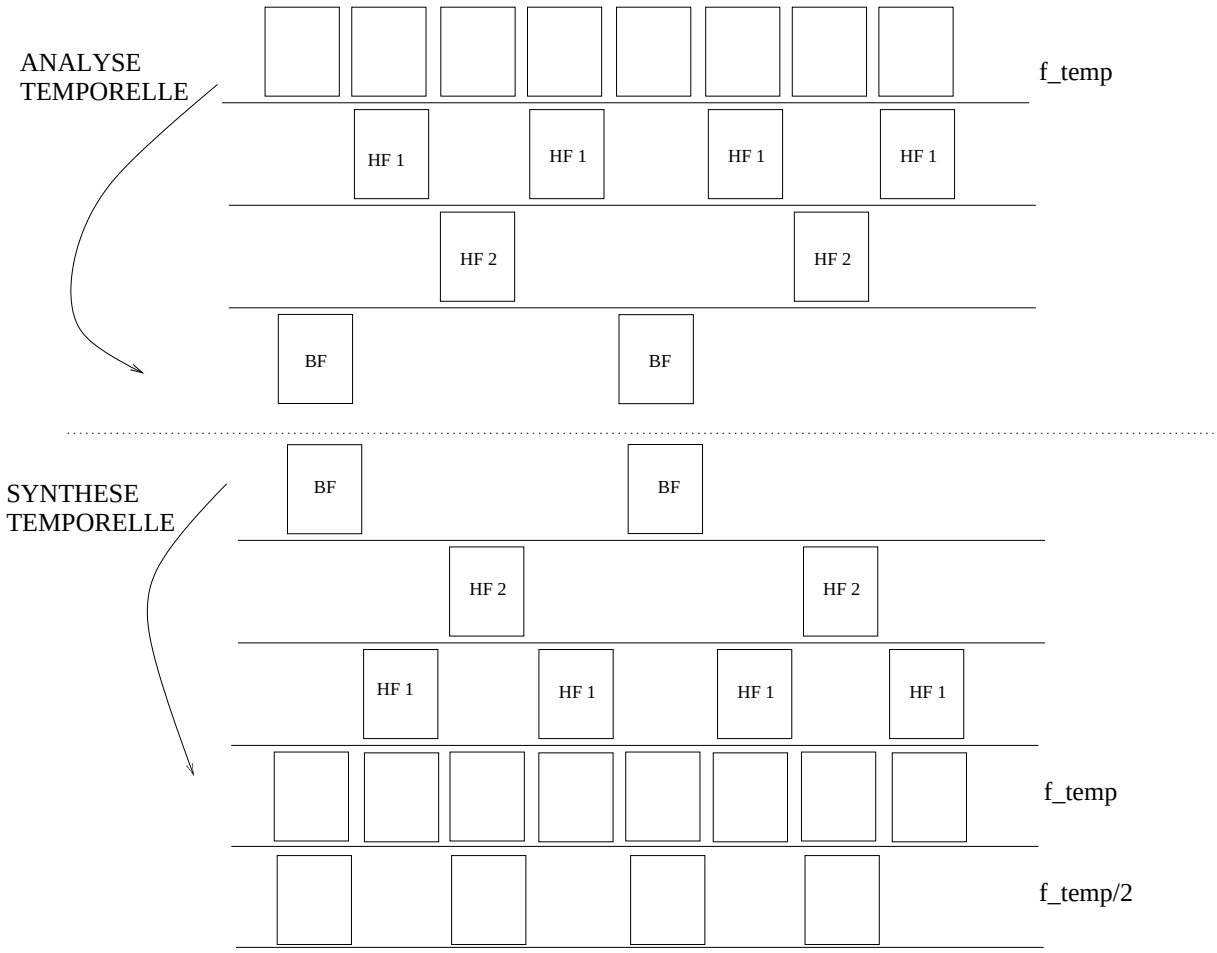


FIG. 3.15 – *Obtention de la scalabilité temporelle après la synthèse*

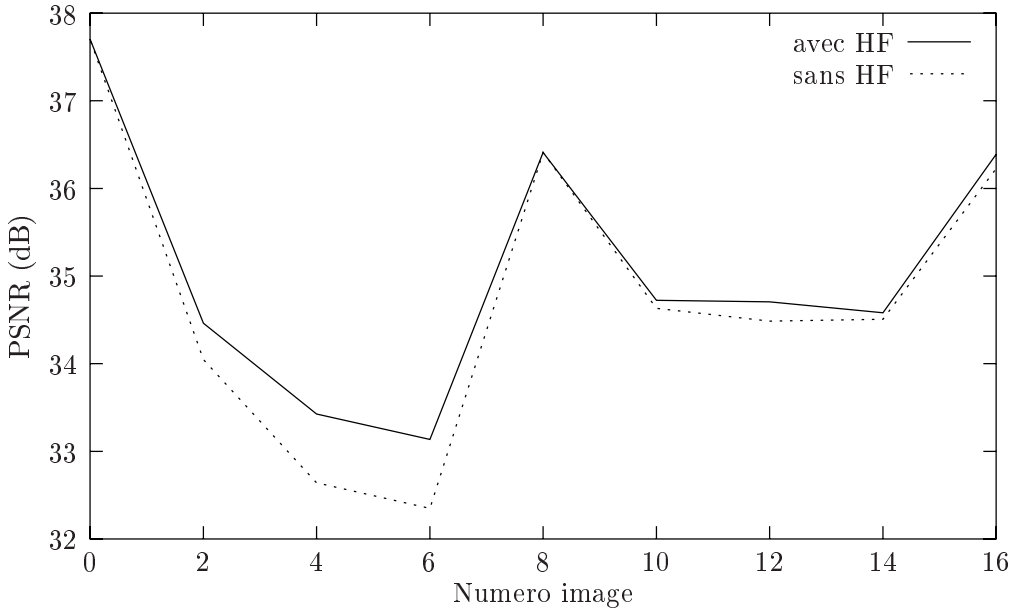


FIG. 3.16 – *PSNR des textures synthétisées avec et sans les hautes fréquences*

qui sont écartées au décodage par la première technique présentée contiennent de l'information utile pour les basses fréquences. La figure 3.16 comparent les PSNR des textures synthétisées après un décodage sans les premières hautes fréquences et avec les premières hautes fréquences. Les textures ont été transformées temporellement à l'aide d'un filtre ondelette 5/3 lifting, les sous-bandes temporelles ont été codées par un codeur JPEG2000. Le PSNR n'a été calculé que sur les images de texture permettant une reconstruction de la séquence à une fréquence temporelle plus basse. Seize images de textures ont été codées, la technique écartant les hautes fréquences n'a décodé que huit images de texture, ces huit images sont les basses fréquences du premier niveau de décomposition. La deuxième technique a décodé toutes les sous-bandes. Le débit total pour toutes les images était de 100kb, la première technique disposait donc de 12,5kb par image, tandis que la deuxième technique disposait de 6.25kb par image.
La figure 3.16 montre que bien que la deuxième technique disposait de moins de débit pour décoder les basses fréquences issues du premier niveau de décomposition, les textures synthétisées sont de meilleure qualité que celles obtenues par la première technique qui disposait de plus de débit pour les mêmes basses fréquences. Cette différence de performance est due au fait que les hautes fréquences temporelles écartées par la première technique contiennent de l'information utile à la synthèse des basses fréquences. Le débit donné aux hautes fréquences écartées n'est pas perdu puisqu'il sert à la synthèse des basses fréquences reconstruites même si les images de texture des hautes fréquences ne sont pas visualisées.

Un avantage de notre technique de codage est la possibilité de reconstruire une

séquence à une fréquence temporelle plus élevée que la fréquence temporelle originale. La représentation par maillages du mouvement permet d'avoir une continuité temporelle du mouvement et de la texture sur un GOF entier. Il est alors possible d'interpoler linéairement dans chaque domaine une nouvelle image de texture et un nouveau champ de mouvement. La texture interpolée est ensuite projetée sur sa grille dans le domaine image à l'aide du mouvement interpolé. Ceci permet d'augmenter la fluidité d'une séquence vidéo en ajoutant des images par rapport à la séquence originale, sans créer d'artéfacts, ni de flou dans la séquence reconstruite.

3.4.1.2 Scalabilité spatiale

Dans un schéma de codage vidéo par ondelettes t+2D, la scalabilité spatiale peut être obtenue comme pour la scalabilité temporelle par un découpage du volume d'informations le long des axes spatiaux. Cependant, la structure en deux couches de notre schéma de codage impose que le décodage et la transformation ondelette inverse des images se fassent au même niveau de résolution que l'analyse. Les textures décodées sont ensuite sous-échantillonnées, le mouvement est mis à l'échelle de la résolution de restitution. Les textures résultantes sont projetées à l'aide du mouvement sur leur grille originale à la résolution de décodage désirée.

3.4.1.3 Scalabilité à grain fin

La scalabilité à grain fin est obtenue grâce au codage progressif des informations de texture et de mouvement. La couche haute est codée par un codeur JPEG2000 pour les informations de texture et par un codeur arithmétique par plans de bits avec optimisation débit/distorsion pour les informations de mouvement.

3.4.1.4 Scalabilité en complexité opératoire

Pour notre schéma, nous avons vu que pour les scalabilités spatiale et temporelle, nous utilisons toutes les sous-bandes du volume d'informations codées. La transformée inverse t+2D est effectuée pour tous les niveaux de décomposition, ainsi la complexité opératoire du décodeur est la même quelle que soit la fréquence temporelle et la résolution spatiale de décodage. Cette technique nous permet d'avoir une meilleure qualité visuelle globale de la séquence reconstruite que la technique de coupure des hautes fréquences. Cependant, elle ne nous permet pas d'avoir une scalabilité au niveau de la complexité opératoire du décodage.
La scalabilité en complexité opératoire influe sur la qualité de la séquence reconstruite. Dans le cas où cette complexité doit être prise en compte, notre schéma basculerait vers les techniques classiques de coupure des hautes fréquences. Beaucoup de schémas proposés dans la littérature utilisent cette technique pour obtenir la scalabilité, cependant cette technique entraîne un effet de dérive lors de l'utilisation du mouvement dans une

transformée spatio-temporelle. Dans le cas d'un traitement de la séquence par GOF, de taille réduite, cette dérive reste acceptable car elle reste limitée à la taille du GOF.

3.4.2 Mode objet

Le codeur proposé est compatible avec une représentation objet de la vidéo. Par rapport à l'approche en mode rectangulaire, le codeur en mode objet nécessite la connaissance de la segmentation de la vidéo en objet qui lui est fournie sous la forme de masques binaires. Le codeur fonctionne alors comme en mode rectangulaire mais en ne considérant que la partie définie par le masque de segmentation.
Pour un codage de plusieurs objets dans une séquence vidéo, le codeur est alors appelé sur chaque objet indépendamment.
L'estimation de mouvement de l'objet est effectuée sur le support de l'objet, les images de textures et les informations de mouvement sont décorrélées et codées comme dans le schéma classique.
L'information des masques de segmentation doit être connue au moment de la synthèse lors de la recomposition de la scène. Pour cela, le contour de l'objet est codé à l'aide du codeur de contour introduit dans [Chaumont 03]. Les contours sont extraits des masques de segmentation, puis alignés et complétés. Les contours complétés sont ensuite codés à l'aide d'une transformation ondelettes suivie d'un codage arithmétique par plans de bits. La technique de codage du contour utilise une approche long terme afin d'avoir un codage efficace et progressif mais également pour disposer d'une stabilité temporelle du contour au cours de la séquence.
Sur certaines séquences, le codage vidéo par objets offre de meilleurs résultats que le codage vidéo classique. Cependant, ces performances sont très dépendantes de la qualité de la segmentation des objets.

Conclusion

Dans ce chapitre, nous avons présenté les contributions apportées par cette étude dans le cadre de codage vidéo scalable par maillages et ondelettes t+2D. Nous avons proposé une technique de projection de la texture sur des grilles d'échantillonnage de référence ainsi qu'une technique de définition de la texture dans les zones non définies. Ces deux techniques permettent de définir les trajectoires de mouvement de manière efficace afin de les exploiter par une transformation en ondelettes temporelles. La transformée en ondelettes temporelles a été mise en œuvre dans sa forme lifting et les sous-bandes temporelles ont été transformées spatialement de manière adaptative selon leur nature.
Une étude sur l'influence du codage du mouvement avec pertes sur la mesure objective de la qualité de la séquence reconstruite a été menée et une nouvelle mesure de qualité a été proposée. Enfin, nous avons présenté la mise en œuvre de la scalabilité au sein du codeur, ainsi que le mode objet.
Le prochain chapitre présente les résultats de ce codeur. Il valide les choix que nous

avons faits et montrent les performances de notre codeur en terme de compression ainsi qu'en terme de scalabilité.

Chapitre 4

Résultats de codage

Ce chapitre présente les résultats de codage donnés par le codeur proposé dans le chapitre précédent. Il valide les choix que nous avons faits au niveau de chacune des briques du processus de codage. La première section compare les techniques proposées pour l'étape d'analyse, la deuxième section traite de la transformée temporelle. La troisième section étudie différents types de codage et codeurs possibles pour les images que nous avons à coder.
La deuxième partie des résultats, présentée dans la quatrième section, positionne notre codeur par rapport à d'autres codeurs, scalables et non scalable. Enfin, la dernière section présente brièvement quelques résultats de notre codeur en mode objet.

4.1 L'analyse

4.1.1 Comparaison des trois techniques de projection

Cette sous-section compare les résultats de codage obtenus par les trois techniques de projection présentées dans le chapitre précédent. La première technique projette toutes les images sur la première image du GOF, la deuxième technique projette chaque image sur elle-même et la troisième technique projette les images sur la première et la dernière moitié du GOF. Les tests ont été effectués sur 80 images des séquences Foreman à 256 kbp/s et Mobile And Calendar à 512kbp/s au format CIF YUV 4:2:0 à 30Hz. Les GOFs sont de taille 8. La comparaison est faite à l'aide du PSNR calculé entre les séquences originales et les séquences reconstruites pour chaque technique. Les figures 4.1 et 4.2 présentent les performances obtenues. Les mesures de PSNR montrent que les deux dernières techniques reconstruisent mieux la dernière image de chaque GOF que la première technique, ce qui était un de nos objectifs. Ce résultat est confirmé par la figure 4.3 qui montre la dernière image reconstruite du premier GOF par chaque technique.
Sur les images reconstruites, on peut observer des zones où la projection de l'image sur une référence lointaine a déformé de manière irréversible la texture, créant des motifs dégénérés lors de la synthèse de l'image. Ce phénomène est visible sur l'image 1. de la figure 4.3, sur le calendrier, près du ballon.

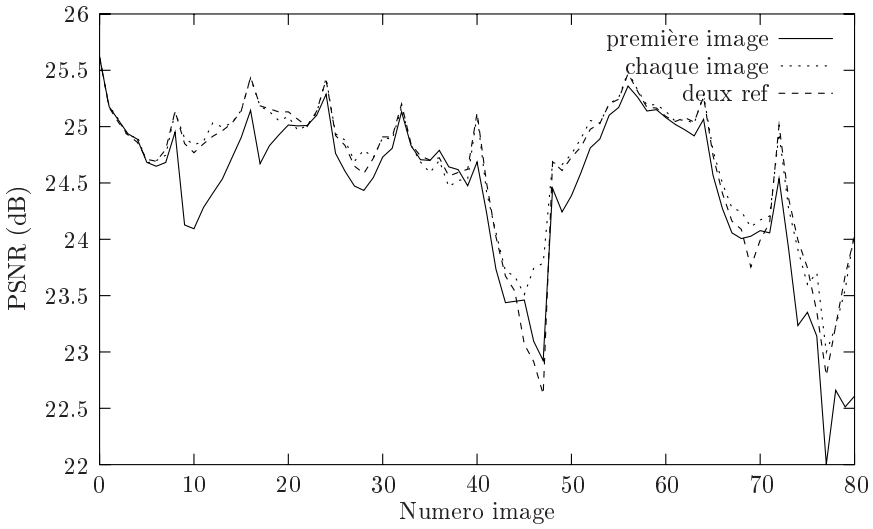


FIG. 4.1 – *Séquence Foreman à 256kbps*

Au niveau PSNR, la deuxième technique offre sensiblement de meilleures performances que la troisième technique à deux références, cependant, dans certains cas, cette différence n'est pas détectable par l'oeil humain sur les images reconstruites, figure 4.4. De plus, la deuxième méthode limite la scalabilité du mouvement car elle en est très dépendante.

4.1.2 Efficacité du padding spatio-temporel

Les trois techniques de padding des mosaïques ont été testées. La première technique effectue le padding spatial de chaque mosaïque indépendamment des autres. La deuxième technique effectue un padding spatial des mosaïques aux extrémités de GOF et utilise une interpolation linéaire pour les mosaïques intérieures. Enfin, la troisième technique effectue un padding spatio-temporel par analyse-synthèse étendu sur deux GOFs.
Les séquences Foreman et Mobile And Calendar ont été codées et décodées aux débits respectifs de 256kbps et 512kbps. Les images de la figure 4.5 donnent l'allure des sous-bandes issues de la décomposition temporelle pour chaque technique de padding. Les sous-bandes qui présentent le moins de hautes fréquences sont celles issues du padding spatio-temporel étendu. Ces sous-bandes seront a priori les plus faciles à coder.
Les courbes 4.6 et 4.7 comparent les mesures de PSNR texture pour chaque technique. Le PSNR texture est mesuré entre les images mosaïques issues du padding et les images mosaïques décodées sur les parties définies dans l'image . Cette mesure nous permet d'étudier le comportement du codeur par rapport à l'allure du signal des mosaïques. Les mesures de PSNR confirment le fait que les sous-bandes temporelles issues du padding spatio-temporel étendu sont plus faciles à coder.

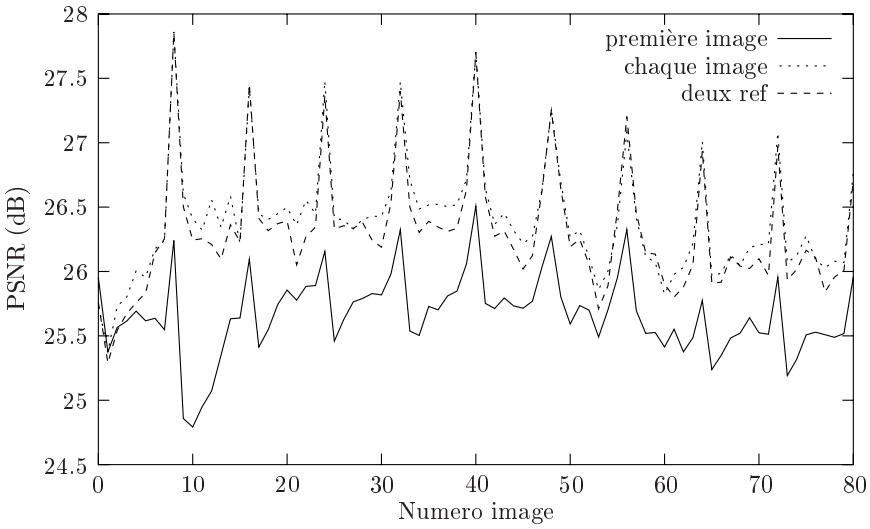


FIG. 4.2 – *Séquence Mobile And Calendar à 512kbps*

Nous observons de plus sur la figure 4.5, sur les sous-bandes hautes fréquences de niveau 2 et les basses fréquences, que des hautes fréquences artificielles sont introduites autour du cadre de l'image pour les deux premières techniques de padding. Ces hautes fréquences sont nettement moins présentes dans les sous-bandes obtenues par la technique de padding spatio-temporel par analyse-synthèse.

4.2 Transformée temporelle

4.2.1 Influence des filtres de transformée

Les figures 4.8 et 4.9 comparent les performances de décorrélation temporelle de trois filtres de transformée: 5/3, 5/3 tronqué et le 9/7. Les séquences Foreman et Mobile And Calendar ont été codées à 512kbps à 30Hz, les mesures de PSNR texture sont données par les courbes pour chaque filtre de transformée. Les filtres 5/3 et 9/7 donnent des résultats similaires meilleurs que le 5/3 tronqué.
Contrairement à ce qui était annoncé dans [Vieron 02], le filtre 9/7 donne des résultats équivalents à ceux du filtre 5/3. Ceci est du à la qualité de mouvement utilisé. En effet, le mouvement par maillage offre des sous-bandes temporelles d'une nature plus facile à coder que les mouvements par blocs qui créent des hautes fréquences artificielles aux frontières des blocs qui coûtent cher à coder.

1.


2.


3.


FIG. 4.3 – *Mobile And Calendar: reconstruction de la dernière image du premier GOF (image 8): 1. projection sur la première image, 2. projection sur chaque image, 3. projection sur la première et la dernière image*

FIG. 4.4 – *Mobile And Calendar: images reconstruites à 512kbps*

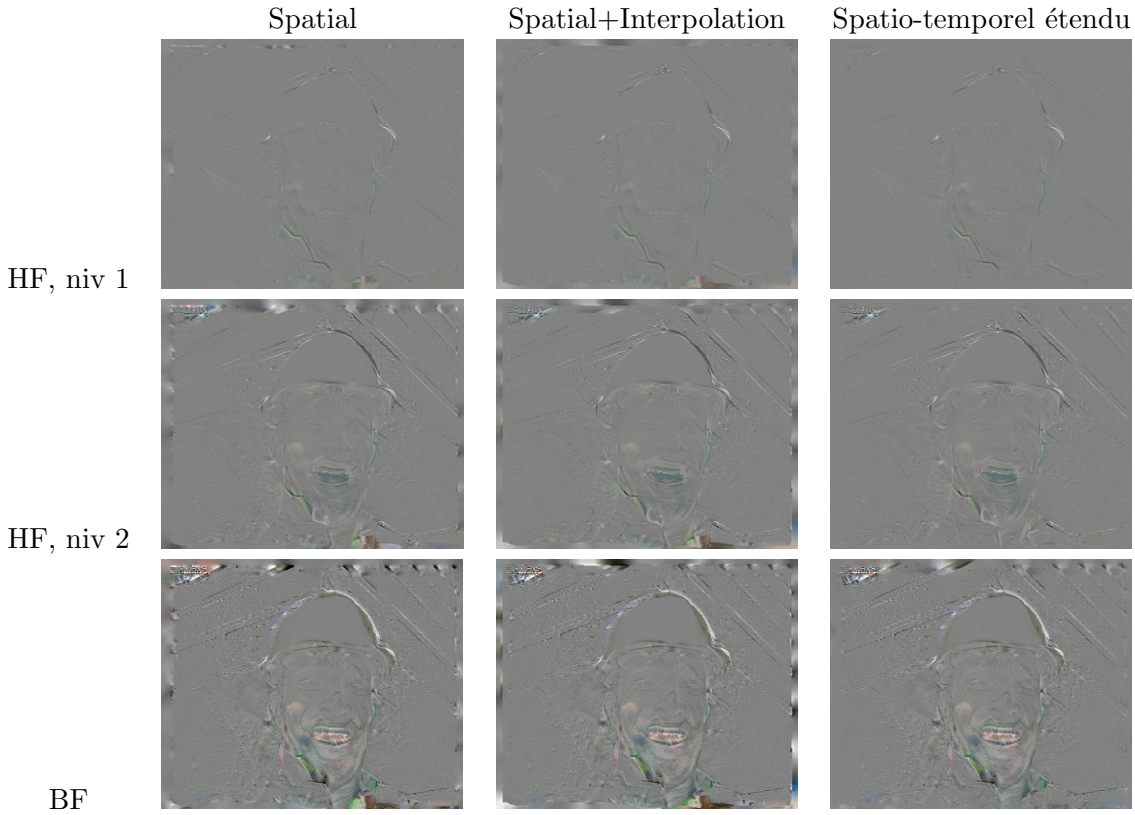


FIG. 4.5 – *Comparaison des sous-bandes temporelles issues des techniques de padding*

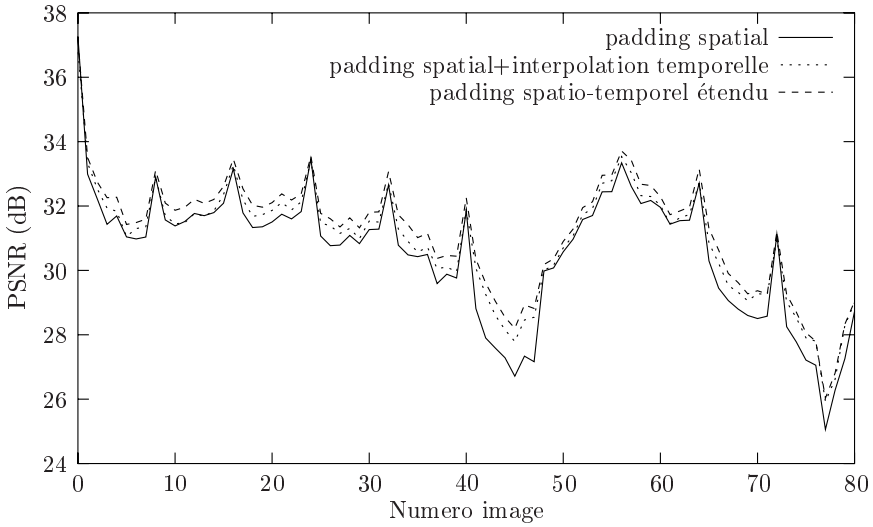


FIG. 4.6 – *Comparaison des techniques de padding, PSNR texture, séquence Foreman à 256kbps*

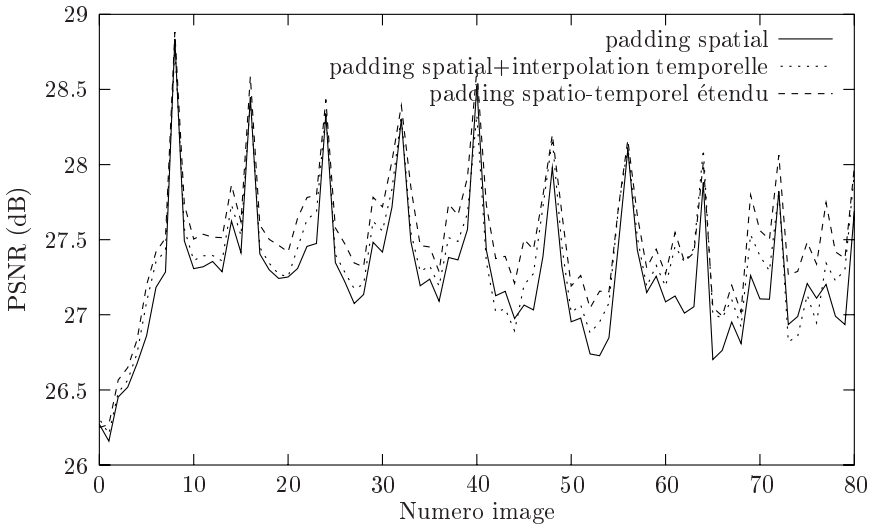


FIG. 4.7 – *Comparaison des techniques de padding, PSNR texture, séquence Mobile And Calendar à 512kbps*

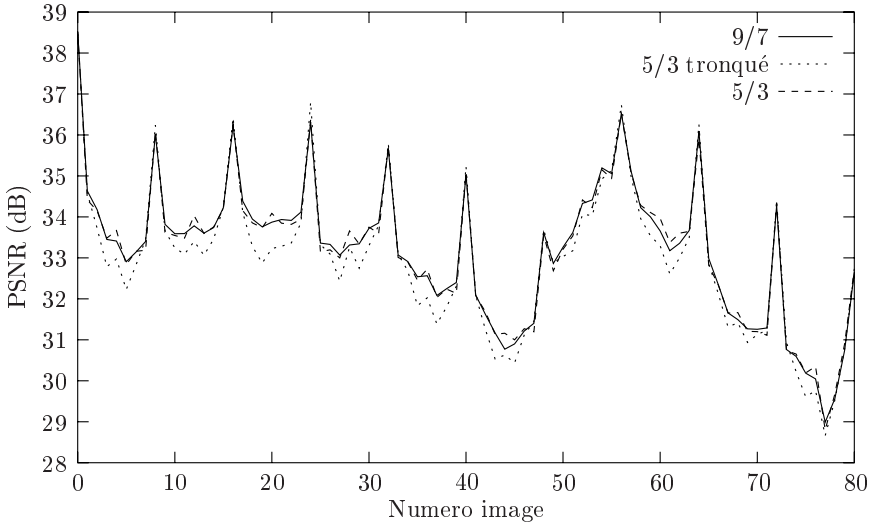


FIG. 4.8 – *Comparaison des filtres de transformée temporelle, PSNR texture, séquence Foreman à 512kbps*

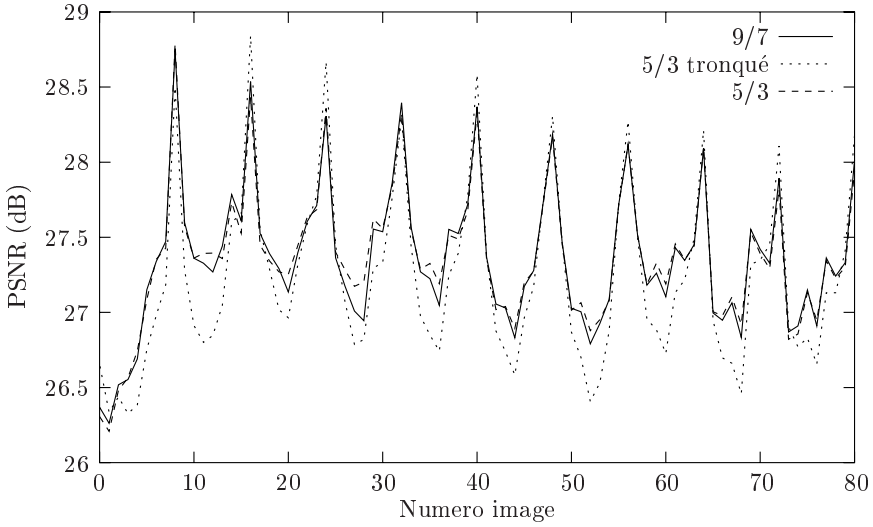


FIG. 4.9 – *Comparaison des filtres de transformée temporelle, PSNR texture, séquence Mobile And Calendar à 512kb/s*

4.3 Codage des images

4.3.1 Codage des images clés

4.3.1.1 Codage INTRA-INTER et INTRA-INTRA

La couche basse contient les informations de mouvement et de texture des images extrêmes d'un GOF. Dans le cas d'un GOF intra, la couche basse code les informations de la première image du GOF codée en intra et les informations de la dernière image du GOF codée en mode inter ou mode intra. Dans le cas d'un GOF inter, la couche basse contient les informations de la dernière image du GOF codée en inter ou en intra. Les GOFs sont recouvrants donc le seul GOF intra de la séquence est le premier GOF. Cette sous-section compare le codage de la couche basse en mode Intra-inter ou Intra-intra. Le mode Intra-inter code la première image de la séquence en intra et toutes les images de fin de GOF par prédiction par rapport à la première image du GOF. Le mode Intra-Intra code la première image de la séquence en intra et toutes les images de fin de GOF en intra également. Pour la dernière image d'un GOF, le débit utilisé en mode inter et en mode intra est le même.
Les figures 4.10 et 4.11 montrent les PSNR texture respectivement pour les séquences Bus et Mobile And Calendar codées à 512kb/s, dont 80kb/s pour la couche basse codée à 30Hz, les GOFs contenant neuf images, le débit alloué à la dernière image du GOF est de 24kb et le codeur utilisé pour la dernière image est le codeur JPEG2000. Les courbes montrent que le codage Intra-intra est meilleur pour la séquence Bus que le codage Intra-inter. Cependant, ce constat s'inverse pour la séquence Mobile And Calendar.

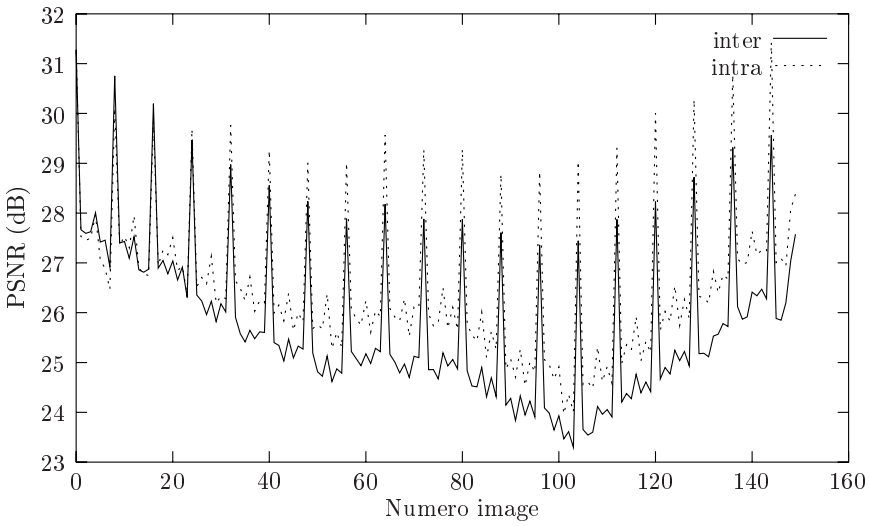


FIG. 4.10 – *Codage de la couche basse en mode INTRA-INTER ou INTRA-INTRA: comparaison du PSNR texture, séquence Bus à 512kbps*

Ces résultats s'expliquent par le fait que dans la séquence Bus, le mouvement par maillage décroche facilement, entraînant une mauvaise prédiction de la dernière image du GOF. La dernière image du GOF est alors plus facile à coder en mode intra que l'erreur de prédiction. Dans la séquence Mobile And Calendar, le mouvement par maillage est bien capté et la prédiction de la dernière image donne une faible image d'erreurs qui est alors plus facile à coder que l'image intra. Cette étude montre que le codage en mode intra de la dernière image d'un GOF n'est intéressant que dans le cas où le mouvement n'est pas bien capté dans la séquence.

4.3.1.2 Codage JPEG2000, EZBC

Le tableau 4.1 compare le codage des séquences Mobile And Calendar et Foreman lorsque la première image en INTRA est codée par un codeur JPEG2000 ou EZBC. Le débit disponible pour l'image INTRA est 90kb, le débit total pour la séquence est de 512kb/s. Le tableau donne le PSNR texture de la première image reconstruite, le codeur EZBC donne des résultats légèrement meilleur que le codeur JPEG2000 sur le codage intra.
Le tableau 4.2 donne les PSNR texture moyens des images de fin de GOF codées par un codeur JPEG2000 et un codeur EZBC, ainsi que les PSNR texture moyen sur toute la séquence. Le codeur JPEG2000 donne de meilleurs résultats que le codeur EZBC.
Ces résultats s'expliquent par les techniques d'optimisation débit/distorsion utilisées dans chaque codeur. Le codeur EZBC utilise un codage par plan de bits tandis que le codeur JPEG-2000 effectue une optimisation débit/distorsion en considérant des

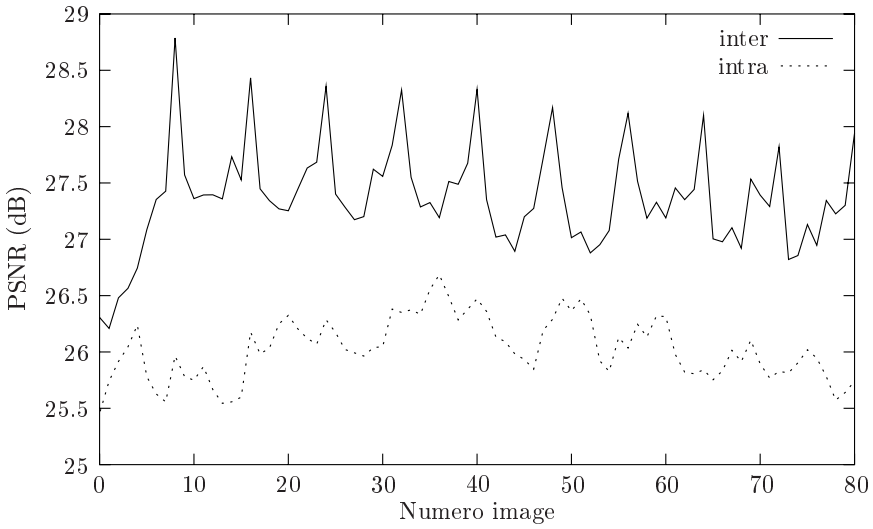


FIG. 4.11 – *Codage de la couche basse en mode INTRA-INTER ou INTRA-INTRA: comparaison du PSNR texture, séquence Mobile And Calendar à 512kbps*

		PSNR Image
Mobile	JPEG2000	26.30 dB
	EZBC	26.51 dB
Foreman	JPEG2000	38.52 dB
	EZBC	38.70 dB

TAB. 4.1 – *Comparaison codage des images INTRA par JPEG2000 et EZBC*

couches de qualité et en calculant la contribution de chaque sous-bande à chaque couche. Les images intra ont une répartition homogène de l'énergie pour laquelle un codage par plan de bits, qui considère tous les échantillons avec la même importance, est mieux adapté; alors qu'une optimisation débit/distorsion telle que celle de JPEG-2000 se trouve pénalisée par le coût de l'information de la répartition des couches.
Les images d'erreur sont des images où l'énergie est très localisée sur des pics, et très faibles dans de grandes zones homogènes. Dans ce cas, l'optimisation débit/distorsion de JPEG-2000 qui calcule la contribution de chaque code-bloc de l'image à chaque couche de qualité est mieux adaptée qu'un simple codage par plan de bits.

4.3.2 Codage des sous-bandes temporelles

4.3.2.1 Codage JPEG2000, EZBC

Cette sous-section compare le codage de la couche haute par un codeur JPEG2000 et par un codeur EZBC. La couche haute est composée de deux types d'images: les sous-bandes temporelles issues de la décomposition temporelle et les résidus de codage

		PSNR image	PSNR sequence
Mobile	JPEG2000	28.13 dB	27.46 dB
	EZBC	27.69 dB	26.53 dB
Foreman	JPEG2000	35.55 dB	33.33 dB
	EZBC	35.50 dB	33.12 dB

TAB. 4.2 – *Comparaison codage des images INTER par JPEG2000 et EZBC*

des images aux extrémités de GOF. Toutes ces images sont envoyées à chaque codeur qui fonctionne alors en multi-composantes. La figure 4.12 montre le PSNR calculé sur les images de la couche haute. Les images sont classées par type: basses fréquences, hautes fréquences niveau 1, hautes fréquences niveau 2 et résidus de codage. Les courbes montrent que les sous-bandes temporelles sont mieux codées par le codeur EZBC que par JPEG2000. Le constat inverse est fait pour les images de résidus. La figure 4.13 pour la séquence Mobile And Calendar confirme ces résultats.
Cependant, la mesure du PSNR texture (figure 4.14) montre que le codage de la couche haute par JPEG2000 donne de meilleurs résultats sur la séquence reconstruite que le codage par EZBC bien que ce dernier reconstruise mieux les sous-bandes temporelles. Ceci est du à la prédiction des images mosaïques par les images aux extrémités de GOF. Le codage des résidus des images extrêmes dans la couche haute influe sur toutes les images. Ainsi, l'erreur de codage par EZBC sur les résidus se propage à toutes les images de texture, donnant une moins bonne reconstruction de la texture que JPEG2000.
Ces observations ouvrent la perspective d'un codage mixte de la couche haute par EZBC et JPEG-2000. Les sous-bandes temporelles seraient codées par EZBC et les erreurs de codage par JPEG-2000. Cependant, la répartition du débit entre ces deux codeurs n'est alors pas aisée. En l'état actuel, le codeur n'utilise que le codeur JPEG-2000 auquel toutes les sous-bandes sont passées. Le codeur JPEG-2000 effectue une optimisation débit/distorsion globale sur toutes les sous-bandes.

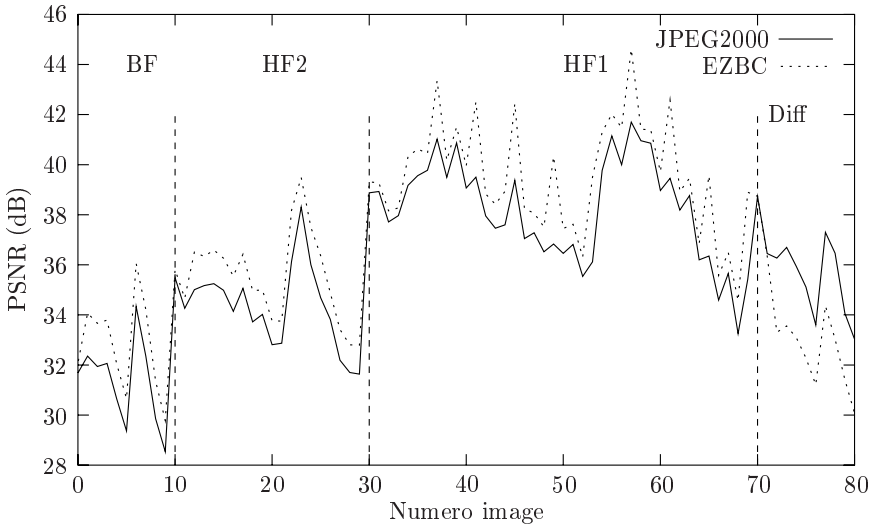


FIG. 4.12 – *Codage de la couche haute par JPEG2000 et EZBC: comparaison du PSNR calculé sur les images de sous-bandes temporelles, séquence Foreman à 512kbps, (BF=Basses Fréquences, HF1= Premières Hautes Fréquences, HF2=Hautes Fréquences niveau 2, Diff=Résidu de codage des images extrêmes*

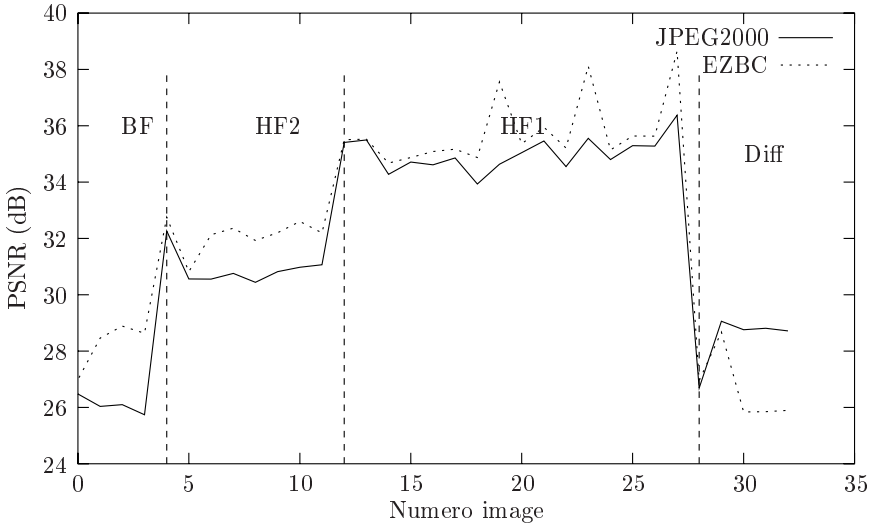


FIG. 4.13 – *Codage de la couche haute par JPEG2000 et EZBC: comparaison du PSNR calculé sur les images de sous-bandes temporelles, séquence Mobile And Calendar à 512kbps*

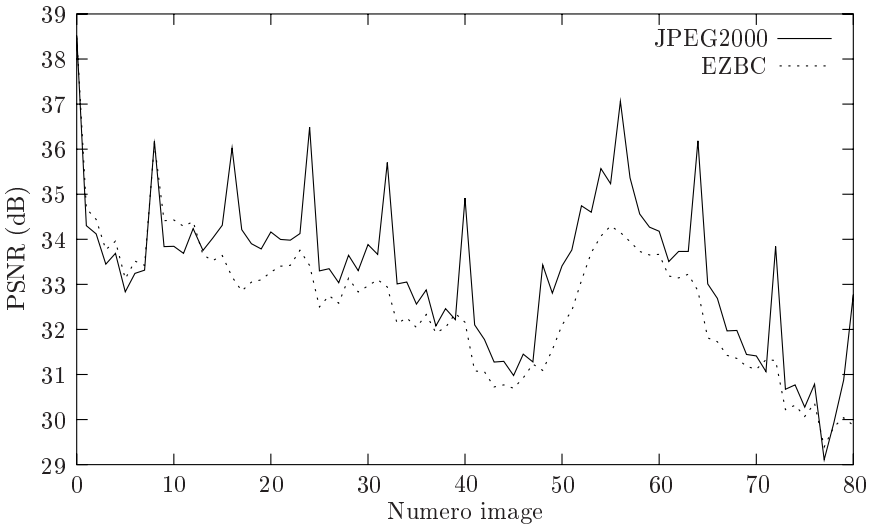


FIG. 4.14 – *Codage de la couche haute par JPEG2000 et EZBC: comparaison de PSNR texture, séquence Foreman à 512kbps*

4.4 Positionnement par rapport à d'autres codeurs

4.4.1 Placement par rapport à un codeur non scalable-AVC

Afin d'évaluer l'efficacité en compression de notre codeur, nous avons comparé ses performances à celles d'un codeur non scalable issu de l'état de l'art: H264 AVC. Deux versions de ce codeur ont été utilisées: le JM 4.2 et le JM8.1. Le tableau 4.3 présente les options utilisées pour chaque version du codeur H264.
Le tableau 4.4 liste les séquences utilisées pour la comparaison et précise les formats respectifs de ces séquences et les débits utilisés. Les codeurs H264-AVC ont effectué des codages séparés des séquences pour chaque débit.
Pour notre approche, notre codeur vidéo a codé une fois chaque séquence à 30Hz avec un débit infini, puis décodé successivement aux débits et fréquences temporelles indiqués. Les paramètres d'encodage sont les suivants: les images sont codées sur deux couches, par groupe de 8 images (GOF). Le premier GOF est Intra, les GOFs suivants sont tous codés en mode Inter (une image est partagée avec le GOF suivant). La couche basse est codée à 64 kbit/s avec un rafraîchissement Intra toutes les secondes. La couche haute est codée à débit infini. Le codage est fait à débit constant.
Les courbes débits/distorsion sont données par les figures 4.15, 4.18, 4.21 et 4.25. La mesure de distorsion utilisée est le PSNR. Le PSNR a été calculé entre les images originales de la séquence vidéo et les images reconstruites pour le codeur AVC et pour notre approche (courbe appelée PSNR image). Nous avons également calculé un PSNR dans le domaine texture (courbe appelée PSNR texture). En effet, pour notre approche, le PSNR texture rend mieux compte de la qualité visuelle de la séquence reconstruite que le PSNR image.

Options	JM 4.2	JM 8.1
Contrôle de débit	oui	oui
RD optimisation	oui	oui
Options spécifiques	loop filter, CABAC	loop filter, CABAC
Nombre d'images références	4	5
Nombres d'images B	2	2
Rafraîchissement INTRA	1s	1s

TAB. 4.3 – *Profils des codeurs H264 utlisés*

Séquence	Résolutions		Débits (kbps)	Profil H264
	Spatiale	Temporelle		
St Sauveur	336x272	30Hz	50,100,150,500	JM 8.1
Tempête	CIF	30Hz	64,128,256,512,1024	JM8.1, JM4.2
Container	CIF	30Hz	128,256,512,1024	JM 4.2
Bus	CIF	15Hz	128,256	JM 4.2
		30Hz	512	JM 4.2

TAB. 4.4 – *Séquences, résolutions et débits utilisés pour la comparaison avec AVC*

La figure 4.15 montre que les performances de notre codeur par rapport au codeur H264 JM 8.1 sont meilleures en terme de PSNR à tous les débits pour la séquence St Sauveur. Les images extraites des séquences reconstruites données par les figures 4.16 et 4.17 confirment cette tendance.
Pour la séquence Tempête (figure 4.18), le codeur H264 JM 4.2 donne de meilleurs résultats que le codeur H264 JM 8.1 et notre approche. A bas débits, on remarque que le codeur H264 JM 8.1 ne peut pas descendre sous 300kbit/s, les performances du JM 4.2 et de notre approche sont sensiblement les mêmes en terme de PSNR. Les images de la figure 4.19 montrent qu'à bas débits, le codeur H264 JM 4.2 est légèrement meilleur que notre approche, l'image 192 pour notre approche présente quelques défauts dans le ciel bleu. Même avec un débit plus élevé (256kbit/s), le codeur H264 JM 8.1 n'atteint pas nos performances à 128 kbit/s. A hauts débits (1024 kbit/s), figure 4.20, les trois codeurs donnent des résultats similaires, avec un léger avantage pour notre approche. On remarque un peu plus de détails sur les images (par exemple sur le caillou situé dans le coin en bas à droite).
Pour la séquence Container (figure 4.21), les performances de notre codeur sont légèrement moins bonnes que celles du codeur H264 JM 4.2 sauf à 128 kbit/s où notre approche est légèrement meilleure. Les images extraites (figure 4.22 et 4.23) des séquences reconstruites à bas débits par notre approche montrent une meilleure finesse de détails que les images reconstruites par H264 JM 4.2 qui sont plus lisses, notamment dans l'eau. A hauts débits (figure 4.24), la différence de performances en terme de PSNR n'est pas

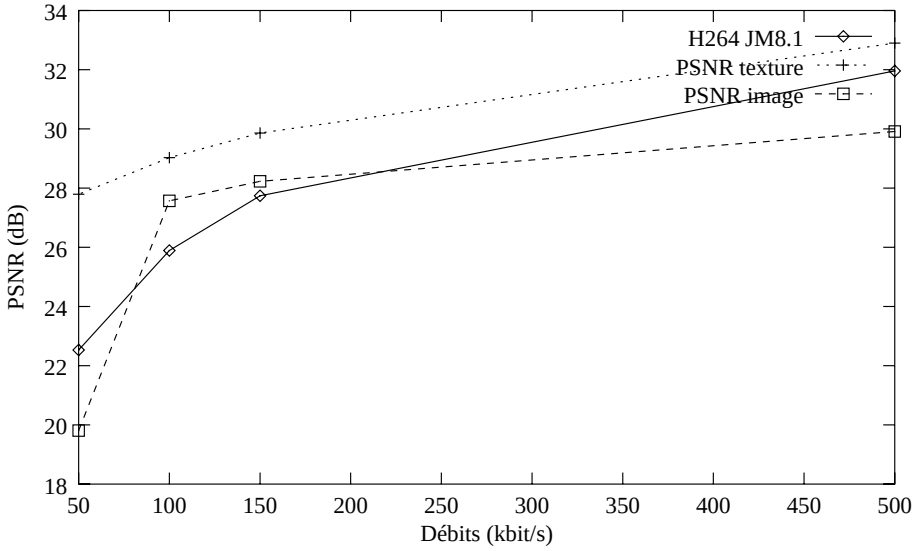


FIG. 4.15 – *PSNR sur la séquence St Sauveur*

visible sur les images, elles paraissent de la même qualité.
Les résultats donnés pour la séquence Bus montrent les limites de notre approche. En effet, les mesures PSNR (figure 4.25) ainsi que les images reconstruites (figure 4.26) montrent de très mauvais résultats sur cette séquence.
La séquence Bus est une séquence dont l'estimation du mouvement est très difficile à l'aide d'une technique par maillages. Le mouvement de translation du bus ne paraît pas a priori très compliqué mais le bus passe devant et derrière des objets auxquels les mailles restent accrochées. Ceci crée des étirements, voire des retournements de mailles rendant la prédiction des images dans ces zones très mauvaise. La qualité de la prédiction influe sur l'efficacité du codage. Pour cette raison, les résultats de codage pour cette séquence sont mauvais.

4.4.2 Codage scalable et placement par rapport à SVC

4.4.3 Présentation des codeurs utilisés pour la comparaison

Nous avons utilisés six codeurs vidéo scalables pour effectuer la comparaison. Les codeurs ont été proposés en réponse au CFP [cfp 03] sur le codage vidéo scalable. Ils peuvent être classés en deux catégories distinctes: les codeurs utilisant une technologie basée AVC et les codeurs utilisant une technologie basée ondelettes.
Parmi les codeurs basé AVC, nous avons testé les propositions faites par l'université de Poznan (proposition S13, [Blaszak 04]) et par Microsoft China (proposition S06, [Sun 04]).
La technique de codage de la proposition S06 est basée sur un codage en couches. Une couche de base code une version basse résolution spatiale, temporelle et basse qualité

FIG. 4.16 – *Séquence St Sauveur, à 50 kbit/s, image 50 et image 75*

FIG. 4.17 – *Séquence St Sauveur, à 500 kbit/s, image 50 et image 75*

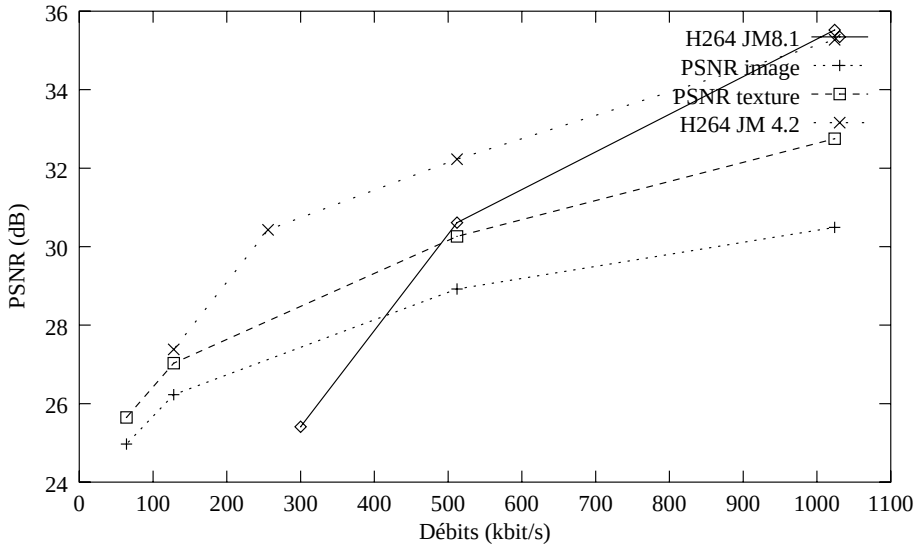


FIG. 4.18 – *PSNR sur la séquence Tempête*

FIG. 4.19 – *Séquence Tempête, à 128 kbit/s pour notre approche et H264 JM 4.2 et 256 kbit/s pour H264 JM 8.1, image 100 et image 192*

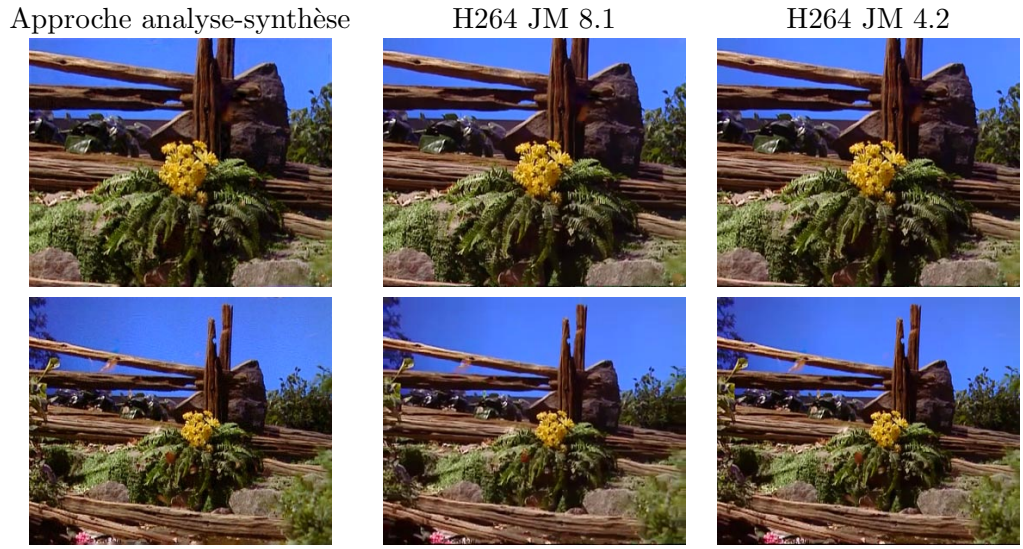


FIG. 4.20 – *Séquence Tempête, à 1024 kbit/s, image 100 et image 192*

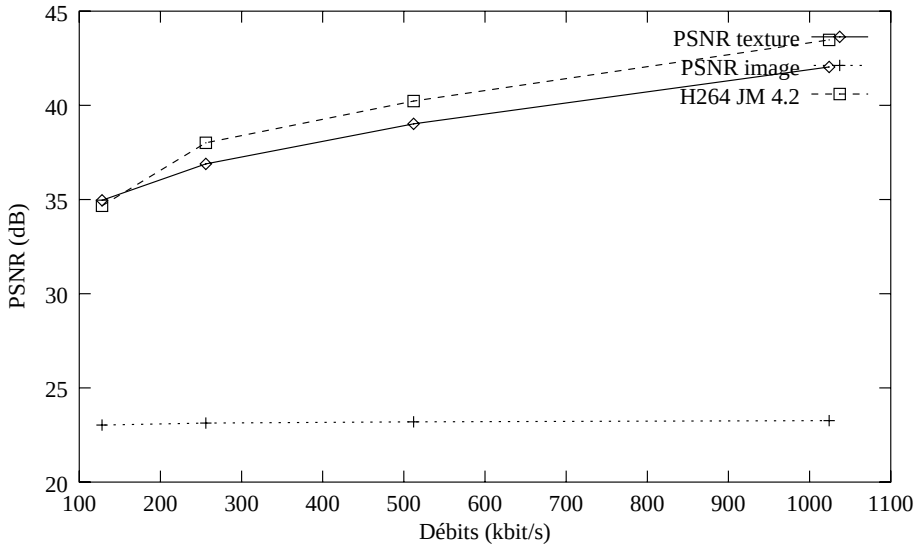


FIG. 4.21 – *PSNR sur la séquence Container*

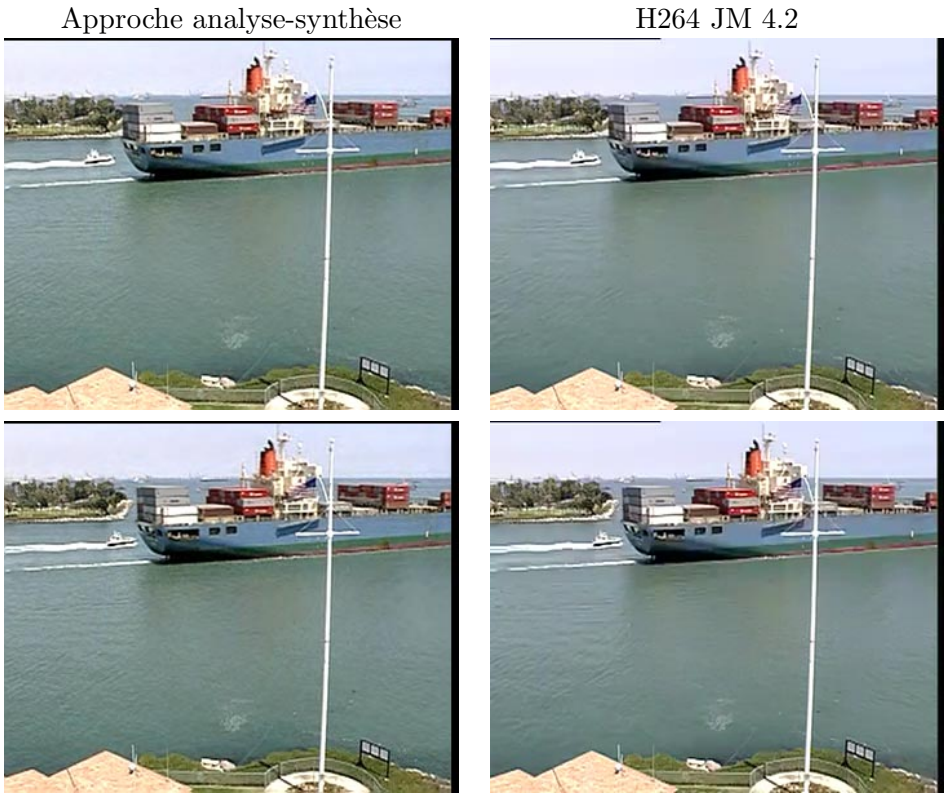


FIG. 4.22 – *Séquence Container 128kbit/s, image 140 et image 190*

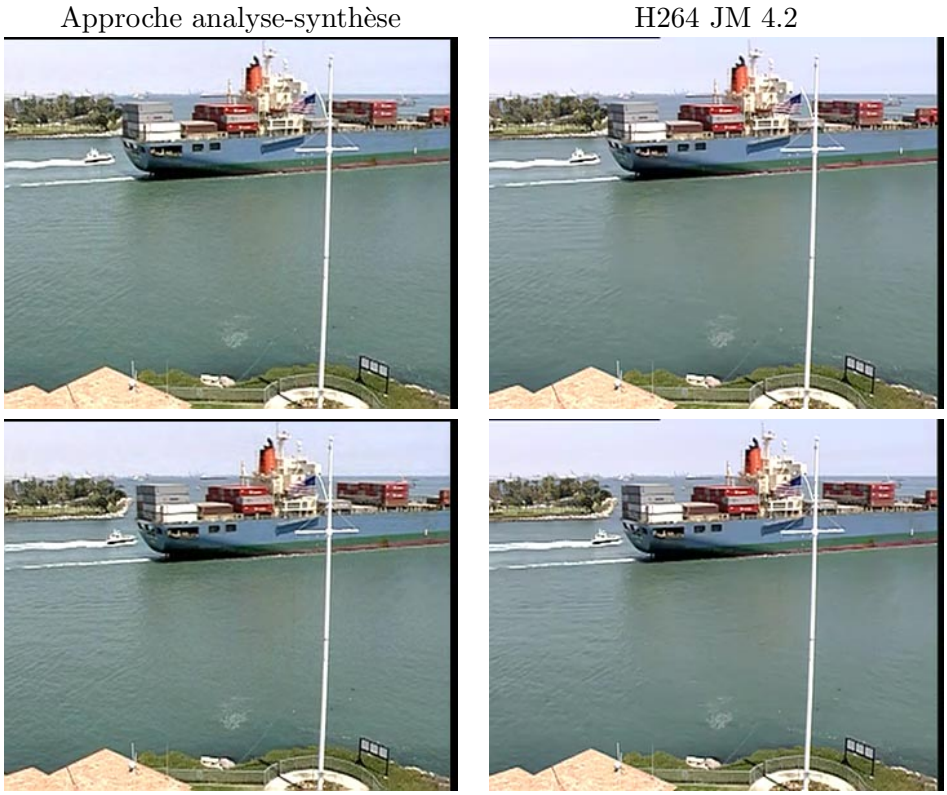


FIG. 4.23 – *Séquence Container 256kbit/s, image 140 et image 190*

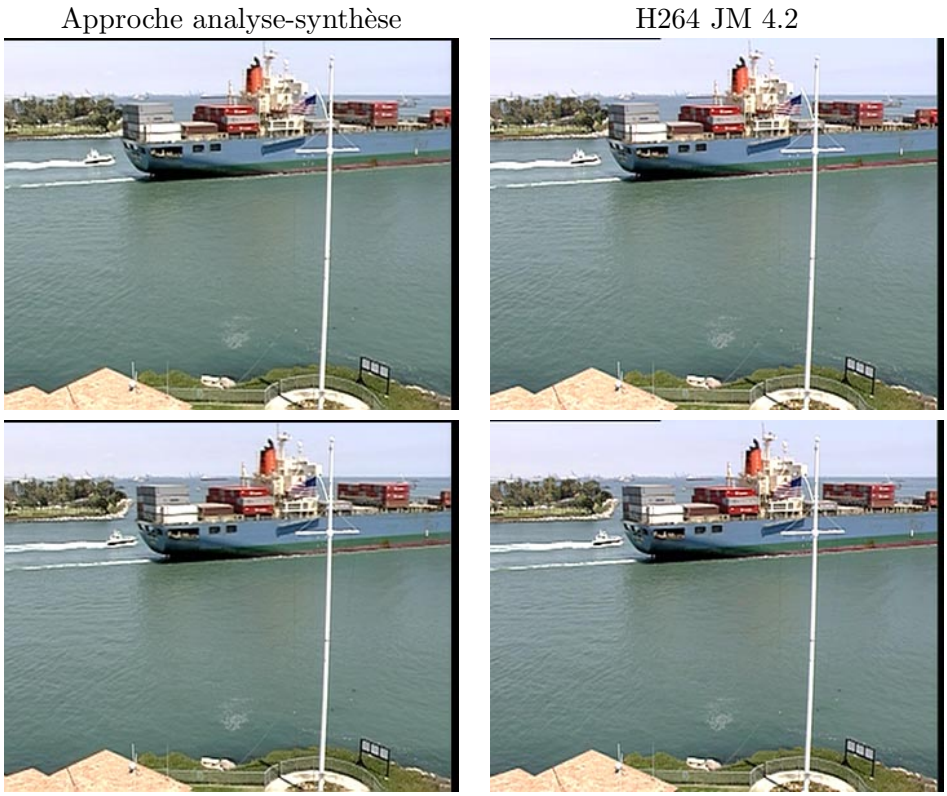


FIG. 4.24 – *Séquence Container 1024kbit/s, image 140 et image 190*

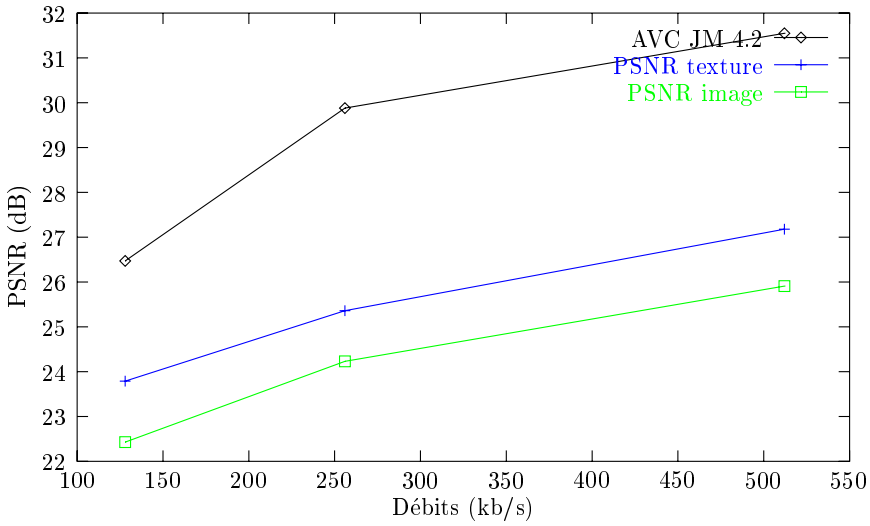


FIG. 4.25 – *PSNR sur la séquence Bus*

Image originale

Approche analyse-synthèse

H264 JM 4.2

FIG. 4.26 – *Séquence Bus 256kbit/s, image 32*

de la séquence vidéo. Puis, une couche de raffinement en qualité est codée par rapport à la couche de base. Enfin, une troisième couche code un raffinement spatial permettant de reconstruire une séquence vidéo à une plus grande résolution spatiale, temporelle et en qualité.
La couche de base est similaire à un schéma MPEG du type IPP. La taille du GOP est égale à la taille de la séquence.
La scalabilité temporelle est obtenue en insérant des images de type B du schéma MPEG dans les couches de raffinement. Ces images ne sont codées que par rapport à des références passées ou futures de la même couche.
La scalabilité spatiale n'est possible que sur deux niveaux de résolutions. Les couches de raffinement (SNR et spatiale) sont codées par prédiction intra et inter couches, ce qui peut entraîner un effet de dérive au décodage.

La proposition S13 est également basée sur un schéma en couches. Le codeur prend en entrée la séquence pleine résolution et construit des versions sous-échantillonées spatialement et temporellement de cette séquence. Ensuite, des sous-codeurs codent ces versions sous-échantillonnées. Les sous-codeurs sont basés AVC.

Dans la catégorie des schémas basés ondelettes, nous avons retenu les schémas proposés par Thomson/Irisa (proposition S18, [Vieron 04]), RPI (proposition S15, [Wu 04]) et par Microsoft China (proposition S05,[Xu 04]).
Les schémas des propositions S15 et S18 sont très similaires. Le codeur proposé par S15 est celui présenté dans [Chen 03b]). Les images sont traitées par groupe de 16. Une transformée temporelle lifting avec un filtre de Haar est appliquée sur les blocs connectés. Dans le cas de blocs contenant trop de pixels non connectés, un correspondant est cherché dans une référence passée, si aucun correspondant n'est trouvé, le bloc est codé en intra avec prédiction spatiale.
L'estimation de mouvement est effectuée de manière hiérarchique avec des blocs de taille variable. l'inversion du champ de mouvement est faite de manière directe en prenant l'arrondi au plus proche entier de l'opposé du vecteur mouvement considéré.
Le mouvement est codé de manière scalable dans les deux schémas. Pour S15, un codage en couches est utilisé; pour S18, la scalabilité du mouvement est basée sur la précision, celle-ci est adaptée à la résolution de décodage.
Les sous-bandes temporelles sont codées par un codeur EZBC pour la proposition S15 et par un codeur JPEG-2000 pour la proposition S18. La scalabilité SNR est obtenue en coupant le flux au débit voulu. Dans S15, la décomposition inverse des images est effectuée à la résolution temporelle et spatiale de décodage, les derniers niveaux des décompositions inverses ne sont pas effectués.

Enfin, la proposition S05 est basée sur un schéma de transformée ondelette t+2D. la transformée temporelle est effectuée par un schéma lifting appelé Barbell lifting. La phase de prédiction du schéma lifting est classique. La mise à jour des pixels basse fréquence est faite pour un pixel donné à l'aide de tous les pixels haute fréquence qui ont contribué à la prédiction du pixel courant. Cette technique est expliquée dans un précédent chapitre. Elle permet d'éviter l'inversion directe du mouvement par blocs qui

Séquences	Résolutions		Débits
	Spatiales	Temporelles	
Mobile And Calendar	QCIF	7.5Hz	64kbps
	QCIF	15Hz	128kbps
	CIF	15Hz	256kbps, 512kbps
	CIF	30Hz	1024kbps
Foreman, Bus	QCIF	7.5Hz	48kbps
	QCIF	15Hz	64kbps
	CIF	15Hz	128kbps, 256kbps
	CIF	30Hz	512kbps
City, Harbour	QCIF	15Hz	64kbps, 128kbps
	CIF	15Hz	192kbps
	CIF	30Hz	384kbps, 750kbps
	SD	30Hz	1500kbps
	SD	60Hz	3000kbps, 6000kbps

TAB. 4.5 – *Conditions de tests pour un codage scalable spatialement, temporellement et en qualité*

peut être nuisible à la compacité de l'énergie dans les sous-bandes.
Le mouvement est représenté par des blocs de taille variable, comme ce qui est fait dans le codeur H264/AVC. Il est codé en couches de manière scalable.
Les sous-bandes temporelles sont ensuite transformée par une décomposition ondelettes spatiale 2D SPACL, puis les sous-bandes spatio-temporelles sont codées à l'aide d'un codeur 3D EBCOT, une extension 3D du codeur arithmétique EBCOT.
Les différentes scalabilités sont obtenues en coupant dans le volume 3D et en omettant d'effectuer les derniers niveaux de décomposition inverse.

4.4.4 Conditions de tests

Le tableau 4.5 présente les séquences, les formats spatiaux et temporels et les débits utilisés pour les tests de comparaison de notre technique avec des codeurs scalables.
Les codeurs utilisés pour la comparaison sont présentés dans la sous-section précédente. Ils ont tous été proposés en réponse au Call For Proposal for Scalable Video Coding technology (CFP-SVC)[cfp 03].

Pour notre approche, le codeur utilisé est celui présenté dans le chapitre précédent. Les paramètres d'encodage sont les suivants: les images sont codées sur deux couches, par groupe de 8 images (GOF). Le premier GOF est Intra, les GOFs suivants sont tous codés en mode Inter (une image est partagée avec le GOF suivant). La couche basse est codée à 64 kbit/s avec un rafraîchissement Intra toutes les secondes. La couche haute est codée à débit infini.
Tous les codeurs n'ont codé qu'une seule fois les séquences, puis décodé successivement

aux débits et formats donnés. Les codeurs basés ondelettes ont codé la séquence originale aux formats spatial et temporel les plus grands. Tandis que les codeurs basés AVC ont codé une couche de base à partir de la séquence en QCIF, puis des couches de raffinement successives permettant d'obtenir les autres formats.
Nous avons utilisé la mesure du PSNR par rapport à la référence propre de chaque codeur pour comparer les différentes approches. Cette mesure n'est valable que dans la mesure où les séquences références des codeurs sont jugées visuellement équivalentes à la séquence originale.

4.4.5 Résultats sur la séquence Mobile And Calendar

Le tableau 4.6 donne les PSNR moyens pour la séquence Mobile And Calendar pour tous les codeurs présentés précédemment. Ces résultats sont illustrés par les courbes 4.27 et 4.28.
Si l'on étudie les résultats donnés pour la séquence Mobile And Calendar, notre technique fournit des résultats moyens au format CIF dans la gamme des débits moyens (128-256kbps), mais elle offre les meilleures performances pour le format QCIF et les bas débits.
Les images des figures 4.29 et 4.30 montrent la qualité visuelle des séquences décodées pour chaque codeur. Pour notre approche, les images reconstruites à 256kbps présentent des rebonds au niveau des contours (principalement sur le calendrier). Les images très texturées de cette séquence sont effectivement difficiles à coder en mode intra par un codeur basé ondelettes tel que JPEG2000 et la qualité des images intra influence la qualité globale de toute la séquence. Pour cette raison, nos performances à moyens débits sont moyennes.
Au niveau de la qualité visuelle perçue des séquences reconstruites, on observe des scintillements dus aux effets de blocs dans les bas et moyens débits. Ces scintillements ne sont visibles qu'en dynamique. Lorsqu'ils sont assez faibles, ils peuvent se confondre avec un bruit d'aliasing dus aux conditions d'acquisition de la séquence, mais ils deviennent très gênants lorsqu'ils sont trop nombreux. Ces scintillements persistent même à haut débit (1024kbps) pour certains codeurs (MC-EZBC S15, S18, S06 et Microsoft S05).
A haut débit, le codeur S13 donne la meilleure qualité visuelle. Malgré les rebonds d'ondelettes présents dans la séquence reconstruite par notre technique, celle-ci est de qualité comparable à celles des autres codeurs. En effet, notre technique ne présente pas les scintillements dus aux effets de blocs, la séquence résultante est alors plus confortable à regarder pour l'œil.
Lorsque l'on descend vers les bas débits, les scintillements augmentent et deviennent vraiment désagréables. Pour la technique S18, des sauts de texture sont également visibles et rendent la fluidité temporelle de la séquence reconstruite instable. Ces sauts de texture sont probablement dus à des rafraîchissements d'images en intra dans le flux.
Pour la technique S13, la qualité visuelle qui était bonne à haut débit, chute rapidement à mesure que le débit diminue. Les images de la séquence reconstruite sont nettes au milieu et de plus en plus floues en s'éloignant du milieu.

	QCIF		CIF		
Débits (kbps)	64	128	256	512	1024
Approche proposée	28.16	**30.17**	25.98	28.25	31.45
MC-EZBC-S15	25.28	28.21	**28.12**	**32.11**	**33.58**
Microsoft-S05	23.97	26.17	26.92	30.40	33.33
S06	**28.82**	28.53	25.72	27.24	27.31
Poznan-S13	26.11	26.14	22.13	24.19	33.19
S18	21.88	23.58	25.26	28.14	30.89

TAB. 4.6 – *PSNR moyens sur la séquence Mobile And Calendar*

Dans les très bas débits, les séquences reconstruites par les techniques ondelettes présentent non seulement des effets de blocs importants mais également des rebonds d'ondelettes importants rendant la texture des images floue.

4.4.6 Résultats sur les séquences Bus et Foreman

Les tableaux 4.7 et 4.8 donnent les PSNRs moyens obtenus par chaque codeur pour les séquences Bus et Foreman. Ces résultats sont illustrés sur les figures 4.31, 4.32, 4.33 et 4.34. Les courbes montrent des résultats par rapport aux autres codeurs scalables qui ne sont pas très bons. Notre technique offre tout de même de bonnes performances au format QCIF pour la séquence Bus.
Ces résultats s'expliquent par le fait que les séquences encodées (Foreman et Bus) sont des séquences difficiles à coder à l'aide d'un mouvement par maillages. Le mouvement contenu dans ces séquences est assez complexe, avec de fortes amplitudes et d'importantes zones de découvrement et de recouvrement. Ces zones sont difficilement prédictibles et les étirements et tassements de mailles générées par l'estimation du mouvement par maillages créent des motifs coûteux à coder. Les images présentées sur la figure 4.35 montrent de tels étirements et tassements et les images reconstruites à bas débits.
La comparaison de la qualité visuelle des séquences reconstruites est assez semblable à celle effectuée pour la séquence Mobile And Calendar. On retrouve les mêmes scintillements dus aux effets de blocs pour les techniques basées blocs.
Pour les hauts débits, la technique S13 fournit de très bons résultats, mais ces résultats chutent rapidement avec le débit.
Les techniques basées ondelettes présentent beaucoup de rebonds d'ondelettes. La technique S18 présente également des artéfacts dus au découvrement de texture sur les bords des images.
Dans les bas débits, les techniques basées AVC sont nettement meilleures que les techniques ondelettes. Ces dernières subissent des rebonds d'ondelettes trop importants qui rendent la texture floue. Les techniques basées AVC sont cependant trop lisses. Ceci est du au filtrage effectué pour éliminer les effets de blocs. Le lissage permet d'offrir une qualité visuelle acceptable mais sans aucune finesse de détails. Sur certaines séquences, le lissage fait perdre du réalisme à la séquence vidéo.

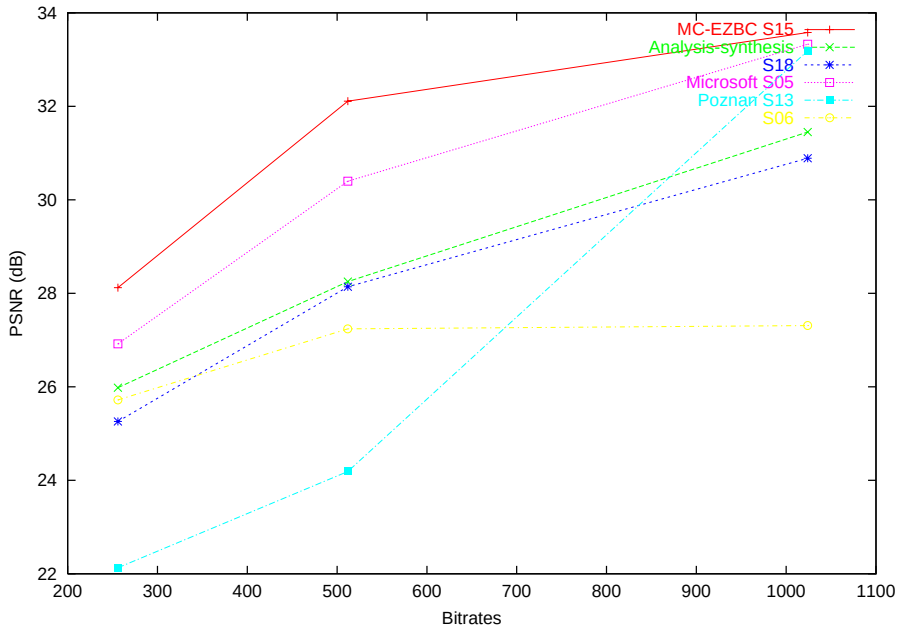


FIG. 4.27 – *PSNR sur la séquence Mobile And Calendar, CIF*

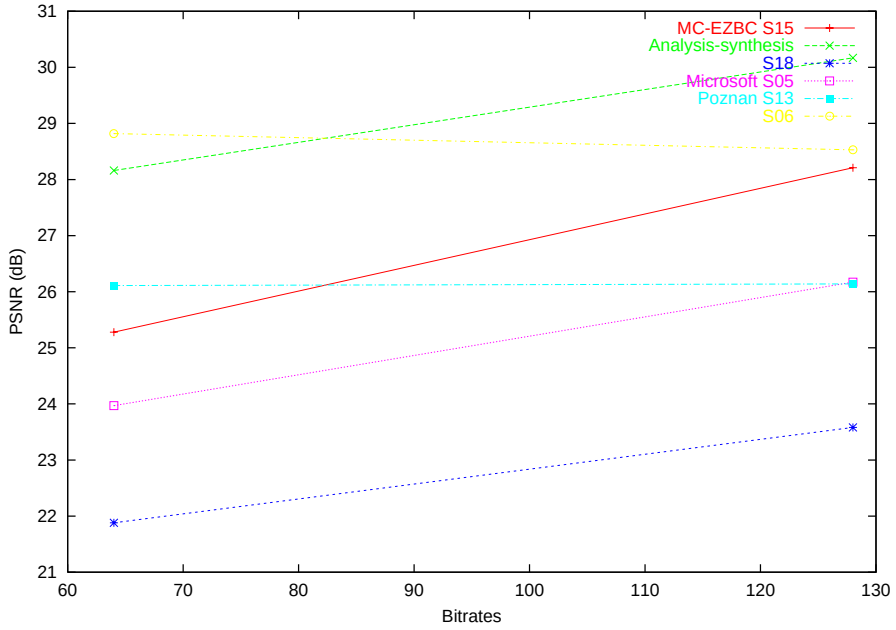


FIG. 4.28 – *PSNR sur la séquence Mobile And Calendar, QCIF*

FIG. 4.29 – *Séquence Mobile And Calendar, image 25: Images reconstruites par chaque codeur scalable, séquence décodée à 256kbps au format CIF, 15Hz.*

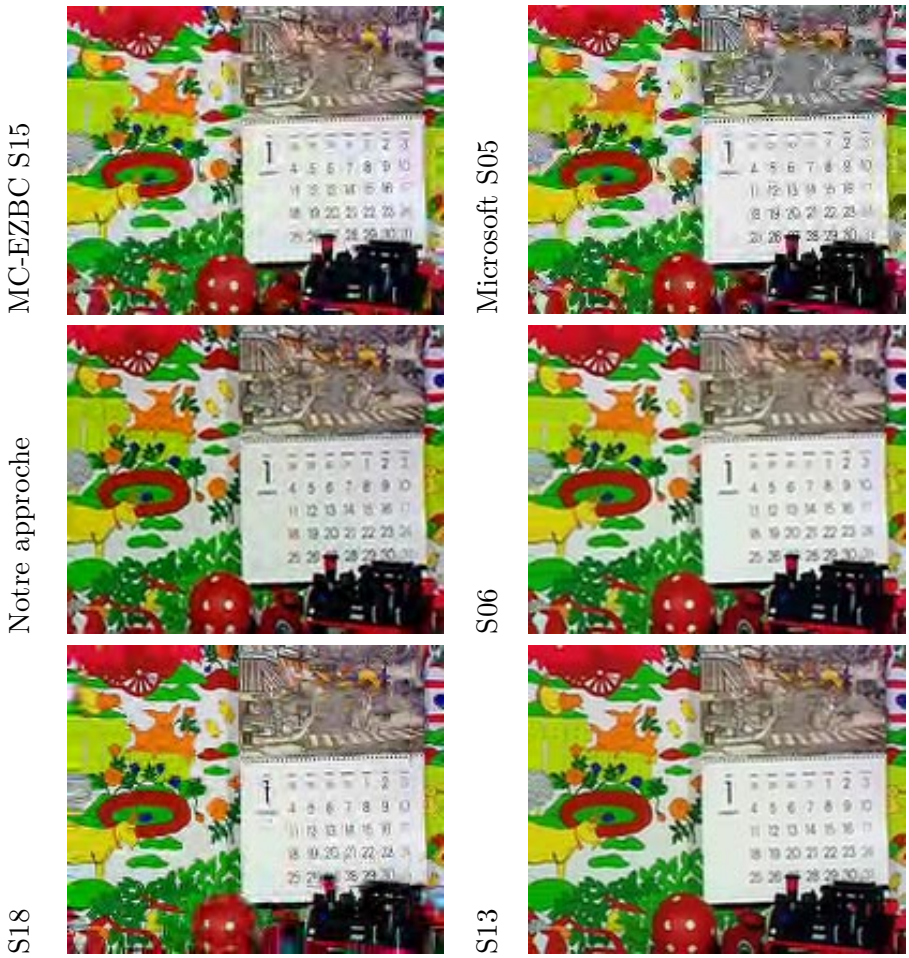


FIG. 4.30 – *Séquence Mobile And Calendar, image 12: Images reconstruites par chaque codeur scalable, séquence décodée à 64kbps au format QCIF, 7.5Hz.*

	QCIF		CIF		
Débits (kbps)	48	64	128	256	512
Approche proposée	25.79	**26.60**	23.79	25.36	27.18
MC-EZBC-S15	23.37	22.55	23.90	**28.19**	30.09
Microsoft-S05	24.64	24.66	**25.43**	28.02	30.32
S06	25.09	24.95	22.60	25.55	25.74
Poznan-S13	**26.03**	25.92	23.14	24.84	**31.08**
S18	21.78	21.97	23.65	26.11	27.92

TAB. 4.7 – *PSNR moyens sur la séquence Bus*

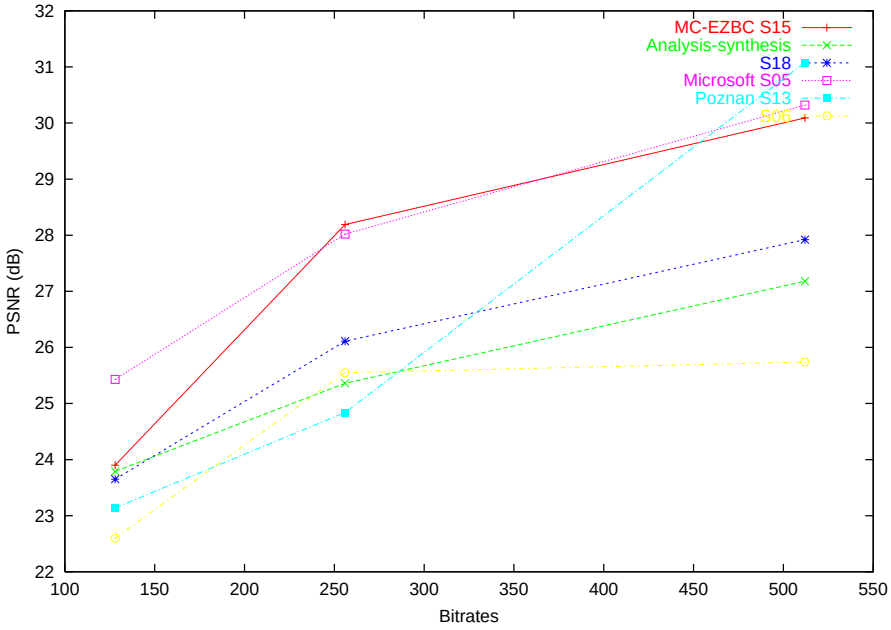


FIG. 4.31 – *PSNR sur la séquence Bus, CIF*

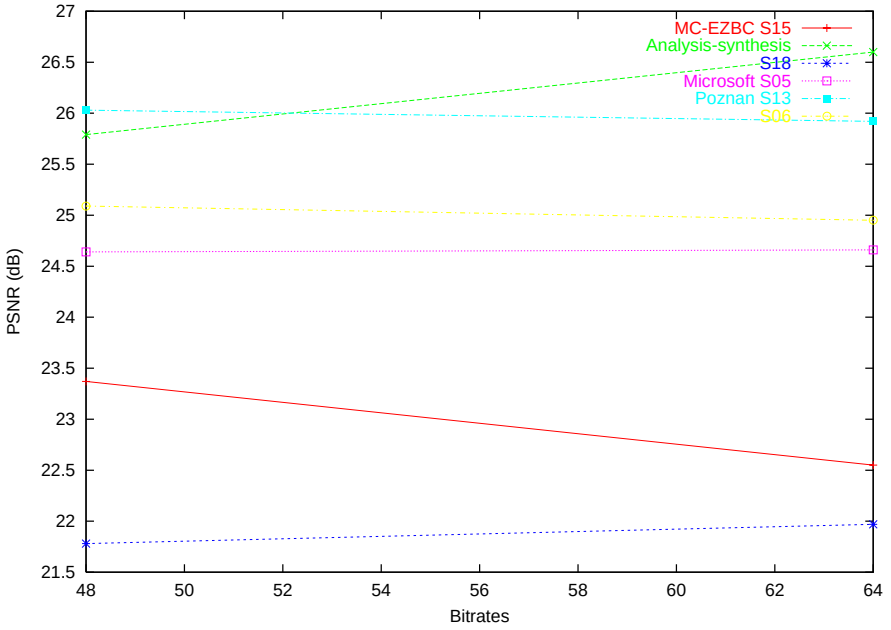


FIG. 4.32 – *PSNR sur la séquence Bus, QCIF*

	QCIF		CIF		
Débits (kbps)	48	64	128	256	512
Approche proposée	27.97	29.24	28.25	31.19	33.53
MC-EZBC-S15	29.89	29.46	30.58	**34.28**	36.16
Microsoft-S05	30.27	30.14	**31.44**	33.61	34.99
S06	24.68	24.65	27.37	28.94	29.00
Poznan-S13	**32.40**	**32.25**	28.80	29.93	**37.45**
S18	27.33	27.46	29.92	32.77	34.72

TAB. 4.8 – *PSNR moyens sur la séquence Foreman*

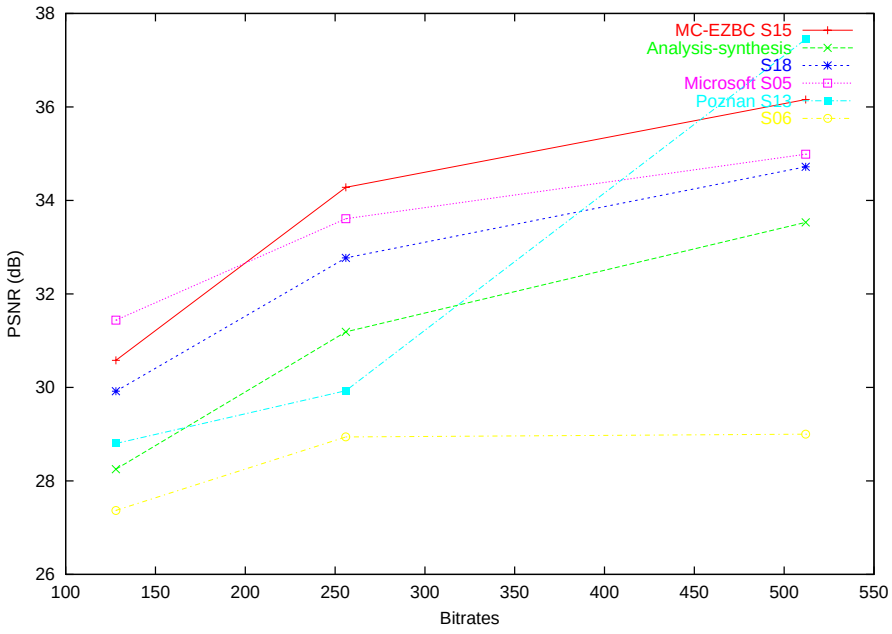


FIG. 4.33 – *PSNR sur la séquence Foreman, CIF*

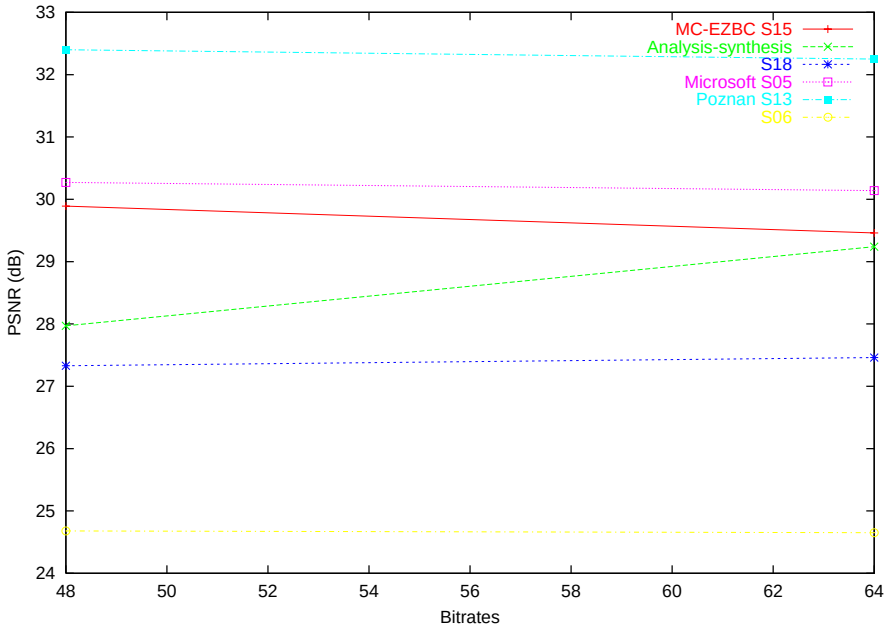


FIG. 4.34 – *PSNR sur la séquence Foreman, QCIF*

4.4.7 Résultats sur les séquences City et Harbour

Les tableaux 4.9 et 4.10 donnent les PSNRs moyens obtenus par chaque codeur pour les séquences City et Harbour. Les courbes débit-distorsion sont données pour chaque séquence sur les figures 4.36, 4.37, 4.38, 4.39, 4.40 et 4.41.
Pour la séquence City, notre technique offre de très bons résultats quels que soient les débits et formats de restitution. Ce qui n'est pas le cas pour les autres codeurs. Si l'on regarde les codeurs scalables performants au format SD les deux premiers codeurs sont la technique proposée par Microsoft (S05) et notre codeur par analyse-synthèse. Pour le format CIF, notre technique est toujours en tête, mais cette fois-ci suivie du codeur MC-EZBC (S15) et le même constat est fait pour le format QCIF. On voit d'après ces courbes, qu'il est difficile pour un codeur scalable d'être performant quels que soient les débits et les formats de restitution.
Pour la séquence Harbour, notre technique est performante au format SD et à haut débit. Mais les performances chutent quand le débit diminue. La technique reste compétitive au format CIF et QCIF. Les chutes de performances quand le débit diminue sont dues aux occultations présentes dans la séquence avec les croisements de mât des bateaux dans lesquels le maillage s'accroche. Ceci crée des étirements de mailles générant des défauts coûteux à corriger.
Au niveau de la qualité visuelle perçue, les mêmes constats que pour les séquences précédentes peuvent être faits. Pour la séquence City, de l'aliasing apparaît lors du décodage à une résolution inférieure au format SD pour les techniques S18, MC-EZBC

FIG. 4.35 – *Séquence Foreman: étirement et tassement de mailles et images recosntruites à 128kbps correspondantes.*

	QCIF		CIF			SD		
Débits (kbps)	64	128	192	384	750	1500	3000	6000
Approche proposée	**32.07**	**35.33**	**31.19**	**32.60**	34.51	32.63	**35.97**	**38.90**
MC-EZBC-S15	30.95	35.21	30.44	32.39	**34.87**	32.40	32.57	34.63
Microsoft-S05	29.26	32.07	28.44	30.25	32.35	**33.70**	35.89	37.63
Poznan-S13		32.84	26.91	28.28	33.28	27.57	29.54	37.39
S18	26.21	28.54	26.75	29.07	31.80	33.08	34.81	37.14

TAB. 4.9 – *PSNR moyens sur la séquence City*

	QCIF		CIF			SD		
Débits (kbps)	64	128	192	384	750	1500	3000	6000
Approche proposée	**27.45**	27.61	26.62	27.95	30.16	30.34	33.92	**37.15**
MC-EZBC-S15	26.43	29.24	**27.14**	**28.65**	**31.11**	31.23	31.93	34.58
Microsoft-S05	25.97	28.98	26.14	28.36	30.86	31.84	**34.16**	36.45
Poznan-S13		**29.85**	23.82	24.99	30.15	27.99	29.26	35.40
S18	24.28	25.95	25.70	27.62	29.67	**32.37**	33.21	35.60

TAB. 4.10 – *PSNR moyens sur la séquence Harbour*

et Microsoft S05. Ceci est dû à la manière dont les codeurs exploitent la scalabilité spatiale. Ces codeurs coupent les hautes fréquences spatiales dans le flux d'informations et ne les utilisent pas lors de la reconstruction. Notre technique ne présente pas cet aliasing car nous gardons les hautes fréquences spatiales même pour un décodage à une résolution inférieure au format SD, nous effectuons ensuite un sous-échantillonnage des informations dans le domaine de la texture et celui du mouvement et nous reprojettons la texture sur le mouvement au format de restitution demandé.
Les figures 4.42, 4.43 et 4.44 montrent des images des séquences reconstruites aux formats QCIF et CIF à 64, 384 et 750 kbps pour notre approche, le codeur MC-EZBC S15 et le codeur Microsoft S05. On remarque l'aliasing spatial des deux codeurs S15 et S05 sur le building en arrière-plan.
Pour la séquence Harbour, nos résultats en PSNR sont en-dessous de ceux des techniques MC-EZBC S15 et Microsoft S05. Cependant, les qualités visuelles des séquences reconstruites sont assez proches car nos séquences ne présentent pas les scintillements dus aux effets de blocs des autres techniques, ni le flou du aux rebonds d'ondelettes au frontière des blocs. Nos séquences reconstruites présentent des artéfacts liés au champ de mouvement par maillage (typiquement des étirements de mailles), mais elles offrent une grande stabilité et un confort d'observation que n'offrent pas les autres techniques.
La figure 4.45 montrent la qualité visuelle des images reconstruites pour notre approche et les techniques S18, MC-EZBC S15 et Microsoft S05 à 64kbps au format QCIF. On observe une meilleure définition des contours du bateau pour notre approche.

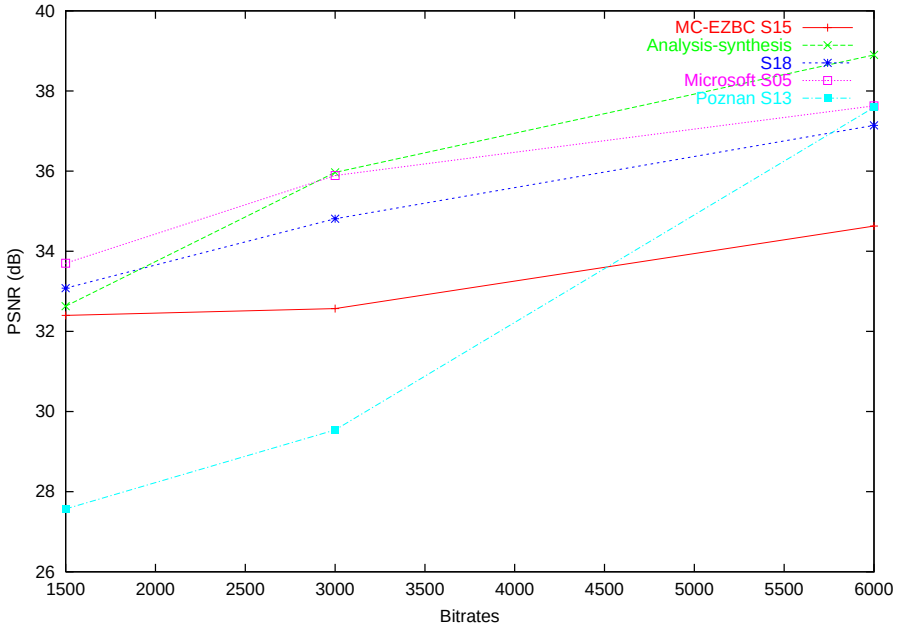


FIG. 4.36 – *PSNR sur la séquence City, SD*

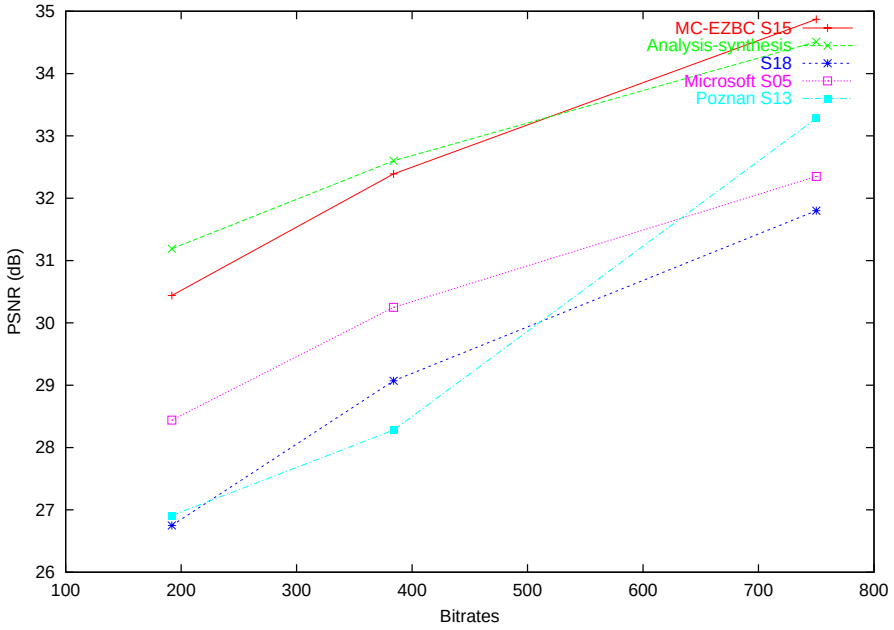


FIG. 4.37 – *PSNR sur la séquence City, CIF*

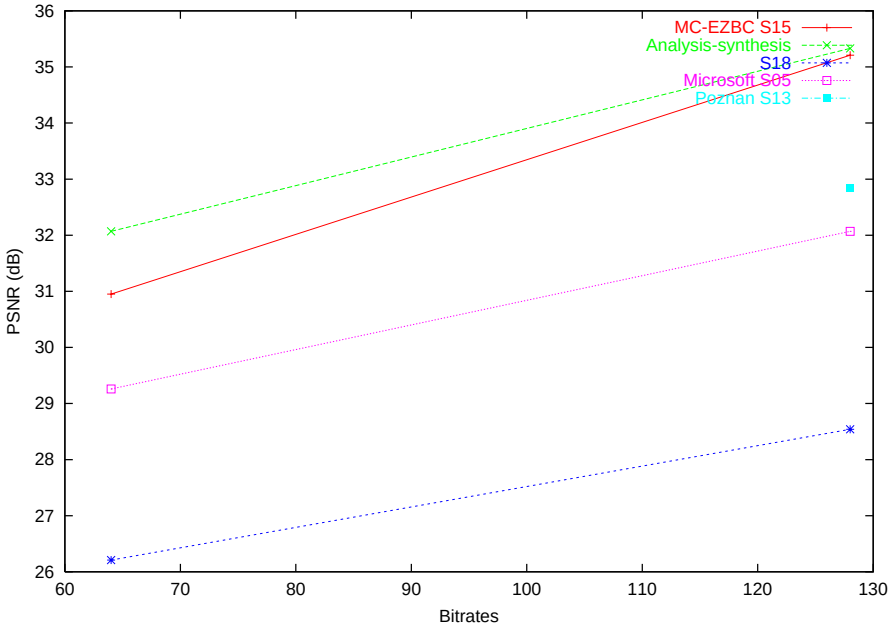


FIG. 4.38 – *PSNR sur la séquence City, QCIF*

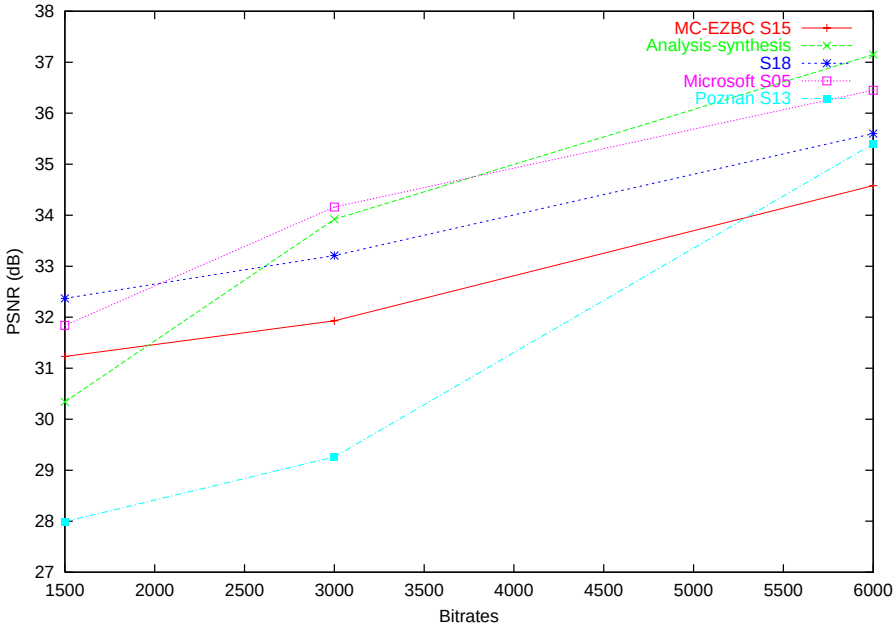


FIG. 4.39 – *PSNR sur la séquence Harbour, SD*

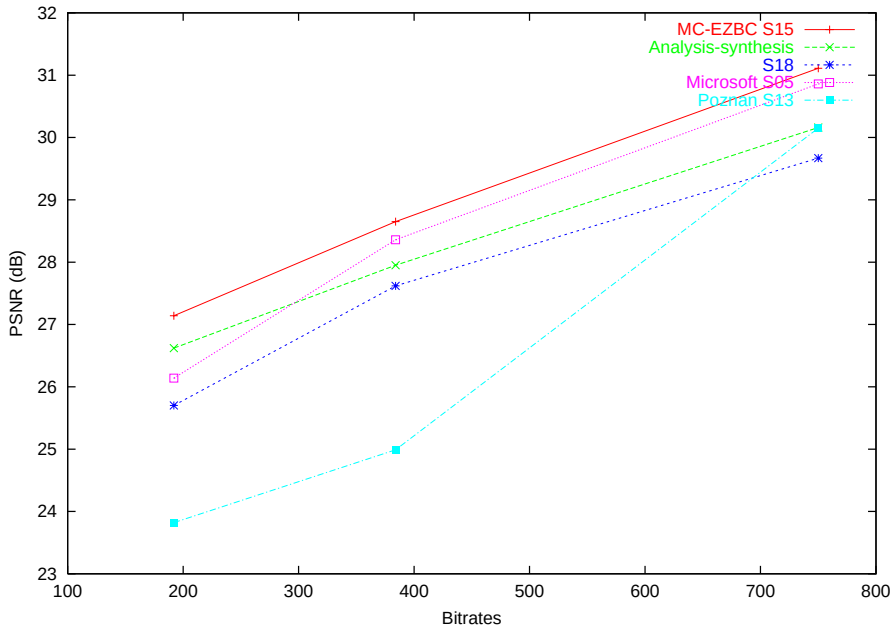


FIG. 4.40 – *PSNR sur la séquence Harbour, CIF*

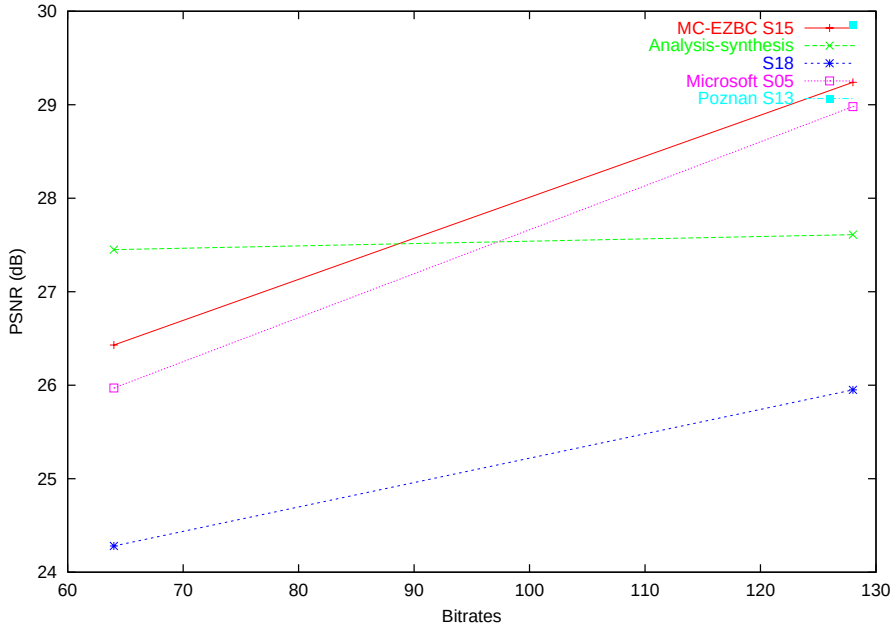


FIG. 4.41 – *PSNR sur la séquence Harbour, QCIF*

FIG. 4.42 – *Séquence City: images reconstruites à 64kbps, QCIF, 15Hz.*

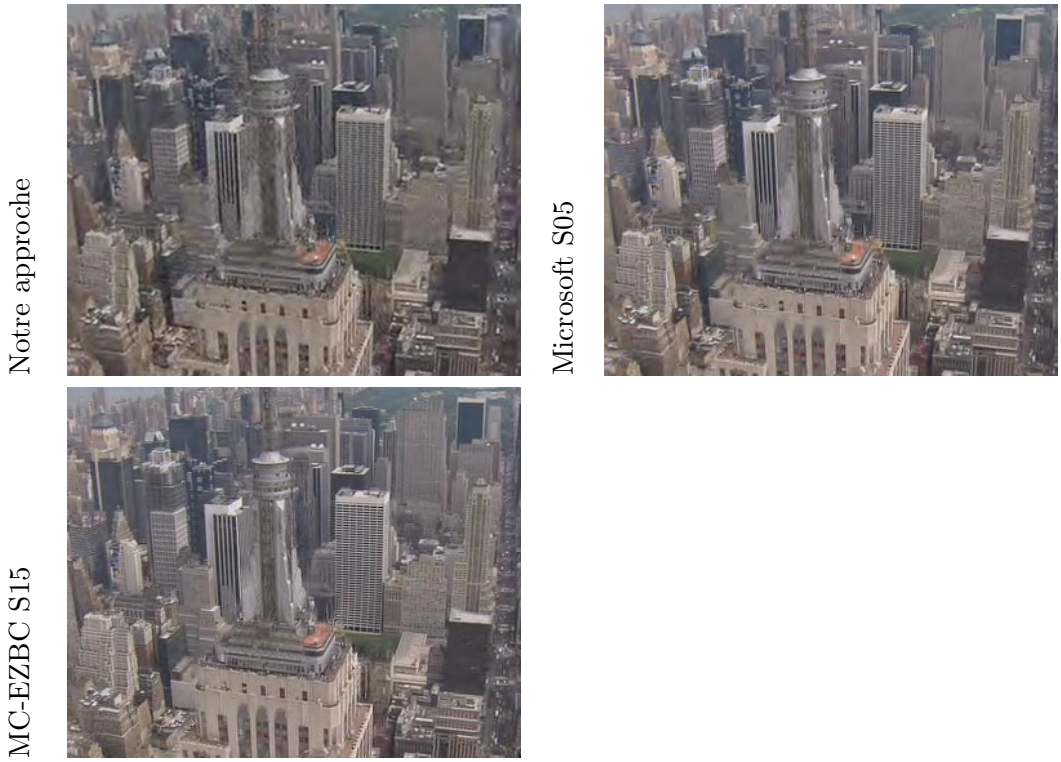


FIG. 4.43 – *Séquence City: images reconstruites à 384kbps, CIF, 30Hz.*

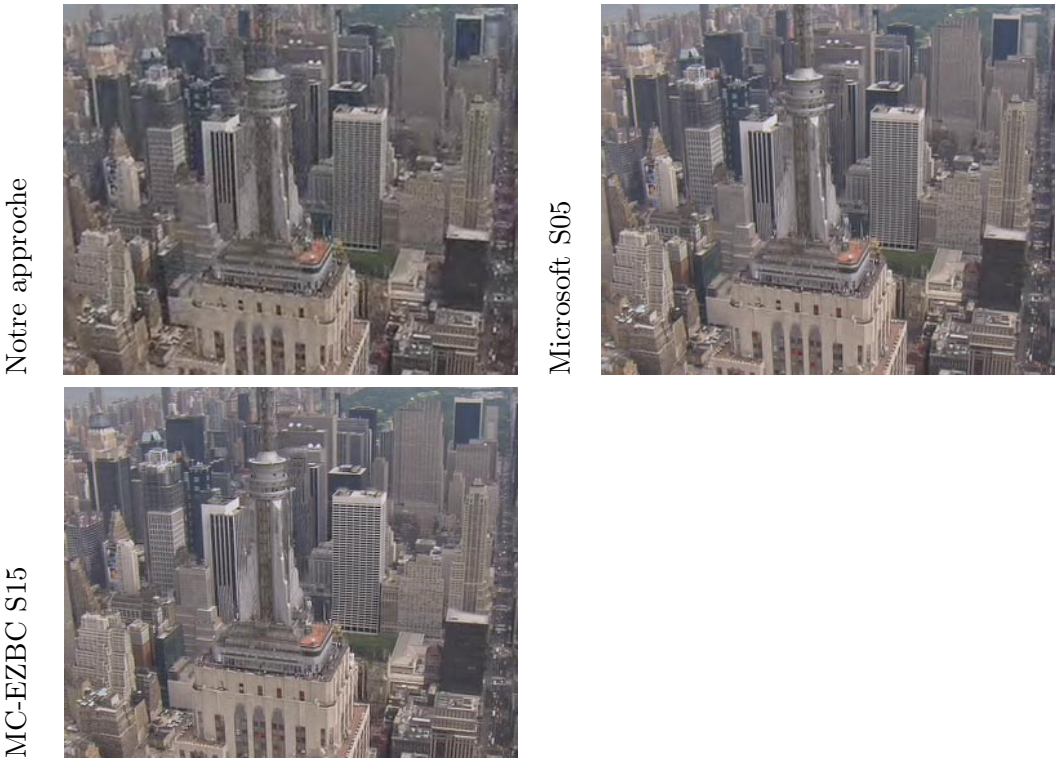


FIG. 4.44 – *Séquence City: images reconstruites à 750kbps, CIF, 30Hz.*

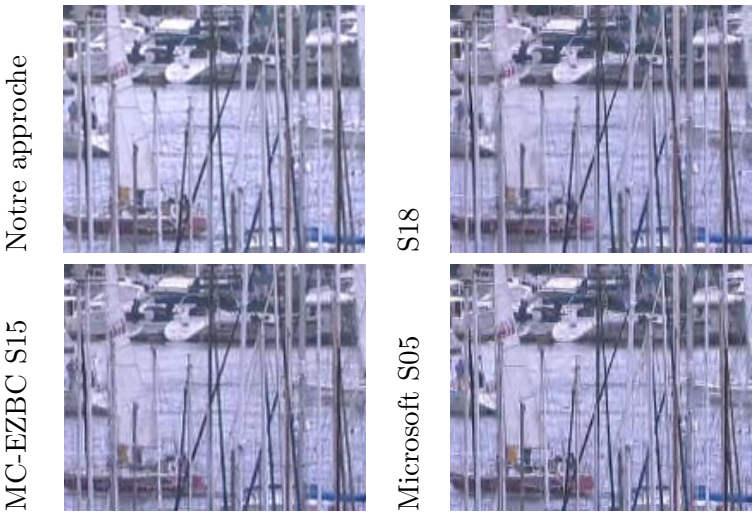


FIG. 4.45 – *Séquence Harbour: images reconstruites à 128kbps, QCIF, 15Hz.*

4.5 Mode Objet

Cette section présente les performances de notre codeur vidéo dans le cas de codage en mode objet. Dans le cas du codage par objet, trois informations par objet sont à coder: le mouvement, la texture et la forme.
Les tableaux 4.11 et 4.12 présentent une comparaison des performances de notre codeur en mode objet avec un codeur non scalable et en mode rectangulaire H264 pour les séquences Foreman et Erik respectivement. Les séquences ont été encodées à 15Hz. Le codeur H264 utilisé est la version du VM8.4 avec un shéma IBBP, 5 références possibles, codage arithmétique CABAC et optimisation débit/distorsion. Les tableaux présentent la répartition du débit pour les différents champs d'information à coder ainsi que les PSNRs obtenus par chaque codeur. Pour notre technique, le PSNR a été calculé dans le domaine texture comme expliqué dans la sous-section 4.4.1 à partir des pixels définis par le masque de segmentation. Le PSNR donné pour le codeur en mode objet et en mode non objet, ainsi que pour H264 correspond au PSNR de l'objet en avant-plan.
La figure 4.46 présente des images reconstruites par H264 et par notre codeur en mode objet. On remarque que bien que le PSNR soit légèrement inférieur par rapport à H264 pour notre approche pour la séquence Foreman, la qualité visuelle des images reconstruites est meilleure. Pour la séquence Erik, le PSNR et la qualité visuelle sont meilleurs qu'avec H264. Les images reconstruites par H264 présentent moins de détails, dus au filtrage effectué pour éliminer les effets de blocs, que les images reconstruites par notre approche qui montrent une finesse de détails.
La comparaison du codeur par analyse-synthèse en mode objet et en mode non objet sur la séquence Foreman montre que le codeur en mode objet atteint de meilleures performances qu'en version non objet. Le mouvement complexe présent dans cette séquence, dus au découvrement et recouvrement de texture est mieux représenté à l'aide d'un codage en mode objet. En effet, le mouvement a pu être estimé de manière indépendante pour le personnage en avant-plan et pour le fond, ce qui a permis de limiter les étirements et tassements de mailles qui étaient présents dans la version non objet.
Le codeur en mode objet donne de bons résultats lorsqu'une segmentation de la séquence vidéo est disponible, cependant comme tous les codeurs en mode objet, son efficacité dépend fortement de la qualité de la segmentation. De plus, le nombre d'objets dans la scène et leur taille influencent les performances du codeur. Dans le cas de petits objets, le coût de codage de la forme devient souvent trop important par rapport au gain que l'on a sur la texture et le mouvement de ces objets et un codage en mode non objet est souvent préférable.

Conclusion

Ce chapitre a montré les résultats en terme de codage donnés par le codeur proposé dans le chapitre précédent. Nous avons d'une part validé les choix faits à chaque étape du processus de codage et d'autre part positionné notre approche par rapport à d'autres codeurs existants scalables et non scalables.
Nous avons montré les bonnes performances de notre codeur en terme de compression

	analyse-synthèse objet	H264	analyse-synthèse non objet
Motion (kbps) Texture (kbps) Shape (kbps)	17.56 69.12 6		
Background (kbps)	16		
Total bitrates (kbps)	108	98	128
PSNR(dB)	32.06	32.59	31.10

TAB. 4.11 – *Comparaison du schéma analyse-synthèse en mode objet avec H264 et avec le codeur en mode non objet pour la séquence Foreman*

	Analyse-synthèse			H264	
Motion (kbps)	11.9	12.23	12.25		
Texture (kbps)	28.6	37.8	47.4		
Shape (kbps)	3	3	3		
Background (kbps)	11.5	11.5	11.5		
Total bitrates (kbps)	55	64	74	52	70
PSNR(dB)	27.9	28.4	29	26.9	28.2

TAB. 4.12 – *Comparaison en mode objet avec H264 pour la séquence Erik*

FIG. 4.46 – *images reconstruites pour les séquences Foreman et Erik en mode objet et non objet et par H264*

pure ainsi qu'en terme de compression et de scalabilité. Nous avons également montré les limitations de notre schéma, notamment au niveau du manque de généricité des séquences traitées. En effet, lorsque le mouvement dans une séquence vidéo devient trop complexe, l'estimation du mouvement par maillage échoue et ne permet pas d'assurer une bonne prédiction des images. Il en résulte un important coût de codage. Nous avons vu qu'une première solution à ce problème pouvait être le codage en mode objet. L'estimation du mouvement et le codage de chaque objet indépendant permet d'éviter les étirements et les tassements de mailles au frontière des objets qui ont un mouvement contradictoire. Le codage en mode objet est cependant très dépendant de la qualité de la segmentation.
L'objet des prochains chapitres est une étude et la proposition d'une solution de la gestion des problèmes d'occlusion dans une séquence vidéo. La solution proposée permettra de pouvoir régler le problème de généricité de notre codeur.

Chapitre 5

Gestion des zones d'occlusions: un état de l'art

5.1 Introduction

Dans la partie précédente, nous avons présenté un schéma de codage vidéo scalable utilisant une approche analyse-synthèse. L'utilisation des maillages pour la représentation du mouvement et le suivi de textures offre de nombreux avantages à notre schéma, notamment une décorrélation efficace du mouvement et de la texture.
Cependant, l'inconvénient majeur de notre schéma est son manque de généricité. Dans les séquences vidéo présentant des mouvements très complexes et de fortes amplitudes, le suivi par maillages décroche. Ceci implique alors une mauvaise prédiction des textures et la phase de compensation aller-retour de l'analyse-synthèse introduit des défauts dans la séquence reconstruite inhérents au schéma et qui ne peuvent être corrigés, même par une augmentation du coût de codage.
Le décrochage du maillage est du au phénomène d'occlusion dans la séquence vidéo. Dans ce chapitre, nous allons définir le phénomène d'occlusion et montrer ses conséquences sur le suivi de la texture et sur la représentation du mouvement. Puis, nous ferons un état de l'art des techniques proposées dans la littérature permettant de gérer les occlusions.

5.2 Le problème des occlusions

5.2.1 Définition d'une occlusion

Le problème des occlusions est un problème non résolu dans le traitement de séquences vidéo. Une séquence vidéo donne une représentation en deux dimensions d'une scène dynamique en trois dimensions. La scène est dynamique non seulement parce que les éléments qui la composent ne sont pas fixes et peuvent bouger et/ou interagir entre eux, mais également parce que la caméra qui acquiert cette séquence n'est pas fixe et peut bouger par rapport au référentiel de la scène.

Le phénomène d'occlusion dans une séquence vidéo est du à l'apparition, à la disparition ou aux croisements d'objets dans la vidéo. C'est le résultat du mouvement de la caméra et des objets dans la scène 3D.
L'apparition et/ou la disparition de texture sont dues au mouvement des objets dans la scène mais peuvent être également dues aux déformations des objets. Par exemple un ballon immobile qui se dégonfle fait apparaître de la texture du fond qui n'était pas visible lorsque le ballon était gonflé.
Les occlusions rendent difficiles l'estimation du mouvement apparent dans la séquence vidéo. En effet, les techniques d'estimation du mouvement entre deux images t_1 et t_2 d'une séquence vidéo sont basées sur les ressemblances existant entre ces deux images. Si des zones apparaissent ou disparaissent entre les deux images, les ressemblances diminuent et l'estimation échoue dans ces zones.
Dans le cadre de notre étude, le codage vidéo, la représentation et l'estimation du mouvement apparent ont un impact direct sur l'efficacité de codage. Que l'on utilise un codeur prédictif type MPEG ou un codeur utilisant une transformation temporelle décorrélante, les trajectoires de mouvement dans la vidéo doivent être connues de manière exacte afin de réduire l'information d'erreur de prédiction ou d'augmenter la compacité du signal par la transformée. C'est l'efficacité de ces deux transformations qui influe sur la compression du signal.

5.2.2 Représentation du mouvement apparent

Le champ de mouvement peut être représenté de manière discontinu à l'aide d'une représentation par blocs (type MPEG 1,2,4, H26x), ou bien de manière continu par une représentation par maillages [Wang 94, Dudon 95, Dudon 97, Toklu 96].
La plupart des techniques d'estimation utilisent l'hypothèse que la variation d'illumination d'un objet dans une scène est quasiment nulle. Ceci permet d'estimer le mouvement des objets dans une séquence vidéo en se basant sur leur valeur de luminance et en recherchant des correspondances entre les images.
Les méthodes par blocs effectuent un appariement de blocs entre une image t_1 et une image t_2. Pour chaque bloc de l'image t_2, le processus recherche un correspondant dans l'image t_1. La complexité est contrôlée en utilisant une fenêtre de recherche et une taille de blocs plus ou moins grandes.
La figure 5.1 illustre une estimation du mouvement par appariement de blocs. La représentation du mouvement par blocs affecte le même vecteur déplacement à tous les pixels d'un bloc. Ceci se traduit par un champ de mouvement discontinu; en effet, le mouvement est continu à l'intérieur d'un bloc, mais aux frontières des blocs, il présente de brusques sauts de valeurs.
La représentation par maillage modélise le champ de mouvement à l'aide de nœuds placés sur une grille d'échantillonnage de manière régulière ou non. Les nœuds et les connexions entre des nœuds adjacents permettent de définir des mailles. L'estimation du mouvement par maillages déforme les mailles en évitant les recouvrements et les découvrements de mailles. Contrairement aux méthodes basées blocs, l'estimation du

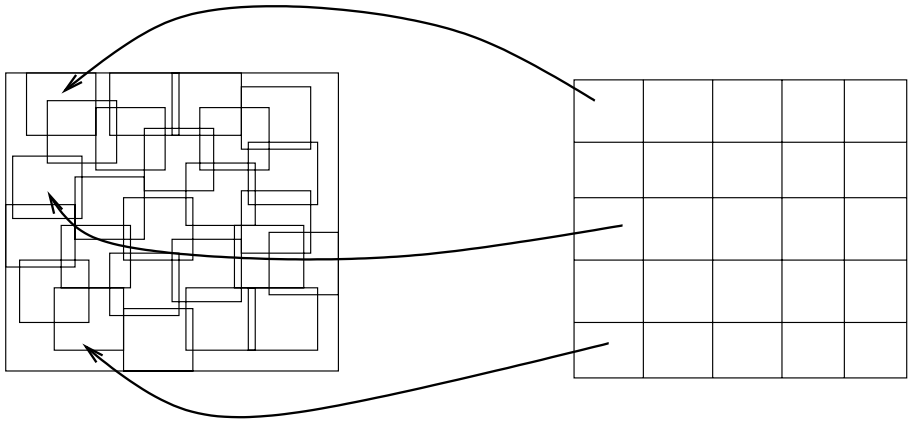


FIG. 5.1 – *Estimation du mouvement par blocs*

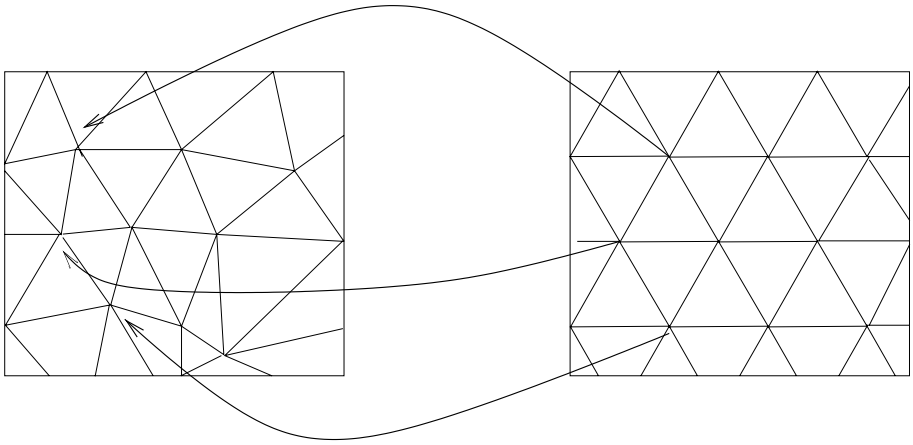


FIG. 5.2 – *Estimation du mouvement par maillage*

mouvement par maillage permet de recouvrir complètement le domaine de l'image.
L'estimation du mouvement calcule le déplacement des nœuds du maillage et le champ de mouvement dense est obtenu en chaque point d'une maille par une interpolation à partir des nœuds de la maille. Le champ de mouvement obtenu est continu.
La figure 5.2 illustre la représentation du mouvement par maillages.

5.2.3 Conséquences des occlusions sur la représentation du mouvement

Les occlusions se traduisent de manière différente selon la représentation du mouvement utilisée.
Dans les méthodes par blocs, les zones de découvrements créent des pixels non connectés. Les pixels non connectés sont ceux dont un correspondant n'a pas pu être trouvé dans

l'image référence et auquel on n'a pas pu affecter de vecteur mouvement. Les zones de recouvrement créent des pixels multiplement connectés, ces pixels se sont vu assigner plusieurs correspondants de l'image référence.
La gestion des pixels non connectés ou multiplement connectés est alors différente de celle des pixels ayant un correspondant unique [Ohm 94, Choi 99]. Elle peut parfois entraîner un surcoût de codage, notamment dans le cas de l'utilisation d'une transformée temporelle. La solution pour les pixels non connectés est de chercher un correspondant dans une autre image, soit en affectant un 'faux' vecteur mouvement au pixel (vecteur mouvement d'un bloc voisin) [Golwelkar 03], soit en effectuant une estimation du mouvement bi-directionnelle [Secker 01]. Dans le cas où un bloc possèderait plus de 50% de ses pixels non connectés, ce bloc peut être codé en mode intra [Chen 03a], car il traduirait une apparition de textures.
Un autre problème posé par la représentation par blocs est la rigidité du mouvement à l'intérieur d'un bloc. La rigidité des blocs ne permet pas de représenter la déformation éventuelle d'un objet. Dans l'estimation du mouvement, à un bloc de taille n×m est associé un bloc de même taille.
L'utilisation de blocs de taille variable [Schäfer 03] permet d'adapter la représentation du mouvement au contenu de la vidéo mais ne permet pas la prise en compte des déformations des objets. L'adaptation au contenu permet dans une certaine mesure de réduire le nombre de pixels sans correspondant. La représentation du mouvement par blocs de taille variable permet d'approcher la forme des objets contenus dans la vidéo, mais ceci conduit à une représentation par segments grossière du contour des objets et induit des perturbations au niveau de ces contours, notamment dans la séquence reconstruite où les contours d'un objet ne sont pas stables.

Dans la représentation du mouvement par maillages, les zones de recouvrement et de découvrement créent des étirements et des contractions de mailles produisant des mailles dégénérées, voire retournées. L'étirement de mailles a pour conséquence un étalement des textures dans la zone découverte, la contraction de mailles implique des pertes de résolution de la texture et introduit un phénomène d'aliasing, voir figure 5.3. Le cas extrême des retournements de mailles crée des zones non prédictibles dans la texture.
La déformation extrême des mailles provoque des pertes de résolution irréversibles lors d'une compensation en mouvement. Dans le cas d'une zone étirée, la texture étalée tend à être floue, tandis que dans une zone de contraction, un phénomène d'aliasing apparaît.
La déformation des mailles lors de l'estimation du mouvement permet dans une certaine mesure de représenter la déformation éventuelle des objets de la vidéo. Cependant, le maillage étant placé sur toute l'image, des perturbations peuvent apparaître dans le cas de mouvement ou déformation de plusieurs objets en conflit.

Dans la suite de ce chapitre, nous nous intéressons aux techniques proposées dans la littérature pour la gestion des occlusions dans les méthodes de représentation du mouvement par maillages.

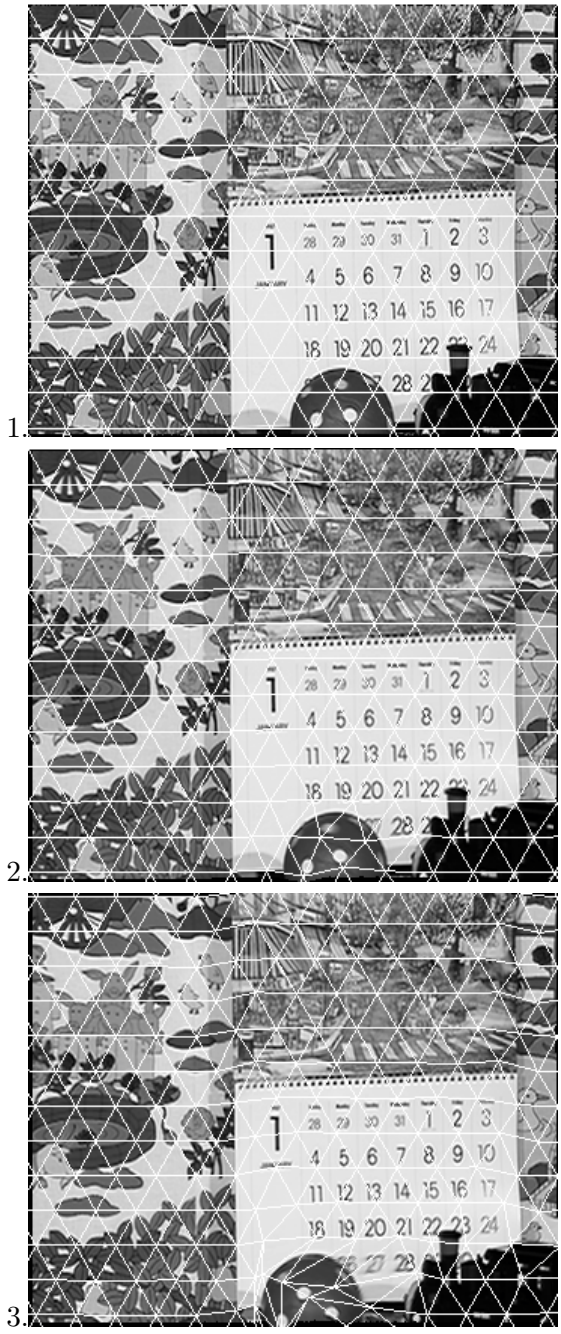


FIG. 5.3 – *Conséquences des occlusions sur le maillage: suivi du mouvement sur la séquence Mobile And Calendar. On observe des étirements et des tassements de maille entre le ballon et le calendrier, produisant un étalement et une contraction des textures dans les images prédites.*

5.3 Méthodes globales

Dans les techniques globales de gestion des occlusions, on distingue deux catégories: les méthodes par post-traitements agissant sur les vecteurs mouvement après l'estimation, et les méthodes par contraintes agissant durant l'estimation du mouvement.

5.3.1 Post-traitements des vecteurs mouvement

Dans [Toklu 97], les auteurs proposent un suivi d'objets par maillage 2D. Ils proposent une méthode pour la détection des zones recouvertes. En effet, le mouvement dans les zones recouvertes entre une image t et une image t+1 ne peut pas être estimé correctement puisque la zone n'est plus présente dans l'image t+1. Les auteurs proposent d'effectuer l'estimation du mouvement sur toute l'image, ou sur tout le support de l'objet dans le cas d'un suivi d'objets. Puis, après l'estimation du mouvement, les régions recouvertes sont détéctées par seuillage de la DFD entre l'image t+1 et l'image t+1 prédite. Les vecteurs mouvement des nœuds contenus dans les régions en défauts sont alors interpolés à l'aide du mouvement des nœuds visibles.
Dans [Altunbasak 97], une technique de post-traitement s'applique aux nœuds du maillage qui créent des retournements de mailles. Pour chaque nœud, l'algorithme considère le polygone formé par les nœuds auxquels il est lié, figure 5.4. Si le nœud sort de ce polygone, il crée un retournement. Son mouvement est alors interpolé à partir du mouvement des nœuds qui l'entourent.
Dans [Laurent 00], les zones d'occlusion, de recouvrement et de découvrement, sont détectées à l'aide des mailles retournées. Une nouvelle estimation du mouvement est effectuée en excluant les nœuds des zones en défauts. En effet, lors de la précédente estimation, les nœuds contenus dans les zones d'occlusion peuvent perturber l'estimation du mouvement dans les zones non en défauts. Les nœuds en défaut sont ensuite traités séparément soit par propagation du mouvement des zones continues vers les zones en défaut, soit par fusion de sommets, soit par codage intra de la zone en défaut.

5.3.2 Estimation contrainte des vecteurs mouvement

L'estimation contrainte des vecteurs mouvement permet de gérer le problème d'occlusion avant l'apparition des mailles dégénérées ou retournées.
Dans [Wang 94] et dans [van Beek 99], le déplacement des nœuds est contraint dans une région formée par les nœuds entourant le nœud courant. Dans [Wang 94], les auteurs utilisent un maillage quadrangulaire et la région de contrainte est définie par un quadrilatère formé par les quatre nœuds les plus proches du nœud courant. Un nœud déplacé qui sort du quadrilatère de contrainte ou qui est très proche des limites de ce quadrilatère est considéré comme ayant un mouvement créant des dégénérescences. Afin de limiter les perturbations et d'éviter l'apparition des dégénérescences, le déplacement du nœud est limité et le nœud est projeté à l'intérieur d'un quadrilatère plus petit que celui de contrainte, figure 5.5.
Dans [van Beek 99], le déplacement des nœuds est aussi limité à l'aide d'un quadrilatère englobant. Les auteurs utilisent un maillage hiérarchique et la limitation du mouvement

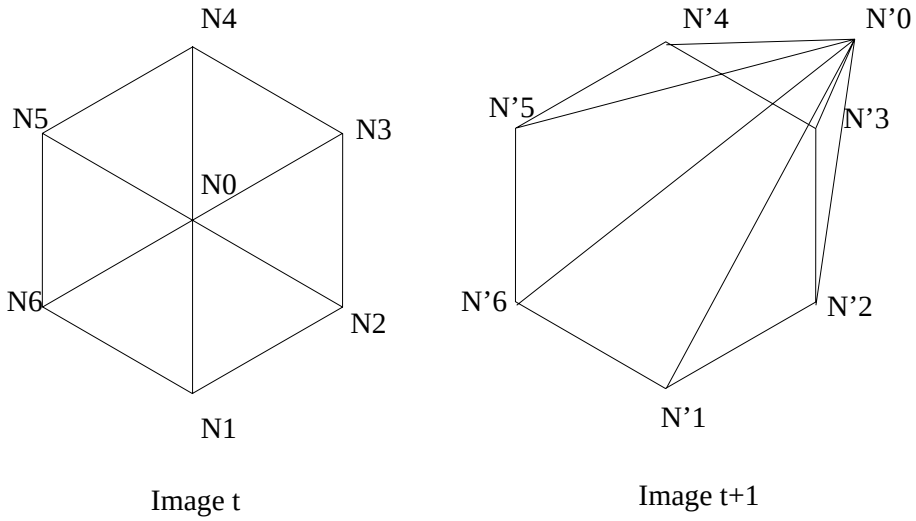


FIG. 5.4 – *Nœud entraînant des retournements*

des nœuds est testée de manière itérative à chaque niveau de hiérarchie. La région de contrainte est définie de manière à ce que le déplacement d'un nœud respecte la topologie initiale du maillage. La région de contrainte est définie par l'intersection des demi-plans des triangles connectés au nœud, figure 5.6. Dans le cas de maillage régulier, la région de contrainte est similaire à l'hexagone défini dans [Altunbasak 97].
La propagation contrainte des nœuds est faite de manière itérative. A chaque itération, les positions des nœuds voisins du nœud à mettre à jour sont fixées. La région de contrainte est diminuée à une certaine échelle, l'intersection de cette région et du vecteur mouvement estimé détermine le vecteur mouvement appliqué à l'itération i. Le processus converge vers le vecteur mouvement estimé si celui-ci respecte les contraintes de connectivité du maillage.

L'estimation du mouvement peut aussi être contrainte dans l'équation d'estimation du mouvement. Dans ce cas, un terme de pénalité est ajouté à l'équation à minimiser [Wang 96]. Dans [Laurent 98], les auteurs proposent d'utiliser un lagrangien augmenté portant sur la compacité des triangles. La compacité des triangles indique le degré de tassement du triangle, quand elle tend vers zéro, le triangle tend à se retourner. L'utilisation d'une telle contrainte permet d'agir avant le retournement du triangle, à terme cependant, elle n'empêche pas l'élongation des mailles.

Les techniques de corrections à posteriori ou de pré-traitement dans l'estimation de mouvement ont tendance à forcer les valeurs des vecteurs mouvement pour qu'ils respectent certaines contraintes. Ces solutions provoquent alors des mouvements biaisés. Elles offrent une représentation continue du mouvement dans les zones à problèmes mais ne donnent pas une représentation exacte du mouvement dans ces zones.

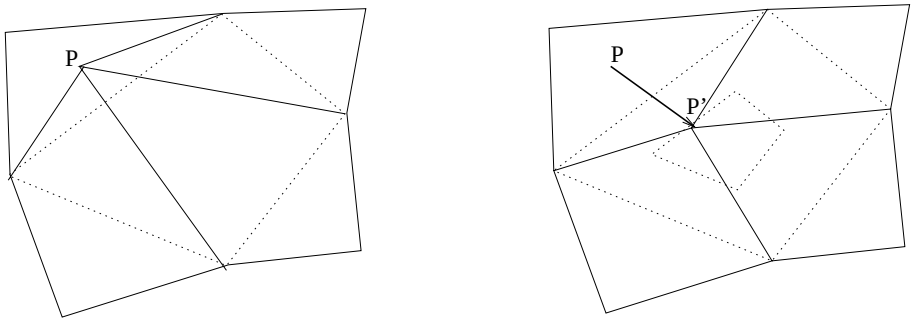


FIG. 5.5 – *Région de contrainte: le nœud P sort du quadrilatère de contrainte, il est projeté sur un quadrilatère plus petit*

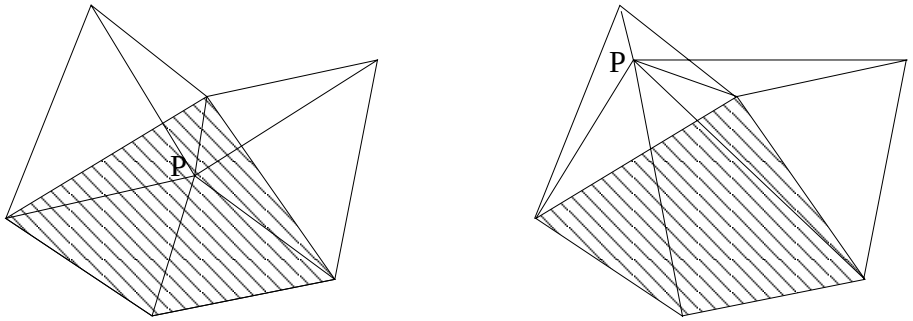


FIG. 5.6 – *Région de contrainte définie par l'intersection des demi-plans*

5.4 Maillage adaptatif

Les techniques présentées précédemment tentent de résoudre le problème des mailles retournées ou dégénérées de manière globale dans l'équation d'estimation du mouvement ou bien par correction a posteriori des vecteurs mouvement créant de trop grandes déformations.
Dans cette section, les techniques tentent d'adapter localement le maillage aux zones d'occlusion. Pour cela, elles tiennent compte du contenu des séquences vidéo.

5.4.1 Adaptation locale du maillage

Dans [Altunbasak 97], les auteurs proposent une technique de maillage qui s'adapte au contour des objets en avant-plan bien que le maillage soit défini sur toute l'image. Deux catégories de régions posant problème se distinguent: les région de type BTBC (Background To Be Covered, arrière-plan qui va être recouvert), et les régions de type UB (Uncovered Background, arrière-plan découvert). Le mouvement pour ces deux types de régions ne peut pas être estimé de manière correcte entre deux images t et t+1 car les régions UB ne sont pas présentes à t et les régions BTBC ne sont plus présentes à t+1.
Les auteurs proposent d'adapter le maillage dans ces régions. Comme les régions BTBC disparaissent, aucun nœud n'est placé dans ces régions. Dans une région UB, un nœud de l'image doit pouvoir se dédoubler dans t+1, il doit donc avoir deux mouvements différents: un suivant l'objet qui découvre et un restant sur le fond découvert. Ce principe est illustré par la figure 5.7.
Les régions BTBC sont détectées après une première estimation du mouvement par un seuillage de la DFD entre l'image compensée et l'image originale. Les frontières des régions sont approximées par des polygones et le maillage est redessiné dans ces zones en suivant les frontières et sans positionner de nœuds. Les positions des nœuds sont ensuite affinées. La compensation en mouvement de t vers t+1 permet de détecter les régions UB par seuillage de la DFD. Comme ces régions ne sont pas présentes dans l'image t, aucun mouvement ne peut être calculé pour ces zones entre t et t+1. Cependant, les régions UB sont remaillées dans l'image t+1 afin de les prendre en compte pour l'estimation du mouvement entre les images futures.

5.4.2 Maillages basés objets

L'utilisation de maillages basés objets constitue également une solution à la gestion des zones d'occlusions. Dans une telle représentation, un maillage est associé à chaque objet de la séquence vidéo et peut bouger indépendamment des autres objets.
Dans [Dudon 96, Buhan 98], le maillage de chaque objet est contraint sur les bords de l'objet. Ceci conduit à une polygonalisation des contours de l'objet qui nécessite un grand nombre de nœuds pour représenter un contour de manière précise. Les mailles engendrées peuvent alors être très petites ou très allongées provoquant des perturbations dans l'estimation du mouvement. Dans [Buhan 98], l'auteur propose une représentation

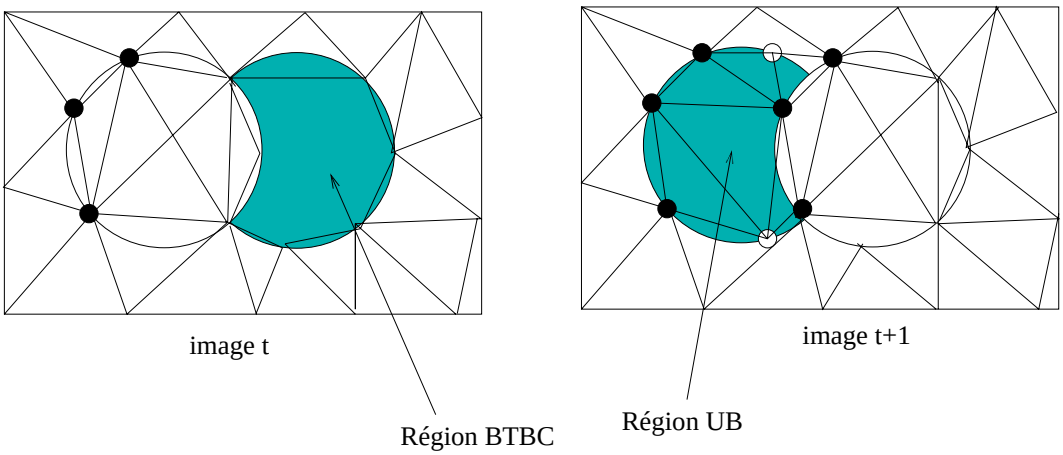


FIG. 5.7 – *Adaptation locale du maillage: régions BTBC et UB*

progressive de la forme des objets et permet une représentation polygonale du contour avec moins de nœuds. Une représentation grossière du contour des objets peut être raffinée progressivement créant des triangles altérés. Ces triangles s'adaptent alors facilement au contour de la forme.
Dans [Eren 03], le maillage d'un objet est défini à l'aide d'une carte de plan alpha, les bords des mailles ne sont pas forcés au contour de l'objet.
Les maillages basés objets permettent d'offrir une solution au problème d'occultations des objets mais des problèmes persistent lors de l'apparition et la disparition de textures. Lors de la disparition de texture, des tassements de mailles en bord d'images ou d'objets sont créés. De même, l'apparition de nouvelles textures crée des trous qui nécessitent des remaillages ou des corrections locales du mouvement créant des mouvements perturbateurs. Dans d'autres cas, un codage intra ([Dudon 96, Buhan 98, Laurent 00]) ou une estimation bi-directionnelle ([Eren 03]) permettent de représenter l'information nouvelle.

5.4.3 Lignes de rupture

La notion de lignes de rupture a été introduite dans [Marquant 00]. L'utilisation des lignes de rupture a pour but de fournir une solution au problème des occlusions dans une séquence vidéo. Dans cette sous-section, nous allons tout d'abord définir la notion de lignes de rupture telle qu'elle est présentée dans [Marquant 00], puis nous étudierons la mise en œuvre que l'auteur a proposé.

Une ligne de rupture est une discontinuité du champ de mouvement modélisable par une courbe. Dans une représentation par maillages, les mailles de part et d'autre de la ligne de rupture peuvent bouger indépendamment. Cette solution permet d'éviter les tassements et étirements de mailles aux frontières de la ligne de rupture qui sont dues à la discontinuité du mouvement dans cette zone.

La figure 5.8 illustre la représentation par maillage associée à une ligne de rupture. La figure 5.8-a représente un maillage traversé par une discontinuité. La ligne de rupture est dessinée en trait gras. La ligne de rupture permet de briser la structure du maillage. Lorsque les deux objets de part et d'autre de la ligne bouge en mouvement contraire, le maillage se brise et les mailles accrochées de part et d'autre de la ligne de rupture bougent indépendamment, figure 5.8-b. Les mailles brisées sont complétées et le maillage brisé de part et d'autre de la ligne est prolongé du côté opposé (figure 5.8-c) de manière indépendante pour chaque côté.
La prolongation du maillage permet de définir un maillage sur les zones invisibles à un instant t qui pourraient devenir visibles à t+k.
Dans [Marquant 00], l'auteur distingue trois types de lignes de rupture:

- type 1: les lignes de rupture externes, qui correspondent aux contours d'un objet. Ce type de lignes de rupture permet de gérer les apparitions/disparitions ou croisements d'objets dans une scène.
- type 2: les lignes de rupture internes, qui correspondent aux contours appartenant à un même objet. Par exemple: une main qui passe devant un corps, le corps et la main appartiennent à la même entité.
- type 3: les lignes de rupture internes ouvertes. Par exemple: le contour du nez sur un visage.

La notion de ligne de rupture permet l'adaptation du maillage au contenu de la séquence vidéo dans les zones où il existe une discontinuité du mouvement. Elle permet d'éviter les retournements, étirements et tassements de mailles générant des problèmes au niveau de l'estimation du mouvement et du codage de la séquence vidéo.
Les lignes de rupture permettent également d'avoir une structure de maillage cohérente avec les évènements de la séquence vidéo. Elles permettent de gérer les recouvrements/découvrements d'objets en considérant qu'un nœud peut être visible ou non. Si le nœud est visible, les valeurs stockées permettent d'effectuer la compensation en mouvement dans la zone. Si le nœud est invisible, les valeurs stockées permettent d'enregistrer l'historique du nœud et de disposer de sa valeur quand il (ré)apparaîtra.
Les difficultés liées à l'utilisation des lignes de rupture restent la détection de ces lignes, leur représentation et leur codage. De plus, la mise en œuvre d'une structure de maillage prenant en compte les lignes de rupture n'est pas immédiate.

Dans [Marquant 00], l'auteur ne traite que les lignes de rupture externes, c'est-à-dire les lignes de rupture correspondant aux contours des objets.
Les lignes de rupture sont détectées et représentées à l'aide des masques de segmentation disponibles pour les séquences vidéo traitées. Un maillage régulier débordant du support de l'objet est généré pour chaque objet. Le mouvement est estimé indépendamment pour chaque maillage.
Cette technique revient à un fonctionnement en mode objet présenté dans une section précédente 5.4.2. Bien qu'elle utilise la notion de lignes de rupture, la technique est soumise aux mêmes problèmes que les représentations en mode objet. Il est nécessaire de connaître la segmentation des objets parfaitement et sur toute l'image.

Conclusion

Nous avons présenté dans ce chapitre les solutions proposées dans la littérature permettant de gérer le problème des occlusions dans une séquence vidéo.
Les méthodes par post-traitement ou par contrainte sur les vecteurs permettent d'obtenir une représentation continue du mouvement à l'aide d'un maillage mais pour cela, elles introduisent de faux mouvements qui conduisent à de mauvaises prédictions dans les zones en défauts.
Les techniques d'adaptation locale du maillage à la zone d'occlusion nous paraissent les plus intéressantes. Elles tentent de donner une représentation du mouvement en considérant les discontinuités du mouvement dans les zones où elles apparaissent.
Dans le chapitre suivant, nous proposons une solution de maillage adaptatif basée sur la notion de ligne de rupture expliquée ici.

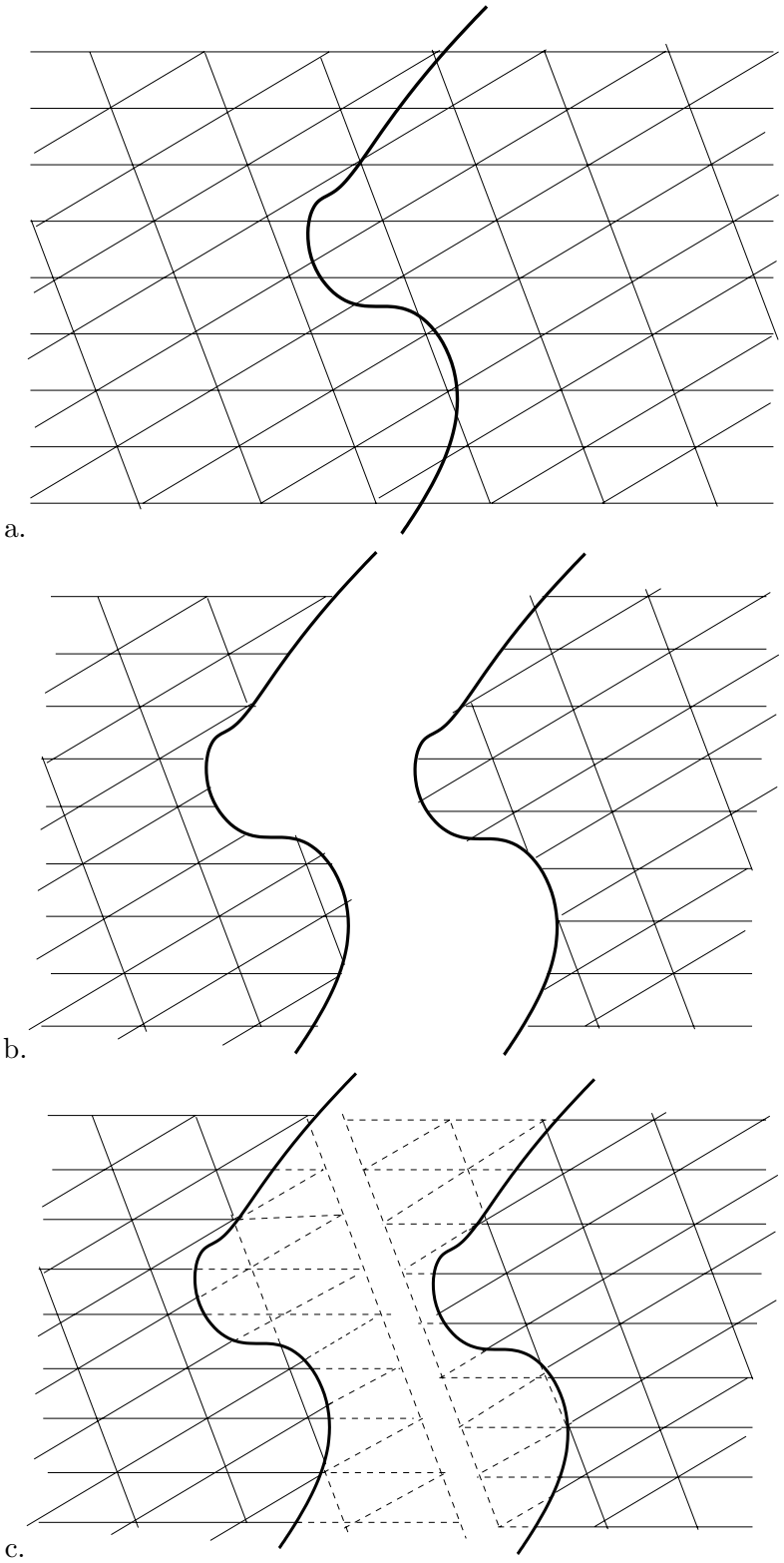


FIG. 5.8 – *Maillage et ligne de discontinuité*

Chapitre 6

Gestion des zones d'occlusions: approches proposées

Afin de gérer le problème des occlusions dans les techniques d'estimation du mouvement par maillages, nous proposons dans ce chapitre une nouvelle représentation du mouvement basée sur les maillages et la prise en compte d'une zone de discontinuité dans le maillage. Nous créons délibérement une déchirure dans le maillage le long d'une ligne de discontinuité, puis remaillons l'intérieur de la zone de discontinuité avec recouvrement de mailles afin de gérer les éventuels découvrements.
Après une brève présentation du processus de création de cette structure, les sections 2 et 3 expliquent plus en détail chaque étape du processus. Enfin, la quatrième section présente la méthode d'estimation du mouvement.

6.1 Les maillages et les discontinuités du mouvement

Quand des objets se croisent, apparaissent ou disparaissent dans une séquence vidéo, il en résulte des discontinuités de mouvement. Ces discontinuités créent des étirements et tassements voire des retournements de mailles lors de l'estimation du mouvement. Dans ces zones, la texture est difficilement prédictible et la compensation en mouvement crée des motifs chers à coder.
Pour résoudre ce problème, une solution appelée lignes de rupture a été présentée dans le chapitre précédent. Cette solution est proposée dans [Marquant 00]. L'idée est de modéliser la discontinuité du mouvement à l'aide d'une courbe et de briser la structure de maillage le long de cette courbe. De chaque côté de la ligne de rupture, le maillage brisé est complété et prolongé indépendamment de chaque côté de la ligne de rupture. Lors de l'estimation du mouvement, les mailles de part et d'autre de la ligne de rupture bougent de manière indépendante. Cette technique permet d'éviter l'apparition de mailles dégénérées dans la zone de dicontinuité du mouvement. Elle permet aussi d'offrir une représentation du mouvement proche du mouvement réel des objets 3D réels.
La mise en œuvre de cette technique dans [Marquant 00] utilise la notion d'objets

vidéo. Un maillage indépendant pour chaque objet vidéo est généré et le mouvement est estimé de manière indépendante pour chaque objet. Cette représentation est assez contraignante car elle nécessite la connaissance a priori de la segmentation de la séquence vidéo.
Nous proposons ici une mise en œuvre plus générique des lignes de rupture sans utiliser la notion d'objets vidéo. Nous proposons une nouvelle structure de maillage hiérarchique prenant en compte les lignes de discontinuité du mouvement localement à la zone d'occlusion.
Après la détection de la zone d'occlusion et de la ligne de discontinuité, le maillage est découpé localement dans la zone d'occlusion et un remaillage est effectué de part et d'autre de la ligne de discontinuité. Le remaillage et la discontinuité sont remontés dans la hiérarchie afin de conserver une structure hiérarchique cohérente du maillage.

6.2 Détection de la zone d'occlusion et de la ligne de discontinuité

6.2.1 Détection de la zone d'occlusion

La zone d'occlusion est détectée après une première estimation du mouvement. Cette première estimation a laissé les mailles se déformer dans les zones de discontinuité. La mesure de cette déformation indique quelles mailles se trouvent dans la zone d'occlusion. Le mouvement estimé entre les images est un mouvement affine. On s'intéresse alors à la déformation des mailles selon la transformation affine estimée. L'équation de la transformation affine est donnée ci-dessous, pour un nœud N de position initiale (x, y), la nouvelle position (X, Y) de N est:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

Dans le cas de mailles triangulaires, chaque maille possède trois nœuds. On dispose donc de six équations à chaque maille pour trouver les six paramètres de mouvement affine. On ne s'intéresse qu'à la matice de changement d'échelle

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

les paramètres de translation (e et f) ne font que déplacer la maille sans la transformer.
On considère un cercle de rayon 1 qui se déforme en une ellipse selon la transformation affine estimée A (figure 6.1). Le cercle se déforme selon deux axes principaux u et v avec deux facteurs de zoom respectifs λ_1 et λ_2. Comme la matrice A n'est pas toujours diagonalisable, on étudie la matrice $A^t A$ [1] qui est symétrique et définie positive, elle est

1. A^t est la matrice transposée de A

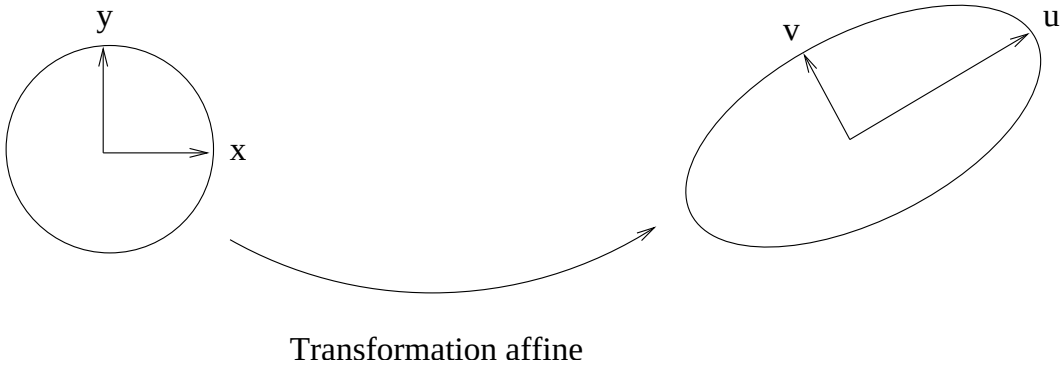


FIG. 6.1 – *Transformation d'un cercle en ellipse*

donc diagonalisable en la matrice D:

$$D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$$

λ_1 et λ_2 représentent les changements d'échelle respectivement en u et v. Ces changements d'échelle traduisent les déformations de la maille selon l'axe u et l'axe v. On étudie le rapport $\alpha = \frac{\lambda_1}{\lambda_2}$. Si la maille est aplatie, α tend vers 0, si elle est étirée, α est très grand.
Les valeurs de λ_1 et λ_2 pour chaque maille sont obtenues à l'aide des équations suivantes:

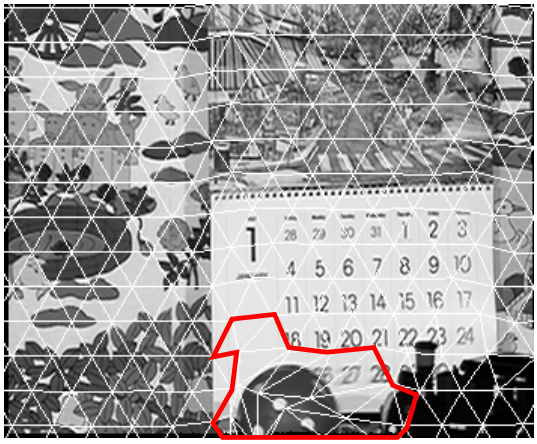
$$Tr(A^tA) = \lambda_1 + \lambda_2$$

$$Det(A^tA) = \lambda_1\lambda_2$$

$$\lambda^2 - Tr(A^tA)\lambda + Det(A^tA) = 0$$

où $Tr(A^tA)$ est la trace de la matrice A^tA, $Det(A^tA)$ le déterminant de A^tA et λ représente λ_1 ou λ_2.
Pour étudier la divergence de α par rapport à 1, on propose alors d'étudier la valeur de $c = \alpha + \frac{1}{\alpha}$. Si c tend vers 2, α tend vers 1, la déformation en u et v est la même, on considère alors que la déformation n'est pas dégénérante et que c'est un changement d'échelle globale sur toute la maille. Le changement de résolution est le même pour tous les points de la texture.
Si c très grand devant 2, on considère que la déformation est dégénérante et que la maille est dans une zone d'occlusion.
On définit un voisinage autour de l'ensemble des mailles dégénérées obtenus par une succession de dilatations par un k-disk. Les mailles dégénérées et le voisinage forment la zone d'occlusion. La détection de la zone d'occlusion est illustrée par la figure 6.2.

a. Image des valeurs du critère c



b. Mailles dégénérées détectées avec le critère

c. Zone d'occlusion: mailles dégénérées + dilatation de la zone avec un k-disk

FIG. 6.2 – *Détection de la zone d'occlusion définie par les mailles dégénérées*

6.2.2 Détection et représentation de la ligne de discontinuité

Une zone de discontinuité dans le maillage se crée quand deux objets ou deux parties d'un même objet ont un mouvement contraire ou trop ample. La ligne de discontinuité est alors positionnée à la frontière de ces deux objets, sur le contour, elle est portée par l'objet en avant-plan. Ce contour est orienté afin de définir une région intérieure (région en avant-plan) et une région extérieure (région en arrière-plan).
Dans notre étude, nous nous sommes intéressés à la mise en place d'une structure de maillage prenant en compte des lignes de discontinuité en faisant l'hypothèse que les lignes étaient déjà connues et placées sur le contour des objets. Nous n'avons pas fait d'étude approfondie concernant la recherche du contour dans la zone d'occlusion, ni sa représentation. Pour cette raison, les méthodes utilisées pour trouver les lignes de discontinuité ne sont pas automatiques. Nous proposons ici les quelques solutions que nous avons envisagées.
La ligne de discontinuité est obtenue à partir des masques de segmentation des séquences vidéo. Si l'on ne dispose pas des masques de segmentation des images de la séquence, ceux-ci sont créés manuellement.
Dans la zone d'occlusion détéctée par la méthode présentée dans la section précédente, le contour est alors extrait à partir des masques.
Afin d'automatiser la technique de détection de la ligne de discontinuité, une méthode par contours actifs peut être utilisée. Cette technique n'a pas été mise en œuvre dans cette étude, nous présentons juste ici une piste de recherche qui permettrait d'automatiser le système de détection de la ligne de discontinuité.
Un contour actif est une courbe qui grâce à la minimisation d'une fonctionnelle se déforme et tend à se placer sur le contour de l'objet à segmenter. La fonctionnelle à minimiser comprend un terme d'énergie interne et un terme d'énergie externe. L'énergie interne traduit les contraintes d'élasticité et de rigidité fixées au contour tandis que l'énergie externe exprime l'attache aux données, elle est souvent exprimée par le gradient de luminance de l'image.
L'inconvénient de cette technique est que la convergence du système est très dépendante de l'initialisation et souvent très lente. Cependant, dans notre approche, la région à segmenter est limitée à la région d'occlusion détectée, ce qui permet de limiter le temps de calcul. De plus, il est fort probable que le contour à trouver dans la zone d'occlusion soit un gradient assez fort. La mise en œuvre d'une technique de segmentation par contours actifs dans la région d'occlusion nous paraît être une bonne approche pour la détection de la ligne de discontinuité de façon automatique.
La figure 6.3 illustre le positionnement de la ligne de discontinuité sur le contour de l'objet qui a créé la discontinuité dans la zone d'occlusion dont la détection est présentée sur la figure 6.2. La figure 6.4 illustre la notion de région intérieure/extérieure.

6.3 Remaillage de la zone d'occlusion

Une fois la zone d'occlusion détectée (figure 6.5) et la ligne de discontinuité trouvée, la zone d'occlusion est remaillée de part et d'autre de la ligne de discontinuité (figure

FIG. 6.3 – *Positionnement de la ligne de discontinuité sur le contour de l'objet créant la discontinuité*

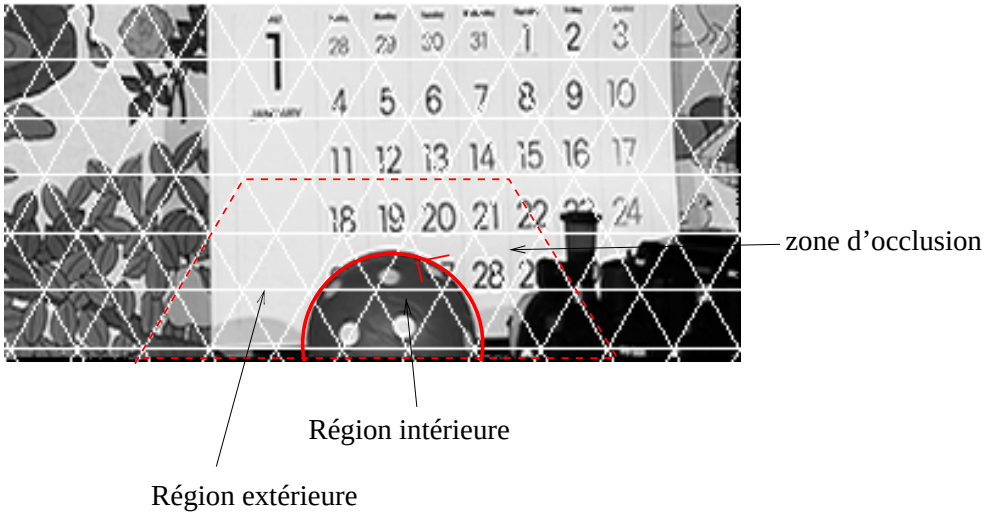


FIG. 6.4 – *Zone d'occlusion: régions intérieure et extérieure*

6.9).
A l'estimation du mouvement, les nouvelles mailles de chaque région peuvent bouger librement dans la zone d'occlusion. La rigidité du maillage qui avait provoqué une dégénération des mailles n'existe plus en cet endroit.

6.3.1 Remaillage de la région d'occlusion au niveau fin de la hiérarchie

Le remaillage consiste tout d'abord à fixer les degrés de liberté du nouveau maillage aux bords de la zone d'occlusion. Ces degrés de liberté sont fixés tels que la reconstruction du contour au niveau de la frontière de la zone d'occlusion doit se faire sans ambiguïté. Pour cela, le contour est porté par les mailles de la région intérieure et les nœuds des arêtes qui intersectent le contour à la frontière de la zone sont contraints à bouger avec le maillage initial. Ces nœuds seront appelés nœuds bordure.
Les nœuds bordure peuvent être de trois types: nœuds bordure gauche, (nœuds servant de base pour le prolongement à gauche), nœuds bordure droit (nœuds servant de base pour le prolongement à droite), nœuds bordure partagés (nœuds partagés par les deux prolongements). La figure 6.6 montre les différents types de nœuds associés au prolongement des deux côtés.
L'identification des nœuds bordure se fait selon que la ligne de discontinuité traverse un triangle (les nœuds sont identifiés bordure gauche ou droite), ou bien qu'elle se termine en un triangle (les nœuds sont identifiés bordure partagés ou non). Les mailles contenues dans la zone d'occlusion sont ensuite détruites, et pour chaque région, de nouvelles mailles sont créées avec pour base les nœuds bordure de la région.
Les figures 6.7 et 6.8 illustrent la création des nouvelles mailles selon que le contour traverse entièrement la maille ou finit en cette maille.
La figure 6.7 montre le contour traversant entièrement la maille. Les nœuds C,F et B sont identifiés nœuds bordure pour le côté bas. Pour le côté haut, les nœuds bordure sont E et D. Pour le côté bas, de nouvelles mailles sont créés: CFE', FE'D' et FD'B'. Pour le côté haut, les nouvelles mailles créées sont: EC'F',EDF' et DF'B'.
La figure 6.8 illustre le cas du contour finissant en une maille. Sur la figure, le contour finit sur l'arête EF. Les nœuds de l'arête EF sont alors identifiés comme nœuds bordure partagés. Ils seront nœuds bordure pour les deux côtés. Pour le côté bas, les nœuds bordure sont alors E, F et B et les nouvelles mailles créées sont EFD' et FD'B. Pour le côté haut, les nœuds bordure sont E, F et D, la maille EDF est conservée et appartient au côté haut, une nouvelle maille est créée: FDB'.

Afin de prendre en compte d'éventuels découvrements pouvant intervenir dans la séquence vidéo, les nouvelles mailles sont prolongées au-delà de la zone d'occlusion, comme sur la figure 6.9. Des mailles du maillage initial et de nouvelles mailles se chevauchent alors, nous verrons dans la sous-section suivante la technique employée lors de la reconstruction d'une image pour éviter les conflits de mailles.

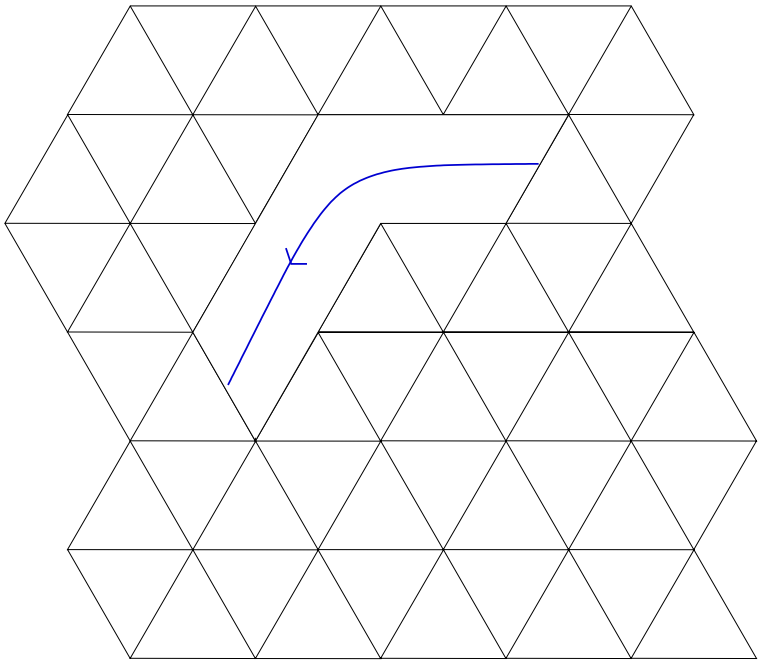


FIG. 6.5 – *Désactivation de la zone d'occlusion dans le maillage*

6.3.2 Méthode du z-order

Lors du remaillage de la zone d'occlusion de part et d'autre de la ligne de discontinuité, des mailles viennent à se chevaucher. Les chevauchements de mailles posent problème lors de la reconstruction d'une image à partir de la nouvelle structure de maillage car alors un point peut être prédit par plusieurs mailles. Afin de gérer ce problème de recouvrement, une technique de z-order et des masques de visibilité sont utilisés.
Une valeur de z (profondeur) est affectée aux nœuds de la zone remaillée. La valeur affectée est fonction que le nœud appartient à l'objet en avant-plan ou non. Les nœuds des mailles portant le contour ont un z positif, les autres nœuds ont un z négatif.
Une valeur de z peut alors être calculée en chaque point de l'image par interpolation à partir des z des nœuds de la maille à laquelle appartient le point:

$$z_M = \sum_{i=1}^{i=3} w_i z_i$$

où z_i est le z du nœud i de la maille à laquelle le point M appartient et w_i son poids barycentrique par rapport au nœud i.
Un masque de visibilité est créé pour chaque maille portant le contour, il permet de définir quels points de la maille sont visibles. Le contour étant orienté, le masque de visibilité d'une maille portant le contour est défini par les points de la maille intérieurs au contour. Ceci est fait pour le niveau le plus grossier de la hiérarchie pour lequel

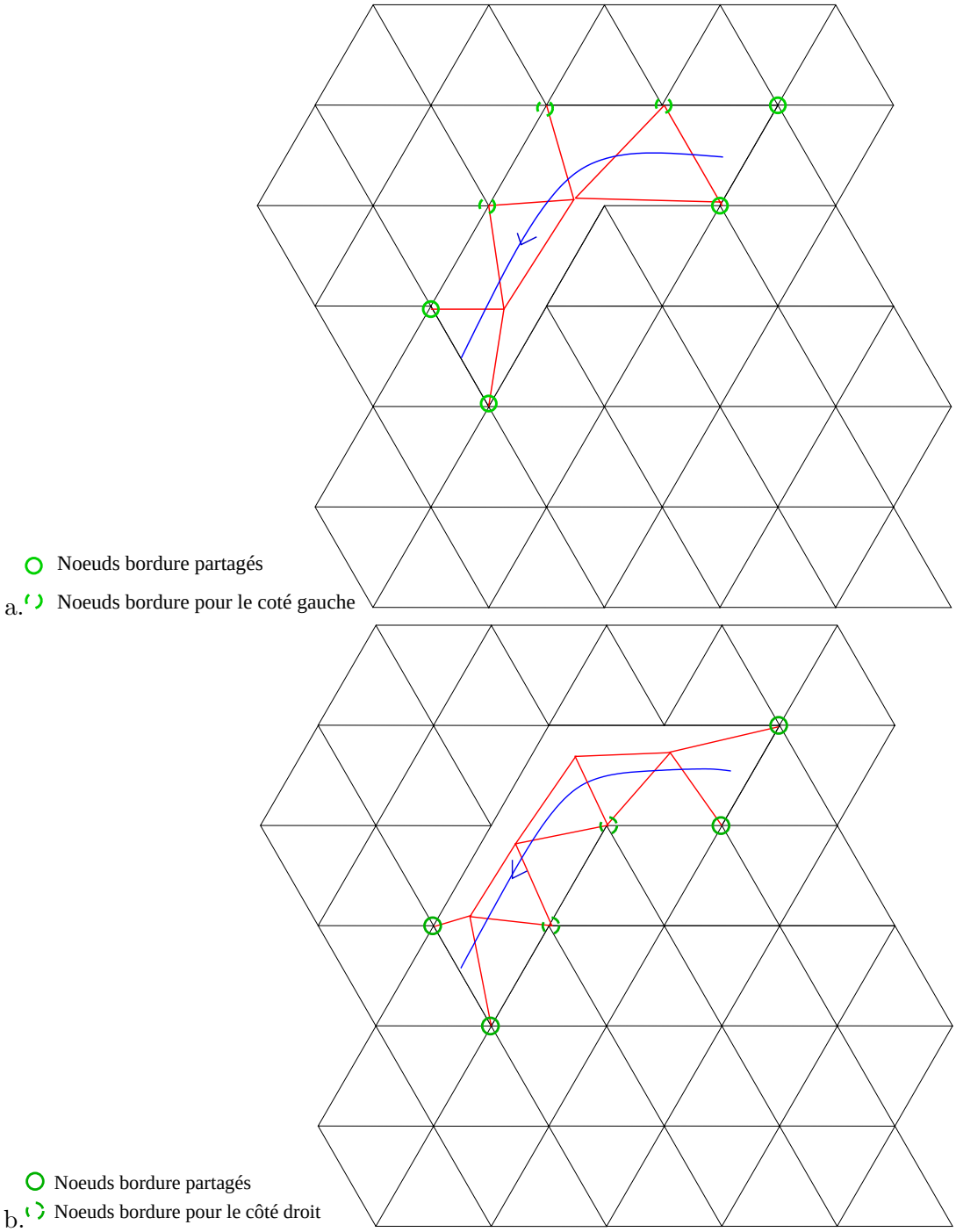


FIG. 6.6 – *Nœuds bordures et nœuds bordure partagés: a.pour le côté gauche, b.pour le côté droit*

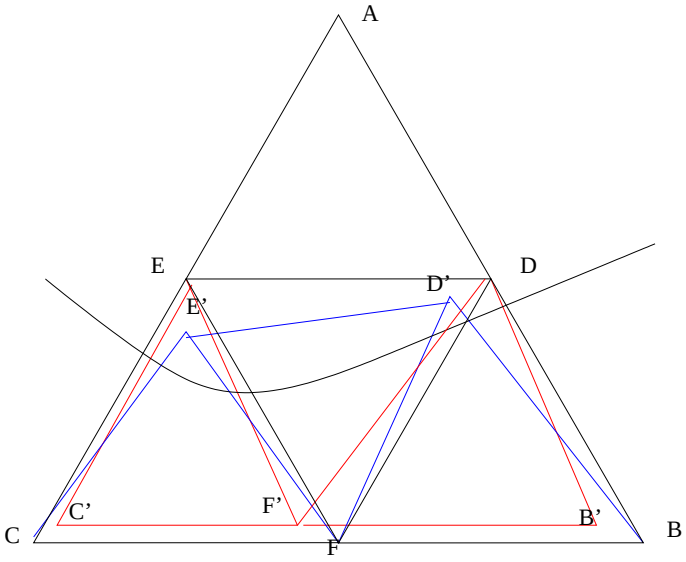


FIG. 6.7 – *Contour traversant entièrement la maille*

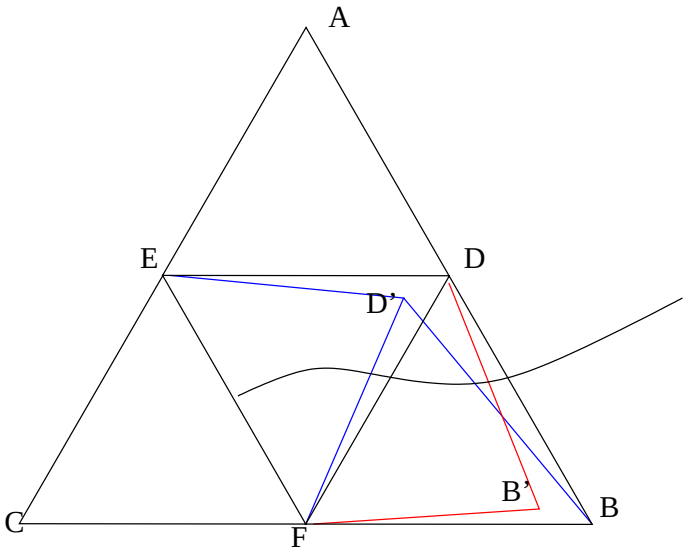
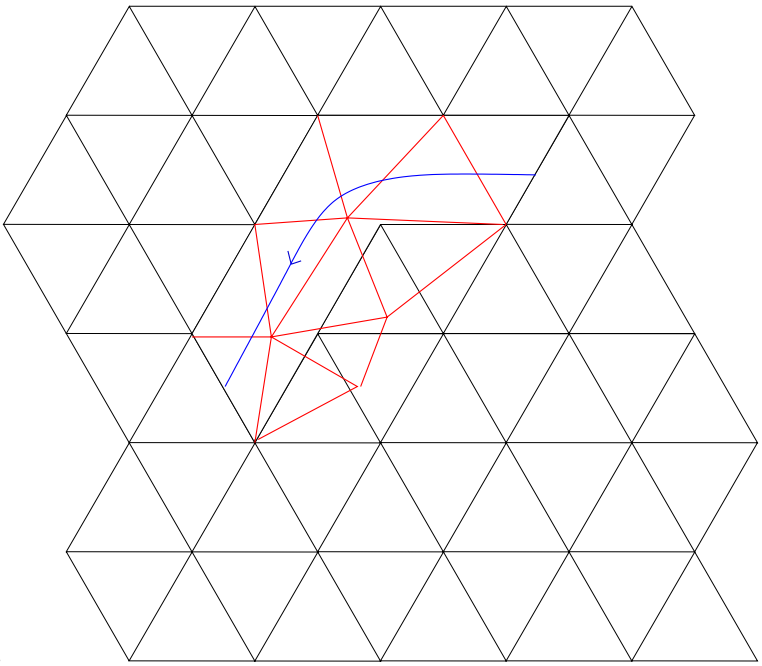


FIG. 6.8 – *Contour finissant dans une maille*

a.



b.

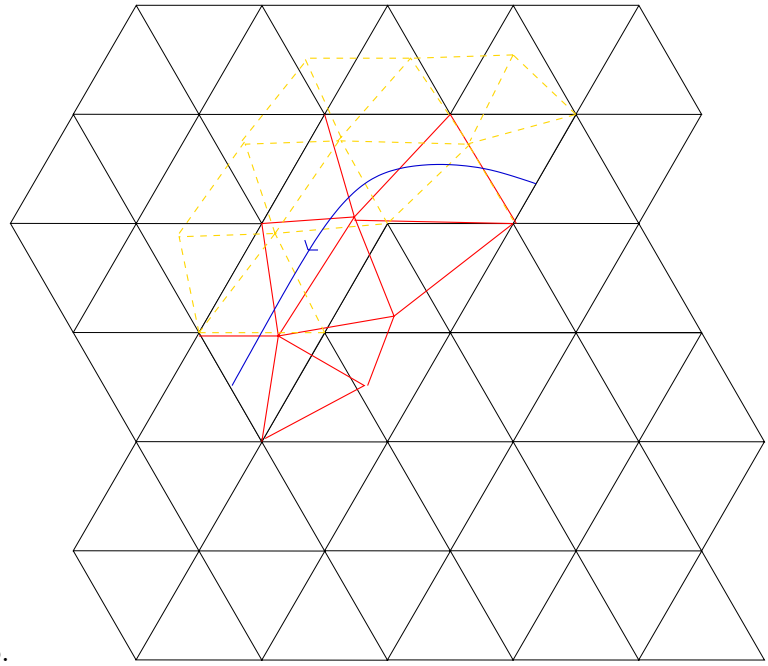


FIG. 6.9 – *Remaillage de la zone d'occlusion: a.pour un côté, b. pour les deux côtés*

la discontinuité est présente, puis le masque est propagé vers les mailles des niveaux plus fins afin de définir le masque de visibilité pour toutes les mailles du niveau fin de reconstruction et portant le contour.
Au niveau du codage, seul le contour orienté est codé. Les masques de visibilité sont reconstruits au moment du décodage à partir du contour décodé.
La figure 6.10 illustre la technique du z-order. Les mailles rouges portent le contour, les nœuds de ces mailles ont un z positif. Les mailles noires sont les mailles en arrière-plan, leur z est négatif. La reconstruction d'un point M de position (x,y) peut être effectuée par une maille rouge ou une maille noire. Si le point M est situé dans le masque de visibilité de la maille rouge, celle-ci reconstruit le point, sinon le point est reconstruit par la maille noire.
L'algorithme de reconstruction d'une image est alors le suivant:

```
imRecons: image à reconstruire
zRecons: z en chaque point de imRecons
m(M): fonction qui renvoit la valeur de la maille m au point M
z(m,M): fonction qui renvoit la valeur de z en M donnée par la maille m
Pour tous les points M de imRecons
  zRecons(M) = -1000;

Pour toutes les mailles m du niveau fin
 Pour tous les points M de m
   si m porte un contour
       {
        si M appartient au masque de visibilité
         {
           si z(m,M) > zRecons(M)
            {
              imRecons(M) = m(M);
              zRecons(M) = z(m,M);
            }
         }
       }
     sinon
      { //m ne porte pas de contour
        si z(m,M) > zRecons(M)
           {
             imRecons(M) = m(M);
             zRecons(M) = z(m,M);
           }
      }
 finpour
finpour
```

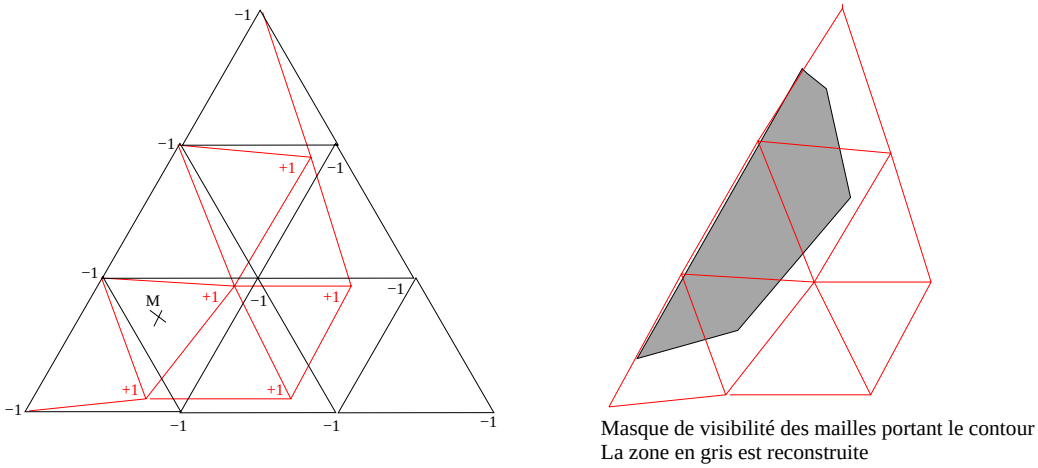


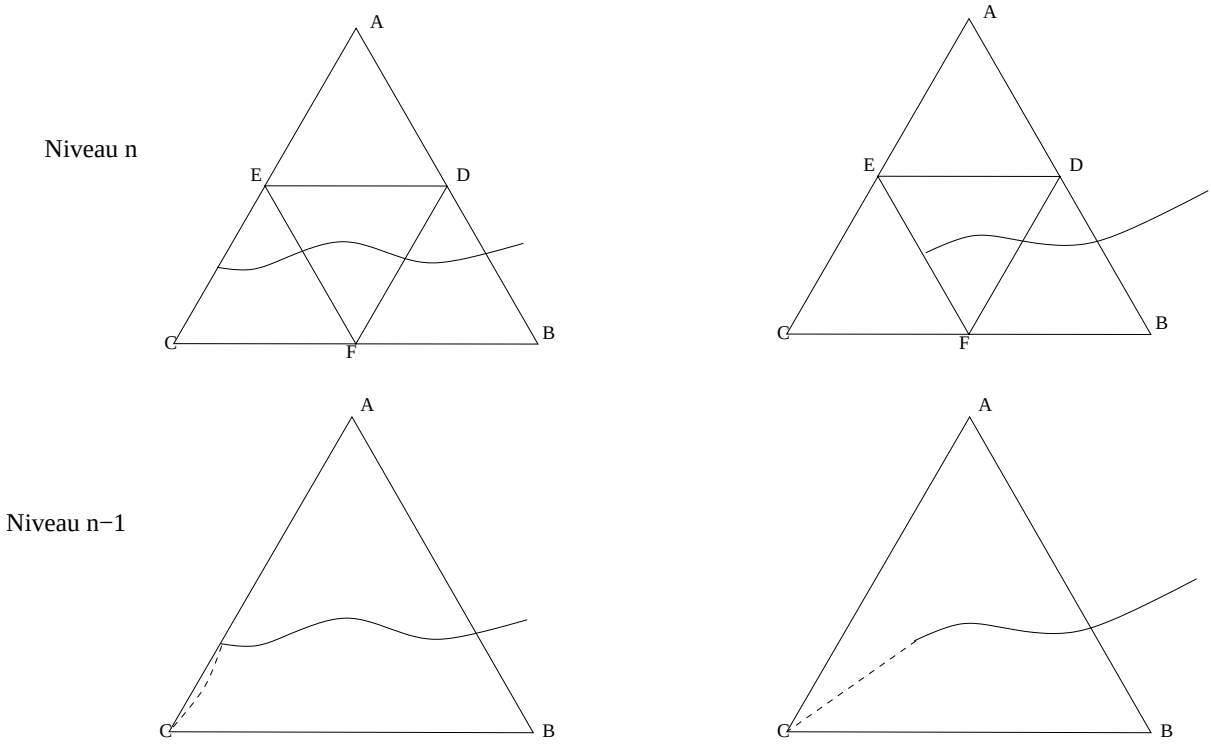
FIG. 6.10 – *Méthode du z-order, avec utilisation de masque de visibilité*

L'utilisation d'un z-order et de masques de visibilité pour les mailles portant le contour permet lors du codage du contour avec pertes de pouvoir reconstruire une image sans trous. En effet, si le contour est codé avec pertes, le masque de visibilité reconstruit à partir de ce contour sera déplacé par rapport à sa position originale et risque de faire apparaitre des zones découvertes qui seront alors reconstruites par les mailles en arrière-plan.

6.3.3 Remontée de la discontinuité en hiérarchie

Le remaillage décrit précédemment est effectué au niveau le plus fin de la hiérarchie. Suivant la taille de la zone d'occlusion, celle-ci peut avoir un impact sur l'estimation du mouvement dans les niveaux plus grossiers. Il est alors nécessaire de remonter la ligne de discontinuité dans la hiérarchie et de remailler les niveaux influencés par la zone d'occlusion.
Soit $nivFin$ le niveau de la hiérarchie le plus fin auquel le remaillage initial a été fait. Si pour un niveau $n \leq nivFin$, la discontinuité a été remontée, la zone d'occlusion au niveau $n-1$ est alors définie par l'ensemble des mailles pères des mailles de la zone d'occlusion du niveau n, c'est-à-dire toutes les mailles de n-1 contenant la discontinuité définie dans n.
Pour le niveau fin, le contour finit sur une arête et fixe les nœuds de cette arête à être partagées par les deux côtés. Pour le niveau grossier, le contour ne finit plus sur une arête. Deux cas sont alors possibles:

- si un des nœuds de l'arête du niveau fin a un père au niveau grossier, le contour est prolongé artificiellement jusqu'à ce nœud, figure 6.11-a.
- si aucun des nœuds du niveau fin n'a de père au niveau grossier, le contour est alors artificiellement prolongé jusqu'au nœud opposé à l'arête par laquelle le contour entre dans la maille, figure 6.11-b.



a. C a un père au niveau n−1

b. E et F n'ont pas de père au niveau n−1

FIG. 6.11 – *Prolongation du contour au niveau grossier*

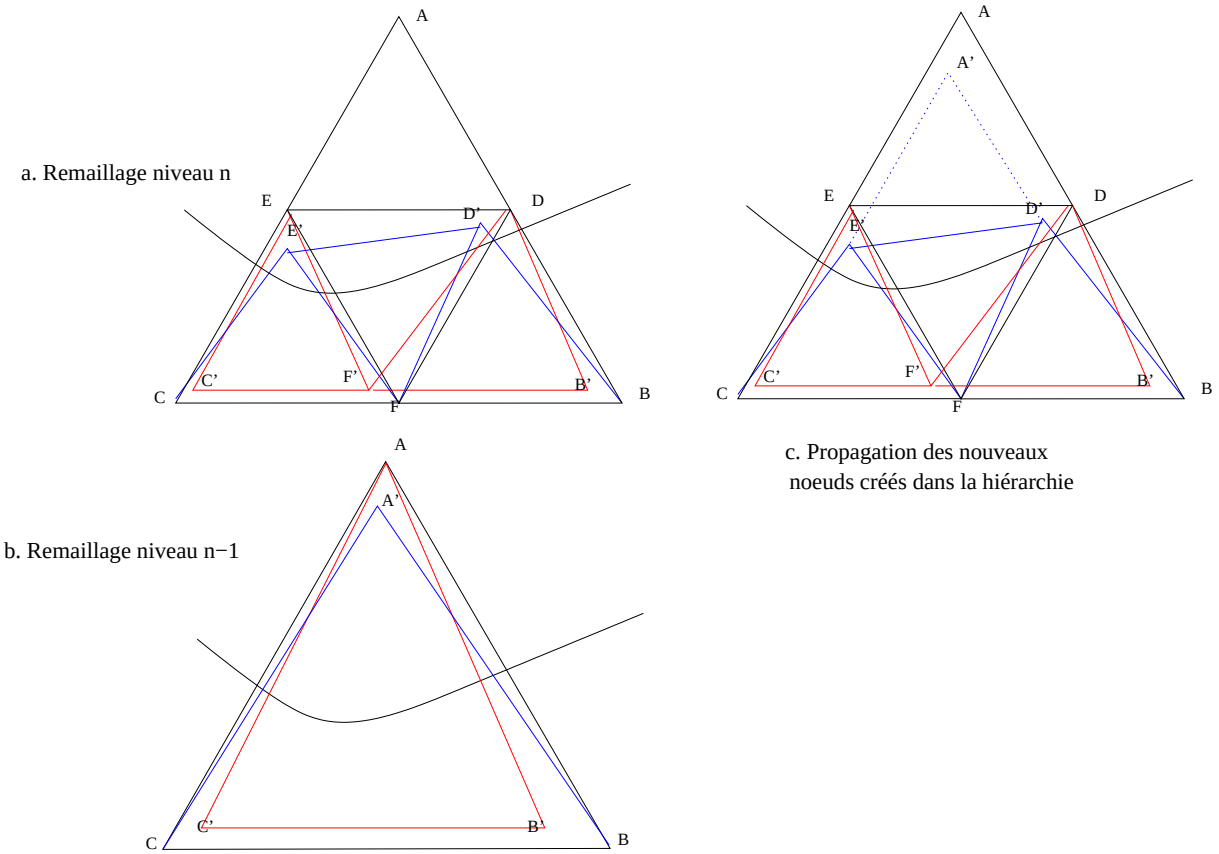


FIG. 6.12 – *Remontée en hiérarchie: Contour traversant entièrement la maille*

La création d'une nouvelle maille pour un niveau de hiérarchie $n-1$ inférieur à un niveau n qui a déjà été remaillé peut alors être de deux sortes suivant que:

– le contour traverse entièrement la maille, figure 6.12

– le contour ne traverse pas entièrement la maille, 6.13

Pour les figures 6.12 et 6.13, a. montre le remaillage au niveau fin, b. montre le remaillage à un niveau plus grossier (niveau n-1) et c. montre la propagation du remaillage du niveau n-1 vers les niveaux plus fins.
Le remaillage du niveau n-1 est effectué de la même manière qu'au niveau fin.

Dans le cas où le contour traverse entièrement la maille à dédoubler ABC,(figure 6.12-b), les nœuds bordure pour le côté bas sont B et C, un nouveau nœud A' est créé au niveau $n-1$. La nouvelle maille A'BC prend pour mailles filles les mailles du niveau n: A'E'D', E'CF,E'D'F,D'FB (figure6.12-a). Afin de respecter la topologie du maillage dans la hiérarchie, la maille A'E'D' est créée au niveau n (figure 6.12-c).
De même, pour le remaillage du côté haut, le nœud bordure est A. La nouvelle maille AB'C' a pour mailles filles: AED, EC'F', EDF' et DF'B'.

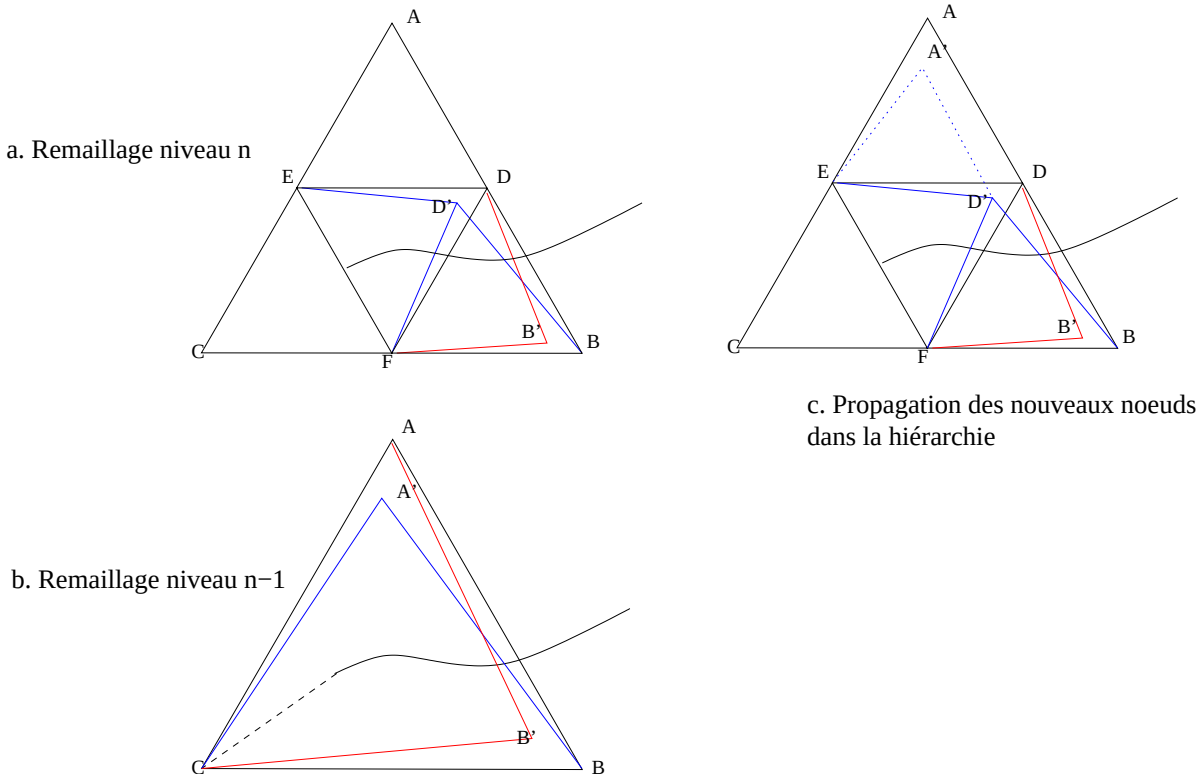


FIG. 6.13 – *Remontée en hiérarchie: Contour prolongé vers un nœud*

La figure 6.13 illustre le cas du contour ne traversant pas la maille entièrement. Le nœud C est nœud bordure partagé. Pour le côté bas, les nœuds bordure sont C et B. Le nœud A' est créé (figure 6.13-b) et la nouvelle maille A'BC prend pour mailles filles: CEF, FD'B,ED'F et A'ED'. La maille A'ED' a été créée au niveau n afin de respecter la topologie du maillage dans la hiérarchie (figure 6.13-c).
Pour le côté haut, les nœuds bordure sont alors C et A. La nouvelle maille ACB' prend pour maille filles: CEF, EFD, EAD et DFB'. On remarque que dans ce cas, la maille CEF est une maille fille partagée par A'BC et ACB'.

6.3.4 Disparition de la ligne de discontinuité

La remontée de la discontinuité dans la hiérarchie est faite jusqu'à sa disparition. Lorsque la ligne de discontinuité, définie au niveau de hiérarchie le plus fin, est entièrement contenue dans une maille, elle disparaît au niveau inférieur. Les nouveaux nœuds introduits au niveau courant pour la création de la nouvelle maille sont alors définis par leurs coordonnées barycentriques dans la maille père du niveau inférieur. Ces nœuds auront ainsi un mouvement global influencé par les nœuds de la maille du niveau inférieur.
La figure 6.14 montre le cas de la disparition de la discontinuité dans la hiérarchie. Au niveau n, le contour a été prolongé aux nœuds B et C. B et C sont devenus nœuds bordure partagés. Le remaillage pour le côté bas a introduit les nœuds E' et D', pour le côté haut F'. Au niveau $n-1$, le contour est entièrement contenu dans la maille ABC. Pour le côté bas, le nœud A' a été introduit pour former A'BC, pour le côté haut, aucun nœud n'a été introduit, le remaillage donne la maille initiale ABC.
La maille A'BC est contrainte à bouger avec le maillage initial ABC. Les nœuds A' et A au niveau $n-1$ sont virtuellement les mêmes. Au niveau n, A' existe et est défini par ses coordonnées barycentriques dans ABC du niveau $n-1$.

6.4 Estimation du mouvement

6.4.1 Remontée des valeurs dans l'approche multi-grille

Une fois la nouvelle structure de maillage construite, le mouvement est estimé de la même manière qu'avec la précédente structure. L'estimation du mouvement consiste en la minimisation de la fonctionnelle:

$$\sum_{p\in\Omega} \rho(I(p,t) - I(p-dp, t-1)),$$

avec Ω le support d'estimation, ρ la métrique d'erreur (par exemple, $\rho(r) = r^2$). $I(p,t)$ est la valeur de l'image I au point p et à l'instant t, et dp le champ de mouvement dense. Le champ de mouvement dense est exprimé en fonction du mouvement des nœuds du maillage:

$$dp = \sum_i w_i(p) dp_i$$

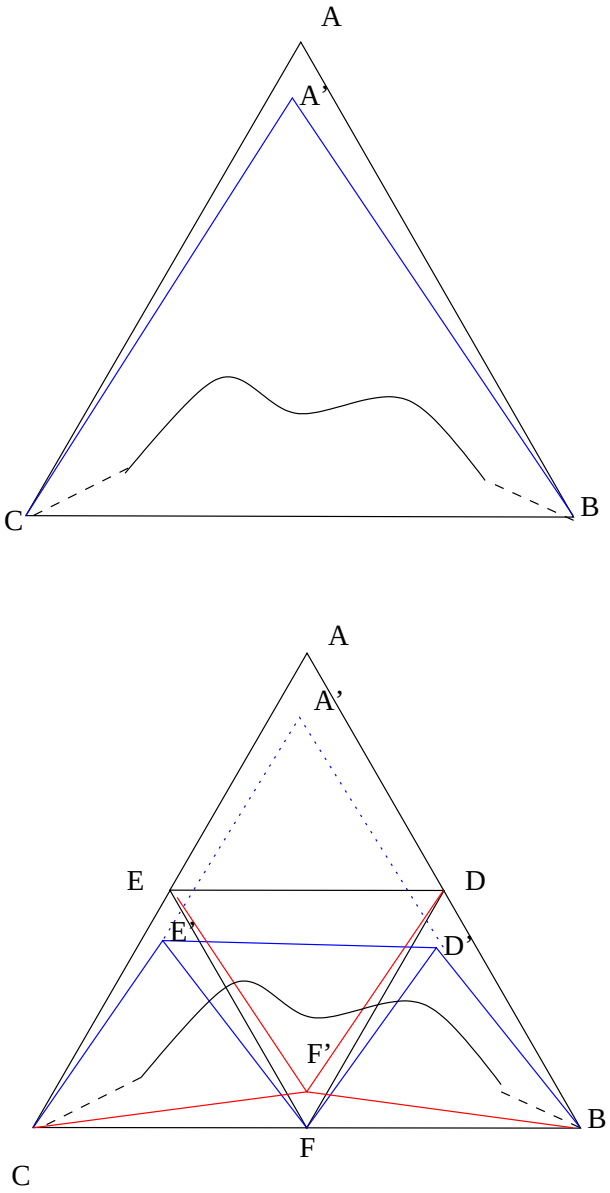


FIG. 6.14 – *Remontée en hiérarchie: disparition de la discontinuité*

où $w_i(p)$ représente les coordonnées barycentriques de p par rapport aux nœuds i de la maille à laquelle le point appartient et dp_i représente le déplacement associé au nœud i. Le déplacement dp en un point est obtenu par interpolation Lagrangienne. La minimisation est effectuée sur les nœuds du maillage, c'est-à-dire sur les dp_i des nœuds de t-1 à t. L'énergie est minimisée par une descente de gradient (type Gauss-Seidel), de manière itérative.
Lors de l'estimation du mouvement, pour propager la valeur des nœuds d'un niveau grossier vers un niveau fin, une technique multi-grille est utilisée. Cette technique nécessite la connaissance des liens de parenté entre des nœuds de niveaux différents dans la hiérarchie.
Ces relations de parenté ont été conservées avec la nouvelle structure de maillage. On a toujours les deux cas de parenté suivants: un nœud est fils direct d'un nœud de niveau inférieur ou un nœud est milieu d'un arc de niveau inférieur (on considère que ce nœud est alors issu de deux nœuds du niveau inférieur). Avec la nouvelle structure, on ajoute une relation de parenté: un nœud est défini à partir d'un triangle de niveau inférieur, il est alors issu de trois nœuds du niveau inférieur. La figure 6.15 illustre ces trois cas de parenté dans la hiérarchie.
La propagation des valeurs d'un niveau grossier vers un niveau fin utilise la pondération suivante selon les liens de parenté du nœud courant:

- si le nœud est fils direct d'un nœud du niveau inférieur, le nœud courant prend directement sa valeur
- sinon la pondération est calculée à partir de tous les nœuds des triangles auxquels le nœud courant peut appartenir

Dans le cas simple où le nœud courant est issu d'un arc appartenant à seulement deux triangles, la pondération du nœud courant utilise les poids barycentriques du nœud calculés par rapport aux quatre nœuds en jeu.
Avec la nouvelle structure de maillage non-manifold, un arc peut appartenir à plus de deux triangles. Dans ce cas, la pondération utilise tous les triangles auxquels le nœud courant est susceptible d'appartenir. La figure 6.16 illustre un tel cas. Sur la figure, le nœud E est issu de l'arc (A_1C_1), cet arc est partagé par les mailles $A_1B_1C_1$, $A_1C_1D_1$ et $A_1B'_1C_1$. La pondération utilisée pour le nœud E est une combinaison linéaire des poids barycentriques associés à ce nœud par rapport aux cinq nœuds: A_1, B_1, B'_1, C_1 et D_1.

Conclusion

Dans ce chapitre, nous avons proposé une nouvelle représentation du mouvement en prenant en compte sa nature discontinue. A l'aide d'une représentation continue par maillage et de la notion de ligne de rupture introduite dans [Marquant 00], nous avons proposé une structure de maillage non-manifold permettant de représenter le mouvement même dans les zones d'occlusions.
Le chapitre suivant présente les résultats obtenus en utilisant cette nouvelle structure de maillage pour l'estimation du mouvement d'une séquence vidéo.

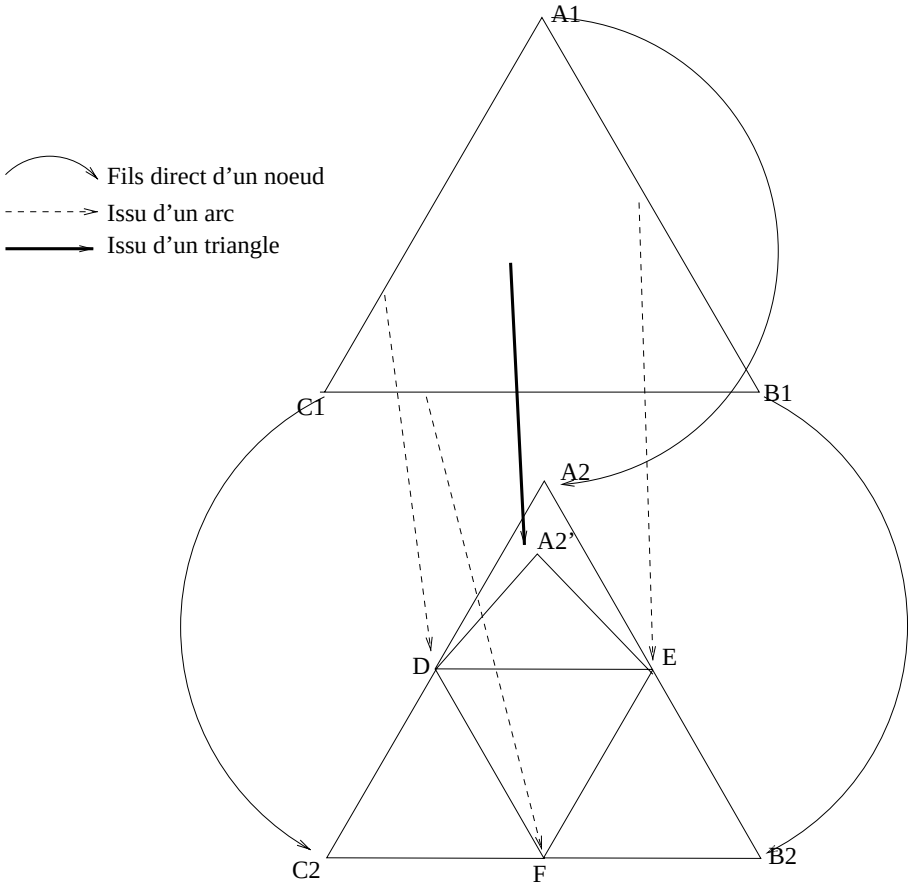


FIG. 6.15 – *Relation de parenté dans la hiérarchie*

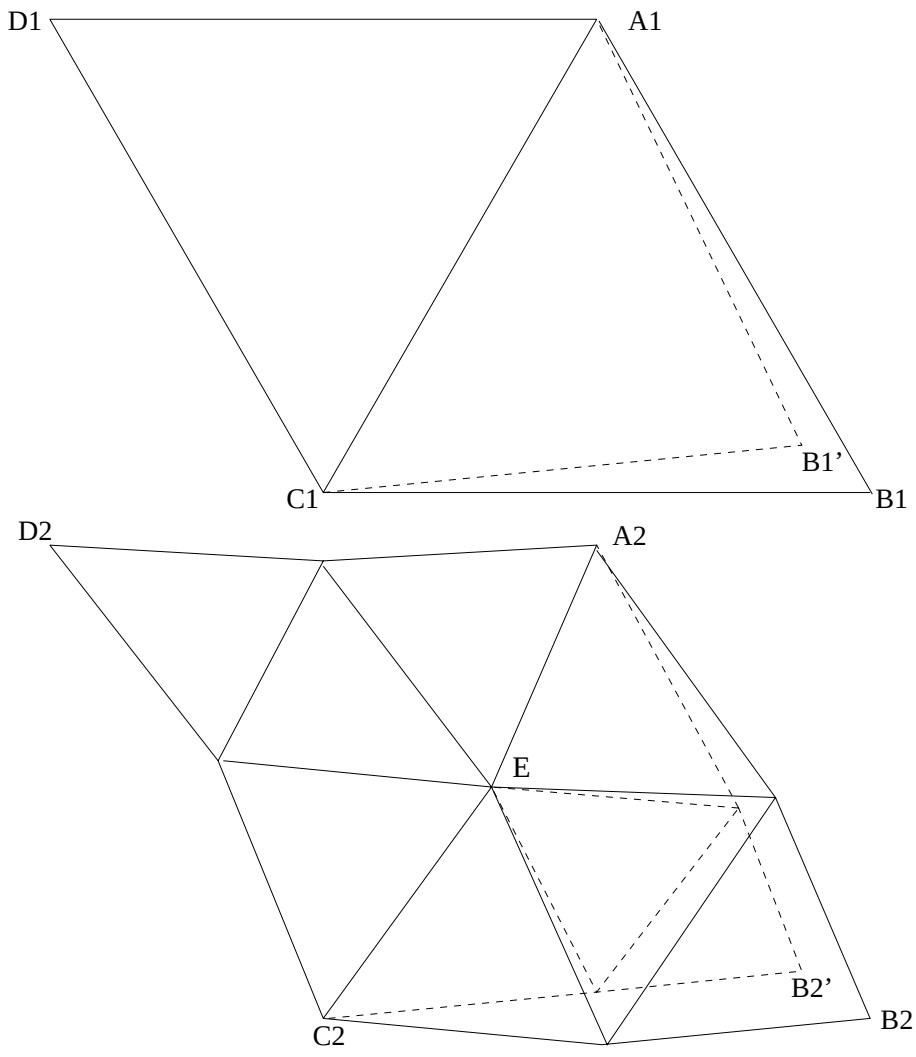


FIG. 6.16 – *Maillage non-manifold*

Chapitre 7

Gestion des zones d'occlusions: résultats et perspectives

7.1 Amélioration de l'estimation du mouvement

7.1.1 Le processus d'estimation du mouvement

La figure 7.1 montre le processus d'estimation du mouvement et la création de la structure de maillage non-manifold.
Après une première estimation du mouvement effectuée sur un groupe d'images, les mailles dégénérées sont détectées. Si aucune maille dégénérée n'a été détectée, l'estimation se poursuit sur le groupe d'images suivant. Si des mailles dégénérées ont été détectées, la zone d'occlusion est détectée. Dans cette zone d'occlusion, la ligne de discontinuité est détectée et le remaillage du niveau fin est effectué.
Le remaillage est ensuite remonté dans les niveaux de hiérarchie jusqu'à ce que la ligne de discontinuité disparaisse. L'estimation du mouvement est alors effectuée sur le groupe d'images avec la nouvelle structure de maillage.

L'estimation du mouvement effectue un suivi long terme de la texture. Le mouvement est estimé entre la première image du groupe et la nième image. Une image de texture dynamique peut alors être créée sur la grille d'échantillonnage de la première image représentant les variations de luminance de la scène dans la séquence vidéo.
La construction de la texture à l'instant t se fait par projection de l'image à l'instant t sur la grille d'échantillonnage de la première image. La définition d'une image de texture dynamique construite sur une grille d'échantillonnage unique permettra alors d'insérer aisément la structure de maillage non-manifold pour la représentation du mouvement dans le schéma de codage scalable présenté dans les chapitres précédents.
Cependant, dans le cas de découvrements de texture dans l'image à l'instant t, la texture construite par projection sur la grille de la première image ne pourra pas prédire la zone découverte à l'instant t (figure 7.2). Dans le cas de l'utilisation de maillage manifold, ceci se traduisait par un étalement des textures dans la zone de découvrement. Avec les maillages non-manifold, ceci se traduit pas une zone détectée comme non prédictible.

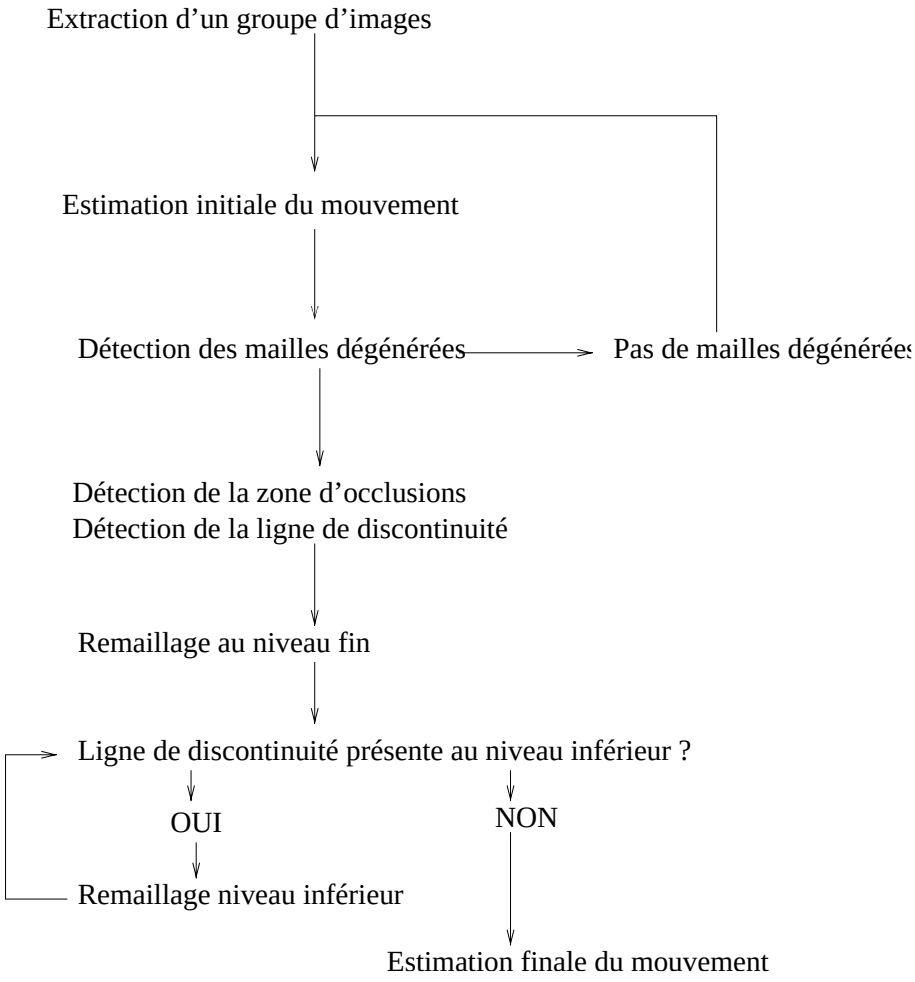


FIG. 7.1 – *Le processus d'estimation du mouvement*

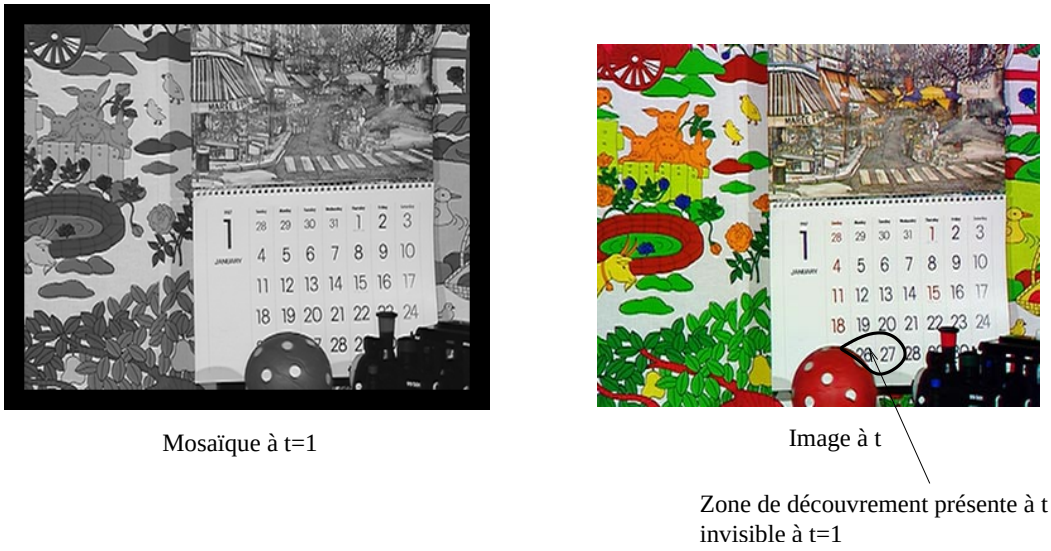


FIG. 7.2 – *Représentation de la texture sur une grille unique et découvrement à l'instant t*

Ces deux effets sont illustrés sur la figure 7.3.
Il s'agit alors de trouver une représentation de la texture qui permette à la fois de construire la mosaïque de la scène sur une grille unique et de sauvegarder l'information de texture découverte. Par simplicité, nous avons choisi d'utiliser deux images de texture définies sur la grille de la première image. La deuxième image de texture permet de sauvegarder l'information découverte à l'instant t qui n'est pas visible sur la première image. Ceci nous permet de pouvoir prédire les zones de découvrement des images à l'instant t et de vérifier que l'utilisation de maillages non-manifold permet d'améliorer l'estimation du mouvement et la prédiction des images dans les zones de découvrements.
Pour une application au codage vidéo, il faudra alors trouver une représentation plus compacte de la texture prenant en compte les zones de découvrements et définies sur une grille unique. Une solution est proposée en perspectives dans la section suivante.

7.1.2 Séquence Mobile And Calendar

Cette sous-section présente les résultats obtenus pour la séquence Mobile And Calendar. Le mouvement du ballon devant le calendrier découvre une partie de ce calendrier et crée une zone d'occlusions dans laquelle l'estimateur de mouvement par maillages manifold crée des étirements de maille comme illustrés par la figure 7.4.
Nous présentons ici l'amélioration de l'estimation du mouvement apportés par l'utilisation de la nouvelle structure de maillage non-manifold en comparaison avec la structure classique manifold. L'estimation du mouvement a été effectuée sur des groupes de 8 images. Le mouvement a été estimé entre la première image du groupe traité et l'image

FIG. 7.3 – *Images prédites par la texture*

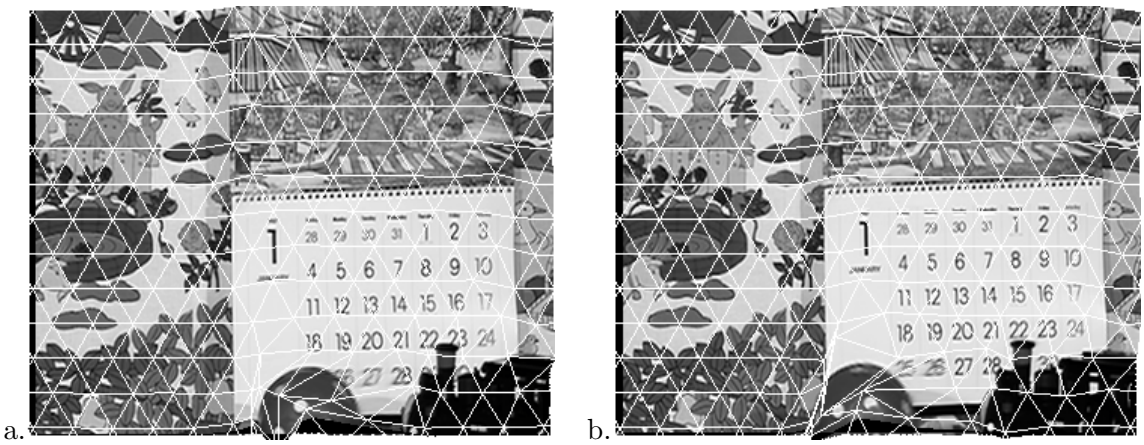
courante à l'instant t.
La ligne de discontinuité a été detectée à l'aide de masques de segmentation créés manuellement dans la zone d'occlusion.
La figure 7.5 montre l'estimation du mouvement sur le GOF des images 8 à 16. Le maillage nommé en avant-plan (figure7.5-a,c,e,g) correspond aux mailles issues du remaillage de la zone d'occlusion pour la région intérieure à la ligne de discontinuité. La zone active des mailles est en rouge, elle correspond à la zone visible des mailles. Les mailles roses sont inactives, elle ne participe pas à la reconstruction. Le maillage nommé arrière-plan (figure7.5-b,d,f,h) correspond aux mailles issues du remaillage de la zone d'occlusion pour la région extérieure à la ligne de discontinuité. Ces figures montrent également les mailles inchangées par le remaillage (en-dehors de la zone d'occlusion). Les mailles actives (pour la reconstruction de l'image) sont en bleu, les mailles inactives en bleu clair. Sur les deux côtés (avant-plan et arrière-plan), les mailles en vert sont les points d'attache des deux remaillages au maillage initial. Le contour est montré en turquoise.
On remarque sur les images du maillage en arrière-plan (7.5-f,h) que les mailles du calendrier dans la zone découverte ne sont plus déformées comme elles l'étaient avec l'estimation du mouvement par maillage manifold (figure 7.4-b). De plus, les mailles actives reconstruisant le ballon (7.5-e,g) sont également moins déformées. Ceci montre l'indépendance locale des deux maillages dans la zone d'occlusion. Ces deux maillages ne font qu'un en-dehors de la zone d'occlusion et se rejoignent également dans la hiérarchie lorsque la ligne de discontinuité disparaît.

Une fois le mouvement estimé, nous avons effectué l'étape d'analyse du codeur vidéo par analyse-synthèse afin de vérifier la qualité de reconstruction des images. Nous avons projeté les images de la séquence vidéo sur une grille de référence (positionnée

sur la première image du groupe traité) La texture dynamique ainsi construite a été complétée par la technique de padding présentée dans la première partie du manuscrit. Puis, la séquence vidéo a été reconstruite par reprojection de la texture sur sa grille d'échantillonnage initiale.
Avec une structure de maillage régulier, l'aller-retour de l'analyse-synthèse crée sur certaines images des artéfacts visuels intrinsèques au schéma analyse-synthèse. L'utilisation du schéma analyse-synthèse dans un schéma de codage implique que ces défauts seront présents quel que soit le débit autorisé au décodage. Ces défauts sont dus à la dégénération des mailles dans les zones où l'estimation du mouvement a échoué.
L'utilisation d'une nouvelle structure de maillage non-manifold permet de résoudre ce problème et d'estimer le mouvement même dans les zones de découvrement. L'aller-retour de la phase d'analyse ne doit alors plus créer ces artefacts visuels dus au découvrement, ou au moins les diminuer.
Le problème de l'utilisation de maillages non-manifold lors du plaquage sur une grille de référence t_{ref} (ici la première image du GOF) est que les zones découvertes à t ne sont pas présentes à t_{ref}. La texture découverte est alors perdue lors du plaquage sur t_{ref}. Ceci se traduit par des zones non prédictibles par t_{ref} lors du replaquage de la texture de t_{ref} sur t. Pour résoudre ce problème, nous utilisons deux images pour construire la texture sur t_{ref}. Ainsi, la texture découverte est sauvegardée et les deux images permettent de prédire l'image t à partir de la texture t_{ref}.
La figure 7.6 montre les images de fin de GOF reconstruite à partir de la projection sur la première image du GOF. On observe que les images reconstruites avec le maillage non-manifold présentent nettement moins d'artéfacts visuels que celles reconstruites avec le maillage manifold. La figure 7.7 montre un zoom de ces images dans la zone d'occlusion. On observe mieux l'amélioration apportée par la structure de maillage non-manifold. Sur les images 7.7-b et d les chiffres du calendrier sont mieux définis que sur les images 7.7-a et c. De même pour l'image 7.7-f où ici l'amélioration porte également sur une zone en-dehors de la région d'occlusion, les numéros 29 et 30 du calendrier sont plus lisibles avec le maillage non-manifold qu'avec le maillage manifold. L'indépendance locale des remaillages dans la zone d'occlusion permet d'améliorer l'estimation du mouvement même en-dehors de la zone d'occlusion.

7.1.3 Séquence Erik

Cette sous-section présente les résultats d'estimation du mouvement pour la séquence Erik. La figure 7.8 montre l'estimation du mouvement à l'aide d'un maillage manifold. L'estimation du mouvement est effectuée sur des groupes de 16 images. On observe un étirement des mailles dans la zone de découvrement du fond par la personne (image 26).
Les figures 7.9 et 7.10 montre l'estimation du mouvement par maillage non-manifold. La ligne de discontinuité a été extraite à partir des masques de segmentation disponibles pour cette séquence.
La figure 7.9 montre le maillage relatif à l'arrière-plan et la figure 7.10 montre le maillage relatif à l'avant-plan. Les mailles en vert sont les points d'ancrage des deux remaillages



a. Image 8 b. Image 16

FIG. 7.4 – *Séquence Mobile: estimation du mouvement avec maillage manifold: présence de mailles étirées dans les zones de découvrement*

au maillage initial. Dans la zone d'occlusion, de chaque côté de la ligne de discontinuité (en bleu turquoise sur les images) les mailles sont localement indépendantes. Les mailles d'arrière-plan (figure 7.9-image 26) de la zone d'occlusion sont moins étirées qu'avec la version manifold.
La figure 7.11 montre une image reconstruite après plaquage aller-retour sur la première image du groupe par un maillage manifold et un maillage non-manifold. L'image reconstruite par le maillage manifold montre une texture étirée derrière la personne dans la zone d'étirement des mailles, c'est-à-dire dans la zone de découvrement. Dans cette zone, la texture n'est pas étirée avec l'utilisation du maillage non-manifold. La figure 7.12 montre un zoom de la zone découverte des deux images reconstruites. On observe bien une amélioration apportée par le maillage non-manifold dans la zone découverte.

7.1.4 Séquence Flower

Cette sous-section présente les résultats d'estimation du mouvement pour la séquence Flower. La figure 7.13 montre l'estimation du mouvement par un maillage manifold. La zone découverte derrière l'arbre crée des étirements de mailles qui déforment la texture dans cette zone.
Les figures 7.14 et 7.15 montre l'estimation du mouvement à l'aide d'un maillage non-manifold. Sur l'image 216 des deux figures, la ligne en bleu turquoise montre la ligne de discontinuité détectée à partir de masques de segmentation obtenus manuellement. Le mouvement est estimé sur des groupes de huit images.
La figure 7.14 montre le remaillage de l'arrière-plan dans la zone d'occlusion, la figure 7.15 montre le remaillage de l'avant-plan. Les deux remaillages sont accrochés au maillage initial au niveau des mailles en vert.
La figure 7.16 montre les dernières images de GOF reconstruites après projection sur la

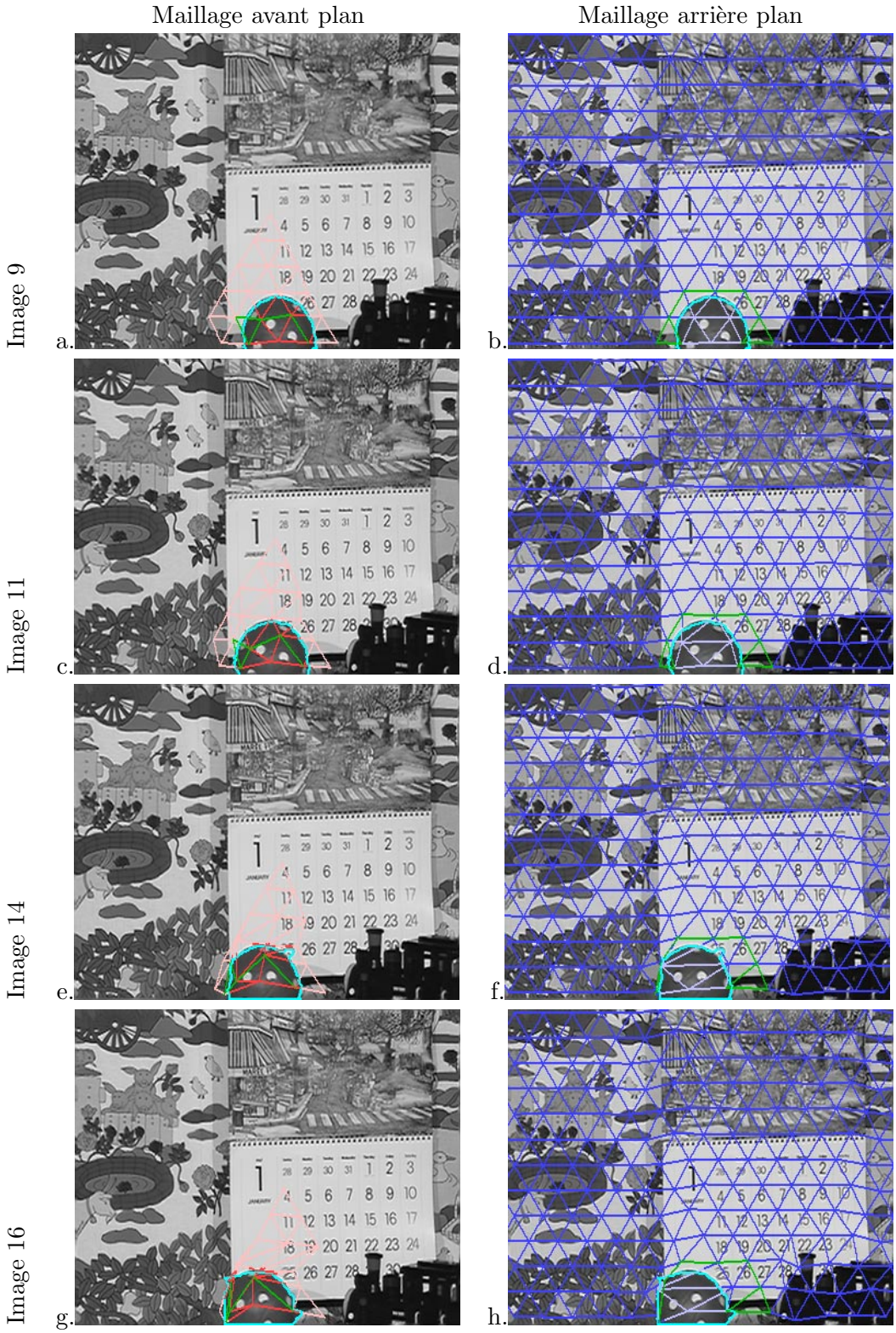


FIG. 7.5 – *Séquence Mobile: estimation du mouvement avec maillage non-manifold sur un GOF de huit images commençant à l'image 8 de la séquence*

FIG. 7.6 – *Séquence Mobile: images reconstruites après projection sur la première image du GOF*

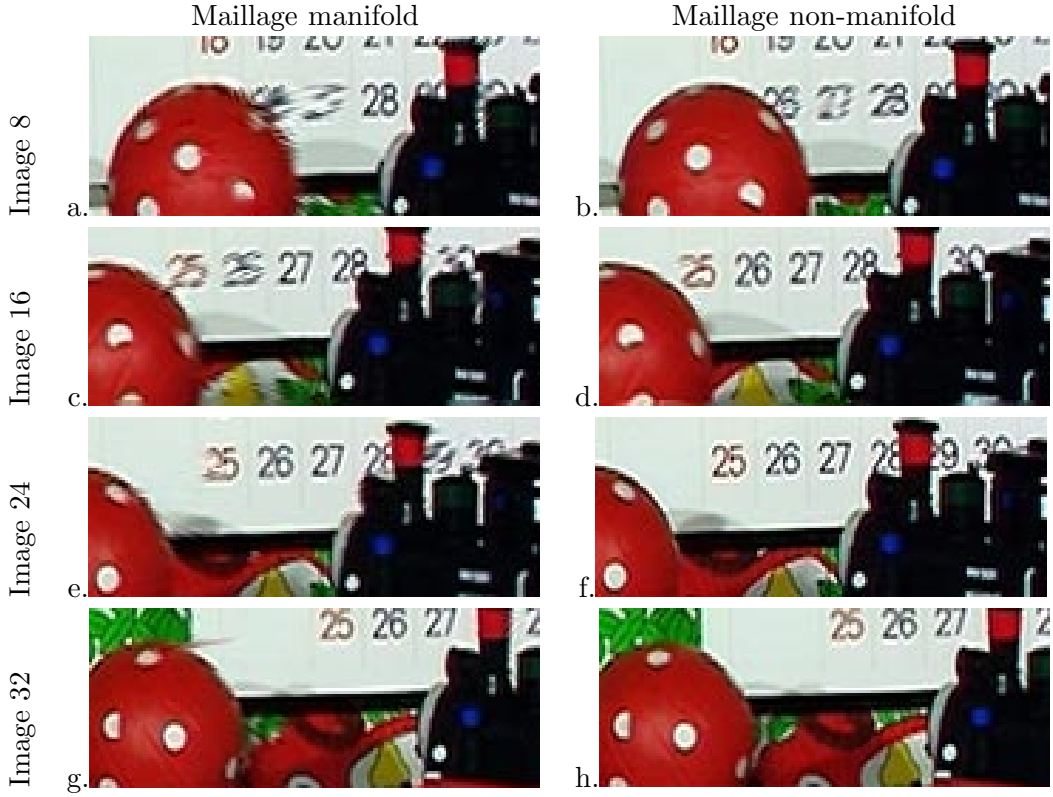


FIG. 7.7 – *Séquence Mobile: zoom sur les images reconstruites après projection sur la première image du GOF*

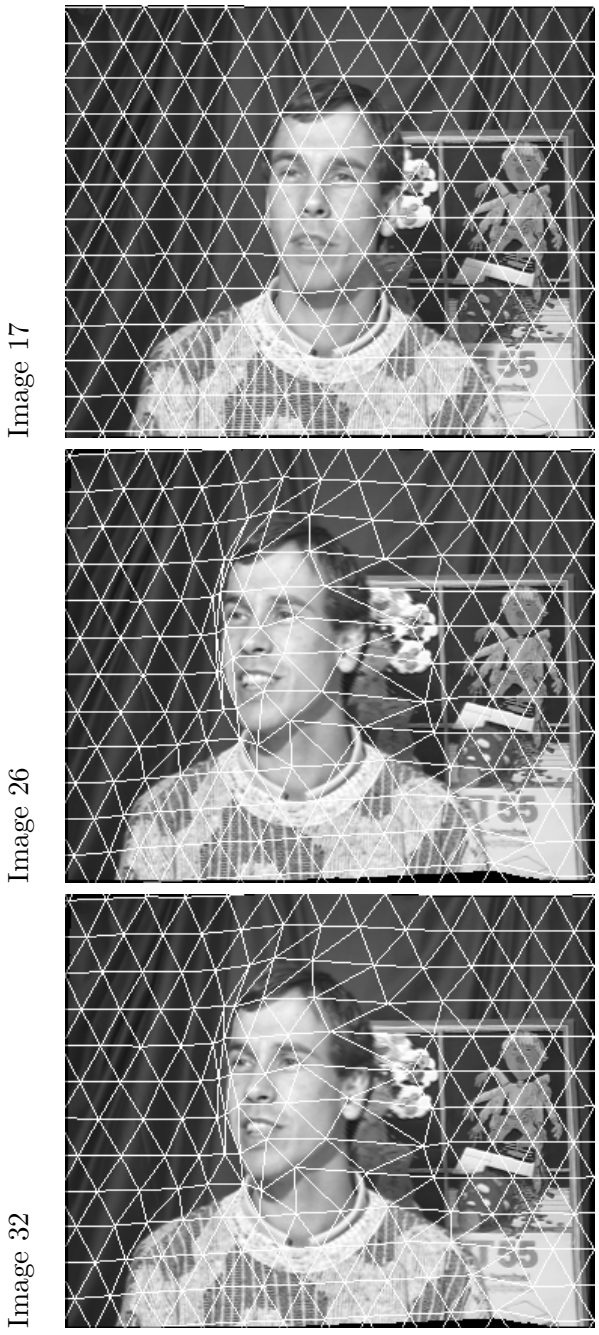


FIG. 7.8 – *Séquence Erik: estimation du mouvement par maillage manifold, sur un groupe de 16 images*

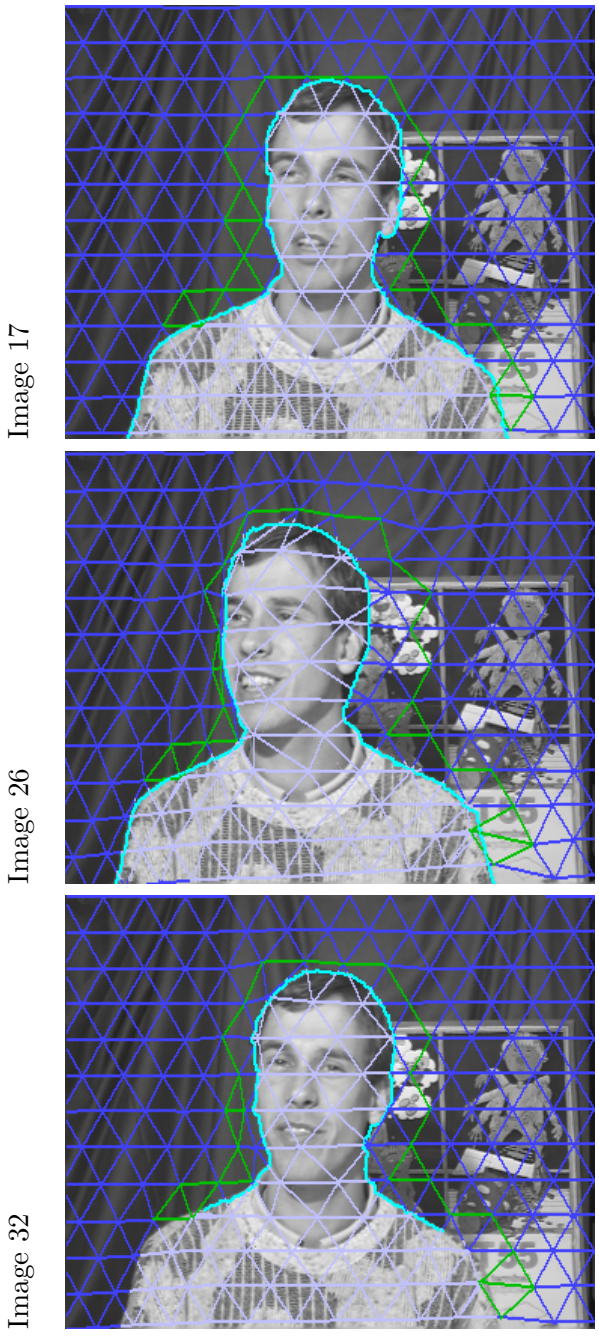


FIG. 7.9 – *Séquence Erik: estimation du mouvement par maillage non-manifold (maillage arrière-plan), sur un groupe de 16 images*

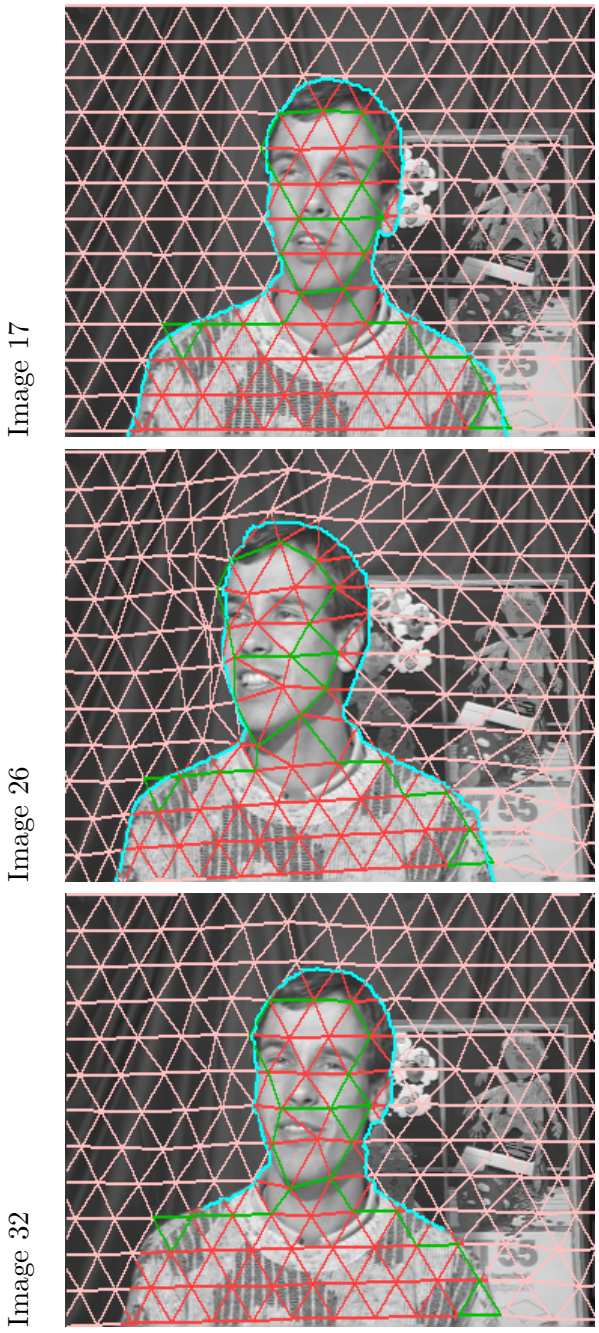


FIG. 7.10 – *Séquence Erik: estimation du mouvement par maillage non-manifold (maillage avant-plan), sur un groupe de 16 images*

Maillage manifold Maillage non-manifold

FIG. 7.11 – *Séquence Erik: reconstruction de l'image 26 par maillage manifold et non-manifold*

Maillage manifold Maillage non-manifold

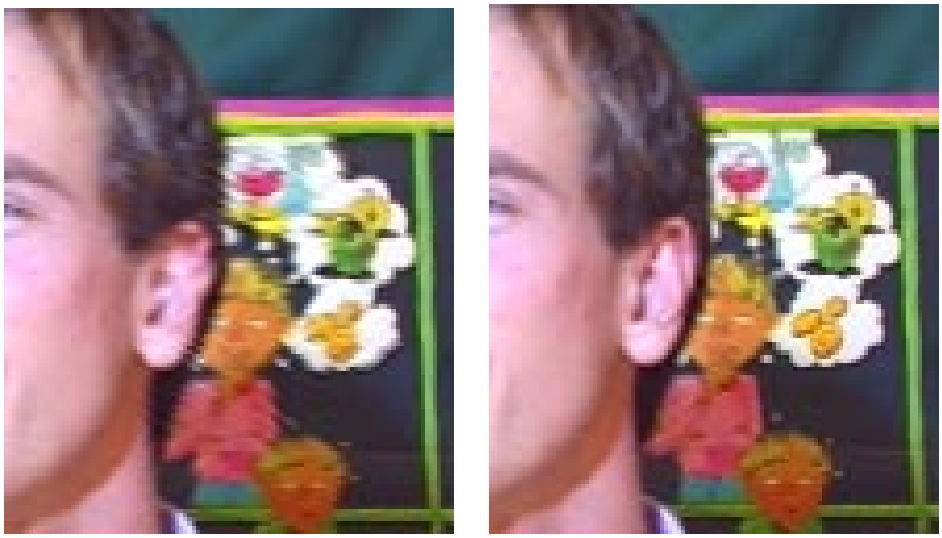


FIG. 7.12 – *Séquence Erik: zoom sur la zone de découvrement de l'image 26 reconstruite*

première image du GOF avec un maillage manifold et un maillage non-manifold. Dans les zones de découvrement, on observe que la texture est moins étirée avec un maillage non-manifold qu'avec un maillage manifold. Ceci est confirmé par la figure 7.17 qui montre un zoom des images reconstruites dans les zones de découvrement.

7.2 Application au codage de séquences vidéo

Afin d'améliorer le schéma de codage vidéo présenté dans la première partie de ce manuscrit, nous pouvons utiliser la nouvelle structure de maillage définie au chapitre précédent pour la représentation du mouvement dans ce codeur vidéo.
Il est alors nécessaire de coder les informations relatives à la zone d'occlusion afin de pouvoir reconstruire la même structure au décodage. Ces informations sont la liste des triangles dégénérés et la ligne de discontinuité. La ligne de discontinuité est représentée par une suite de points qui peut être codée de manière scalable à l'aide d'un codeur de contour comme celui défini dans [Chaumont 03].
Le mouvement est codé de la même manière que dans le précédent schéma de codage. Le seul problème reste celui du codage de la texture. Pour pouvoir coder la texture de la même manière que dans le codeur présenté, il est nécessaire de définir cette texture dans un plan. Or, dans les zones de découvrements et de recouvrements, la représentation image de la texture pose problème.

7.2.1 Le problème de la représentation de la texture

La nouvelle structure de maillages crée des chevauchements de mailles afin de prendre en compte les découvrements de texture apparaissant au cours de la séquence. La texture est définie sur une grille d'échantillonnage unique. Au cours du temps, la projection des images sur cette grille d'échantillonnage permet de construire une image de texture représentant les variations de luminance de la scène. Cependant, lors d'un recouvrement de texture, il serait intéressant de sauvegarder l'information de texture recouverte. De même lors d'un découvrement, il est nécessaire de sauvegarder l'information de texture découverte pour la reconstruction de l'image où ce découvrement est apparu. Ceci n'est pas possible avec une représentation 2D (image) de la texture, car il faudrait qu'en certains endroits, la texture puisse prendre deux valeurs différentes.
Nous devons donc trouver une représentation efficace de la texture, nous permettant de sauvegarder toutes les informations apprises au cours du temps et qui soit facile à coder.
Une première solution simple et rapide qui permette de représenter toute l'information de texture est d'utiliser une image multi-couches. Au lieu d'utiliser une seule image pour représenter la texture, nous proposons d'utiliser plusieurs images superposées qui permettent alors en un point (x, y) d'avoir plusieurs valeurs de texture possibles. Nous avons choisi cette solution pour sa simplicité et sa rapidité de mise en œuvre afin de valider l'utilisation d'une structure de maillage non-manifold dans le cadre d'une application au codage vidéo. Dans la sous-section suivante, nous présentons quelques résultats de codage utilisant cette structure.

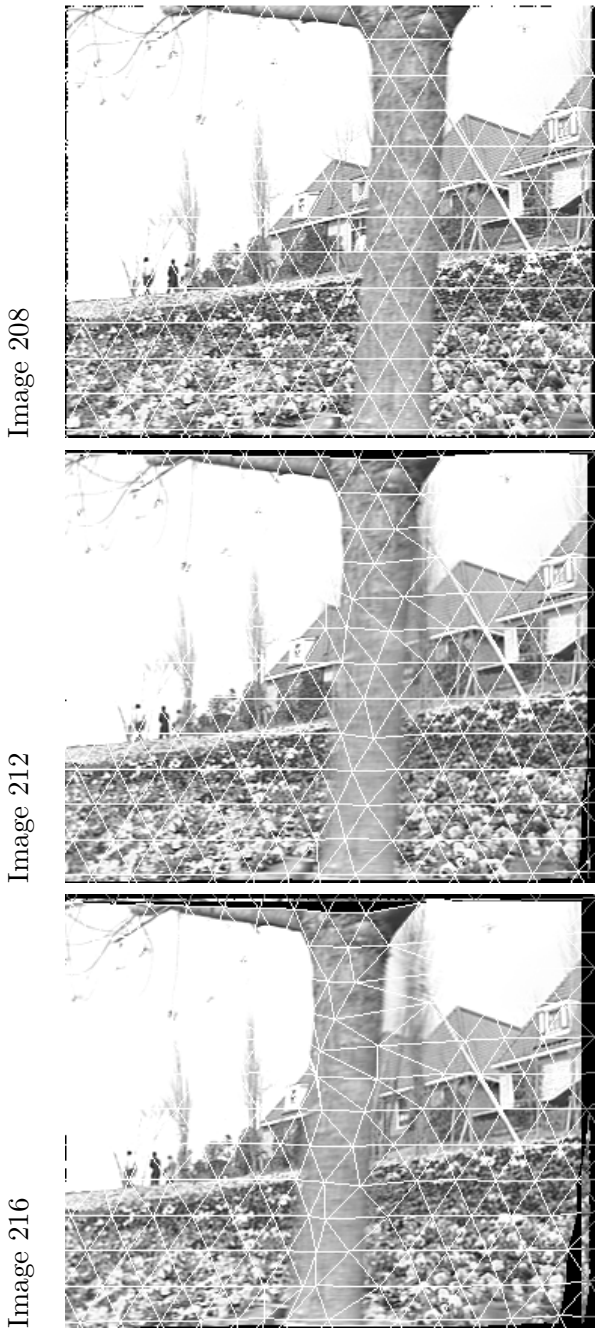


FIG. 7.13 – *Séquence Flower: estimation du mouvement par maillage manifold, sur un groupe de 8 images*

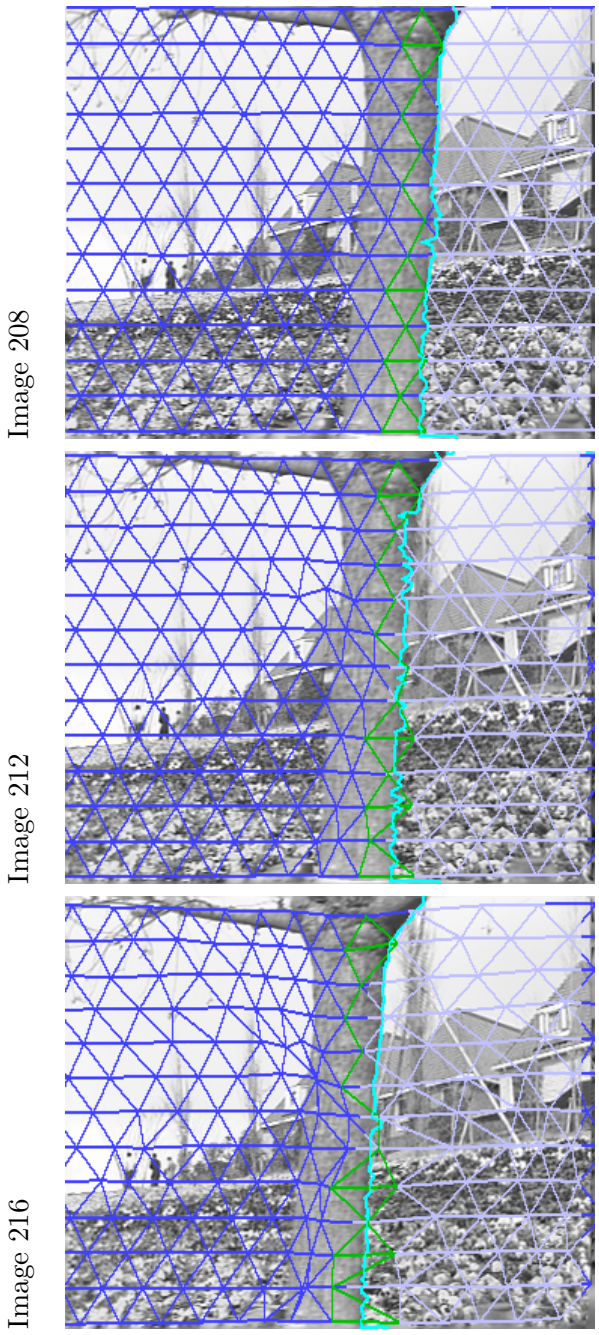


FIG. 7.14 – *Séquence Flower: estimation du mouvement par maillage non-manifold (maillage arrière-plan), sur un groupe de 8 images*

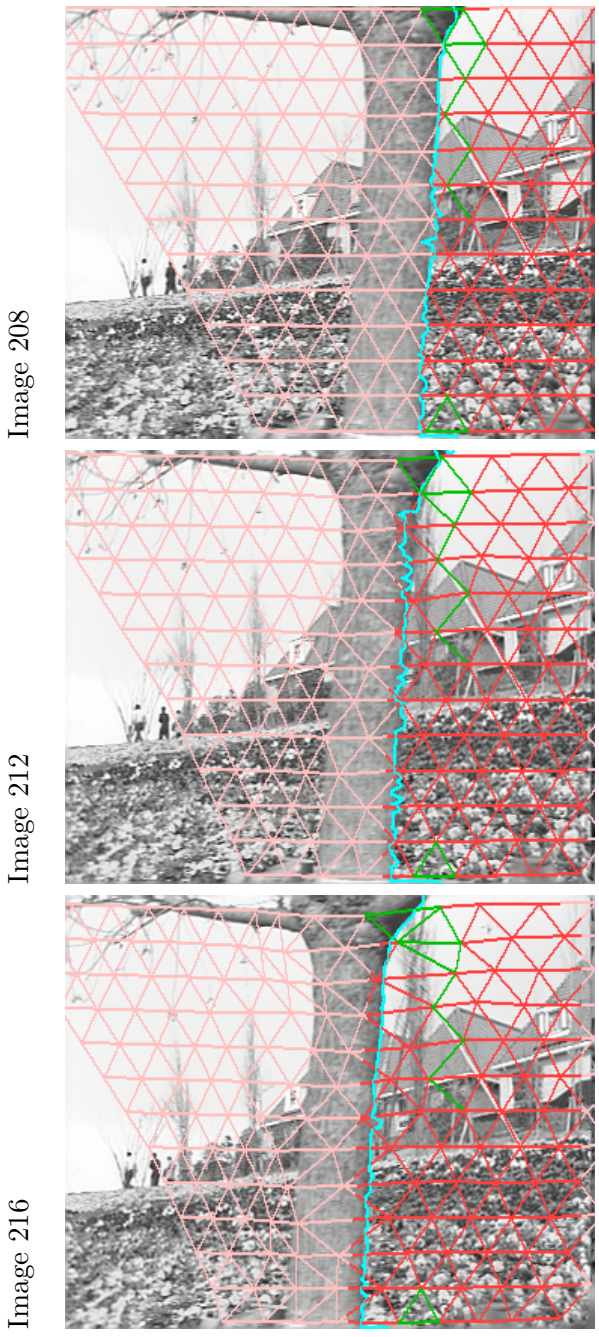


FIG. 7.15 – *Séquence Flower: estimation du mouvement par maillage non-manifold (maillage avant-plan), sur un groupe de 8 images*

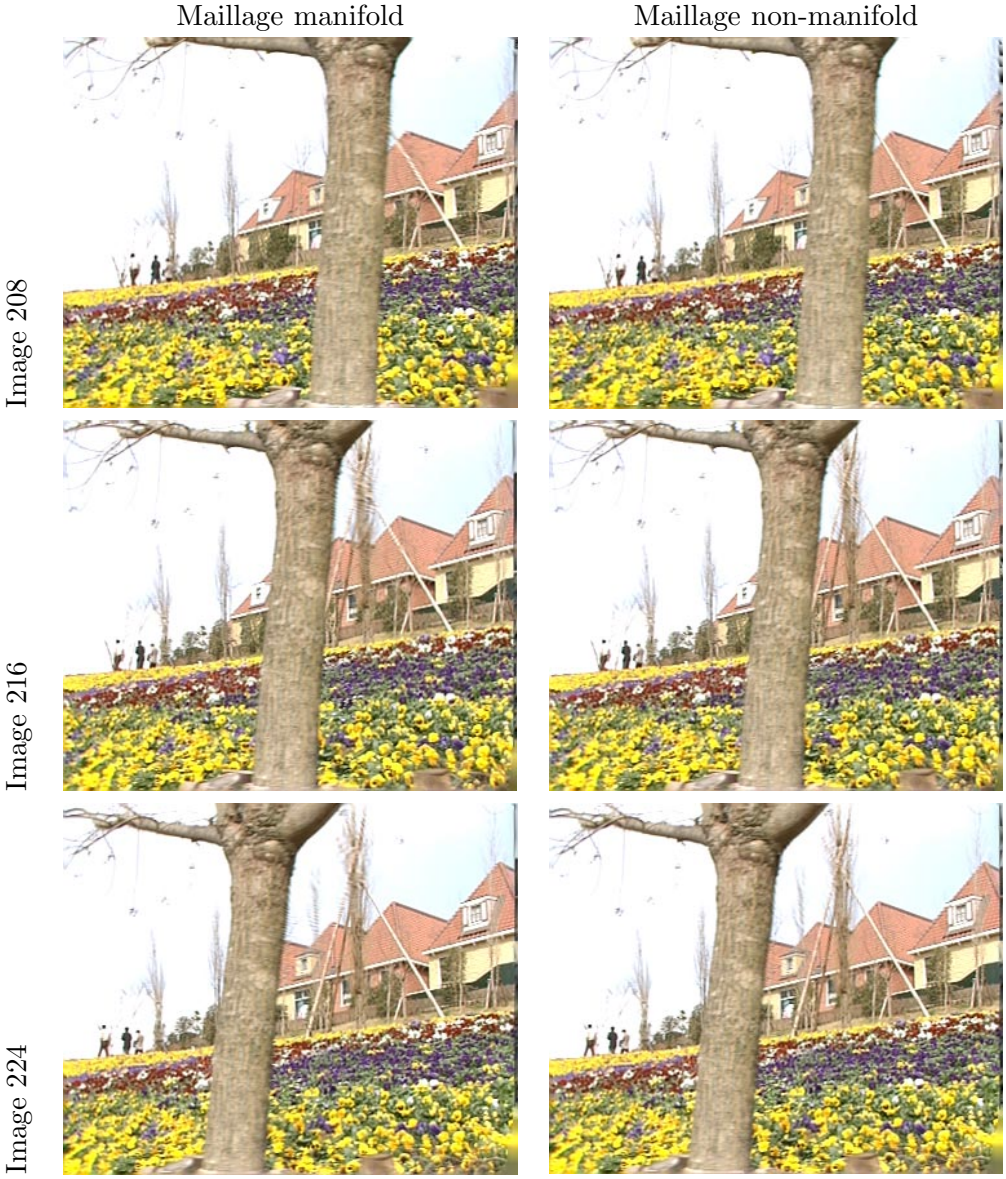


FIG. 7.16 – *Séquence Flower: images reconstruites (dernière image de GOF) après plaquage sur une image référence (première image du GOF)*

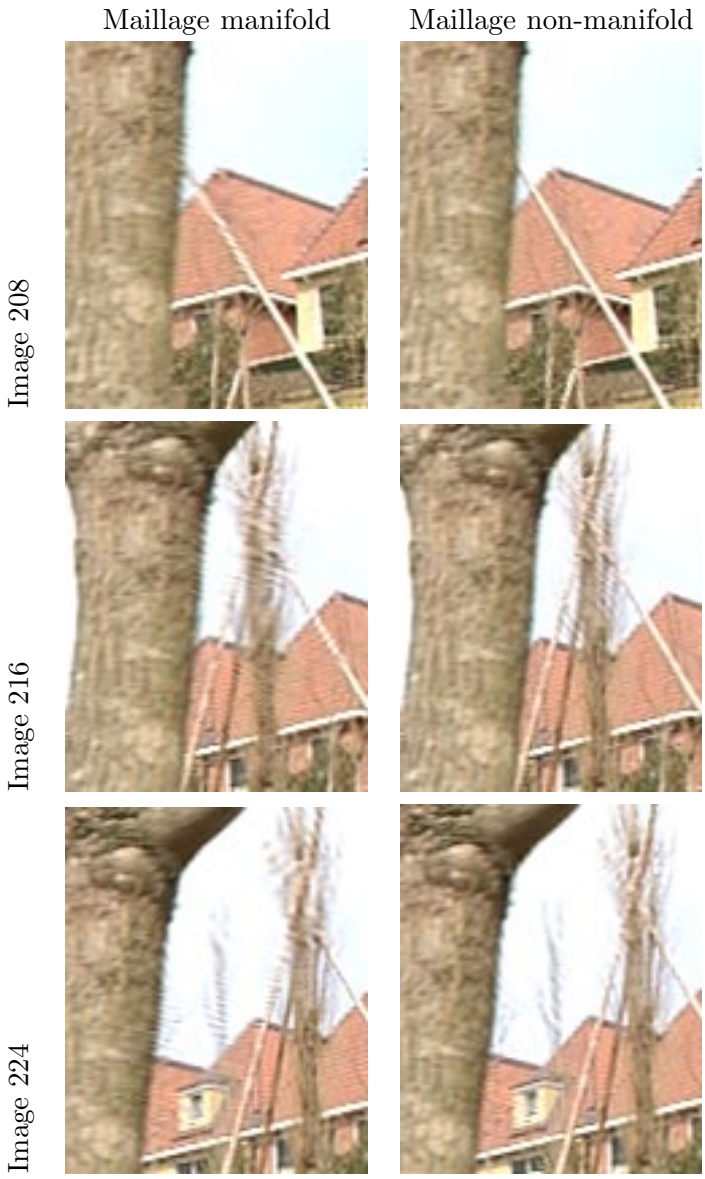


FIG. 7.17 – *Séquence Flower: zoom des images reconstruites*

Coût	Ancien codeur	Nouveau codeur
mouvement	8.1kb/s	7.9kb/s
ligne de discontinuité		1kb/s (127 points)

TAB. 7.1 – *Coût de codage du mouvement et de la ligne de discontinuité pour la séquence Mobile*

Cependant, la solution d'une image multi-couches pour la représentation de la texture peut être pénalisante pour une application de codage car la représentation de la texture est redondante. Plus loin, nous présentons en perspectives des solutions qui permettraient de représenter la texture de manière plus compacte.

7.2.2 Evaluation du coût de codage de la structure

Nous avons insérer la nouvelle structure de maillage dans le codeur vidéo par analyse-synthèse présenté dans les chapitres précédents. Les données à coder sont de trois sortes:

- le mouvement, représenté par le maillage. Nous devons codé les positions des nœuds à chaque instant. Le codage du mouvement est similaire à ce qui se faisait dans le schéma de codage présenté: un codage en deux couches. Une couche basse code les positions des nœuds dans la dernière image à l'aide d'un codage arithmétique. Les positions dans les images intermédiaires sont interpolées à partir des positions dans la première et la dernière image. La couche haute code le résidu d'interpolation des images intermédiaires à l'aide d'une transformation spatio-temporelle, suivie d'un codage par plans de bit avec optimisation débit/distorsion. La structure est générée de la même façon au codage et au décodage, le maillage de la première image n'a pas besoin d'être codé.
- la texture, représentée par deux images superposées. Les deux images de texture permettent de prendre en compte les zones où deux valeurs de textures sont nécessaires. Les images de texture sont obtenues par projection des images sur la première image du GOF. Les deux images sont ensuite codées en deux couches. La couche basse code la texture relative à la première et à la dernière image du GOF, les textures des images intermédiaires sont interpolées à partir des textures des images extrêmes. La couche haute code le résidu d'interpolation des textures intermédiaires.
- la ligne de discontinuité, représentée par une suite de points. La ligne de discontinuité peut être codée à l'aide d'un codeur de contour scalable comme celui présenté dans [Chaumont 03]. La ligne de discontinuité permet de reconstruire les masques de visibilité de chaque maille au décodage.

Les tableaux 7.1, 7.2 et 7.3 donnent les coût de codage des trois informations pour les séquences Mobile And Calendar, Erik et Flower, toutes au format CIF, 30Hz.
Le coût de codage de la ligne de discontinuité est une estimation basée sur les performances du codeur de contour présenté dans [Chaumont 03]. Le codage de la ligne de discontinuité implique un surcoût par rapport à l'ancienne version du codeur sans lignes

FIG. 7.18 – *Séquence Mobile: résultats de codage à 512kb/s (codage par GOFs de huit images)*

Coût	Ancien codeur	Nouveau codeur
mouvement	18.68kb/s	18.20kb/s
ligne de discontinuité		3kb/s (580 points)

TAB. 7.2 – *Coût de codage du mouvement et de la ligne de discontinuité pour la séquence Erik*

Coût	Ancien codeur	Nouveau codeur
mouvement	20.70kb/s	17kb/s
ligne de discontinuité		2kb/s (306 points)

TAB. 7.3 – *Coût de codage du mouvement et de la ligne de discontinuité pour la séquence Flower*

de discontinuité. Cependant, on remarque que ce surcoût est assez faible, de l'ordre de 1 à 3 kb/s.
Au niveau du coût de mouvement, les débits ont été obtenus par rapport à une distorsion donnée (d=0.1). La distorsion utilisée est l'EQM moyenne calculée sur les déplacements des nœuds. On remarque que le coût du mouvement pour la nouvelle structure de maillages est légèrement inférieur à celui du maillage classique. Ceci s'explique sans doute par le fait que l'estimation du mouvement ayant été améliorée, les déplacements des nœuds dans la zone d'occlusion sont plus cohérents avec l'utilisation d'une ligne de discontinuité car les nœuds ont pu bouger plus librement. Le mouvement des nœuds est plus proche du mouvement réel des objets. Il est alors plus continu et donc plus facile à coder.
Au niveau du codage de la texture, les coûts de codage pour une distorsion donnée ne sont pas vraiment comparables avec l'ancienne version du codeur analyse-synthèse. La représentation de la texture utilisée est trop redondante pour pouvoir donner des résultats performants. La figure 7.18 présente les images reconstruites par le codeur analyse-synthèse classique et le codeur avec maillage et ligne de discontinuité. La texture a été décodée à 512kb/s pour les deux codeurs. Les résultats sont assez encourageants. La qualité visuelle de la texture fournit par le codeur avec maillage et ligne de discontinuité n'est pas aussi bonne que celle du codeur classique. Ceci est dû à la représentation redondante de la texture. Cependant, on remarque une amélioration de la qualité dans les zones d'occlusion avec le maillage avec ligne de discontinuité par rapport au codeur classique. Il s'agit maintenant de trouver une représentation plus compacte de la texture afin d'améliorer ces premiers résultats de codage. Quelques perspectives sont données dans la sous-section suivante.

7.2.3 La paramétrisation

La texture peut être représentée en 3D à l'aide d'une surface à trous. Les trous sont dus aux déchirures effectuées dans le maillage au niveau de la ligne de discontinuité. De chaque côté de la déchirure, le remaillage a introduit des languettes de texture qui chevauchent la texture initiale. Cette surface est représentée sur la figure 7.19.

Une paramétisation de l'espace 3D de la texture au plan ([Desbrun 02]) permettrait de représenter notre texture sur une image 2D en conservant toutes les informations. Trouver une paramétrisation, c'est trouver une fonction Ψ bijective telle que:

$$\begin{aligned} \Psi : \quad & D \rightarrow P \\ & d_i \rightarrow p_i \end{aligned}$$

où $d_i = (x_i, y_i, z_i)$ est un point 3D de la surface et $p_i = (u_i, v_i)$ est un point 2D du plan. La paramétrisation associe à chaque nœud d'un maillage 3D une position dans le plan. Elle est très utilisée dans le domaine de la synthèse 3D. Dans notre approche, la projection de la texture sur le maillage d'une image t pour reconstruire cette image est une forme de paramétrisation.
La paramétrisation d'une surface n'est pas unique. Elle utilise en général un certain nombre de contraintes fixées selon l'application. Ces contraintes peuvent être entre autre la conservation des angles des triangles ou la conservation des aires. La paramétrisation minimise la distorsion définie par rapport aux contraintes fixées.
Dans notre cas, nous devons trouver les positions des nœuds d'un maillage 2D reconstruisant au mieux la surface 3D à partir des positions des nœuds du maillage 3D de la surface. Le maillage 3D de la surface est en fait un maillage 2D non-manifold. Comme la surface est dynamique, l'optimisation de la position des nœuds du maillage est spatio-temporelle. On cherche une grille 2D unique pour toutes les images du groupe traité. Un exemple de paramétrisation est illustrée par les figures 7.20 et 7.21. On voit que la paramétrisation peut parfois conduire à des mailles déformées, c'est pourquoi une des contraintes dans le choix de la paramétrisation porte souvent sur la conservation de l'aspect des mailles.
La paramétrisation de la surface permet de représenter la texture sur un plan 2D, cette texture peut alors ensuite être codée par le codeur ondelettes t+2D présenté dans la première partie du manuscrit. Cependant, il est nécessaire de transmettre la paramétrisation afin de pouvoir reconstruire les images de la séquence au décodage.
La paramétrisation à coder est en fait un maillage 2D. Il suffit alors de coder les positions du maillage 2D. Ce maillage est unique pour chaque groupe d'images traité. Le surcoût de codage est alors assez faible.

7.2.4 Représentation maillée et ondelettes de seconde génération

Nous pouvons également envisager une représentation maillée de la surface 3D. A l'aide d'un remaillage de la surface, il est possible de générer un maillage intra permettant de reconstruire cette surface. Les nœuds du maillage portent alors une information de texture.
La décorrélation temporelle de la texture peut être effectuée au niveau surface: par prédictions, ou bien par transformation en ondelettes temporelles. Puis, les images d'erreurs ou les sous-bandes temporelles sont représentées à l'aide d'un maillage intra et codées par des ondelettes de seconde génération appliquées au maillage [Brangoulo 04].
Le codage de la surface texture ne nécessite plus de projection dans le plan, la représentation par maillage intra de la texture permet un codage au niveau surface. Il s'agit alors de

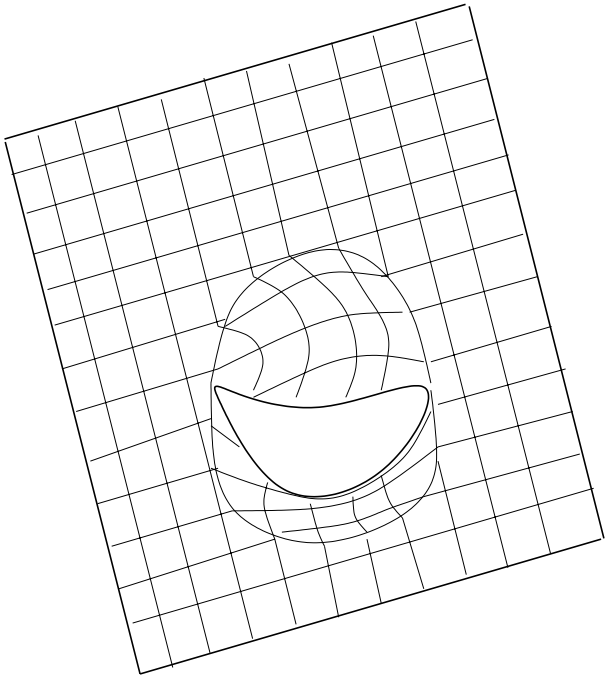


FIG. 7.19 – *Représentation de la texture par une surface*

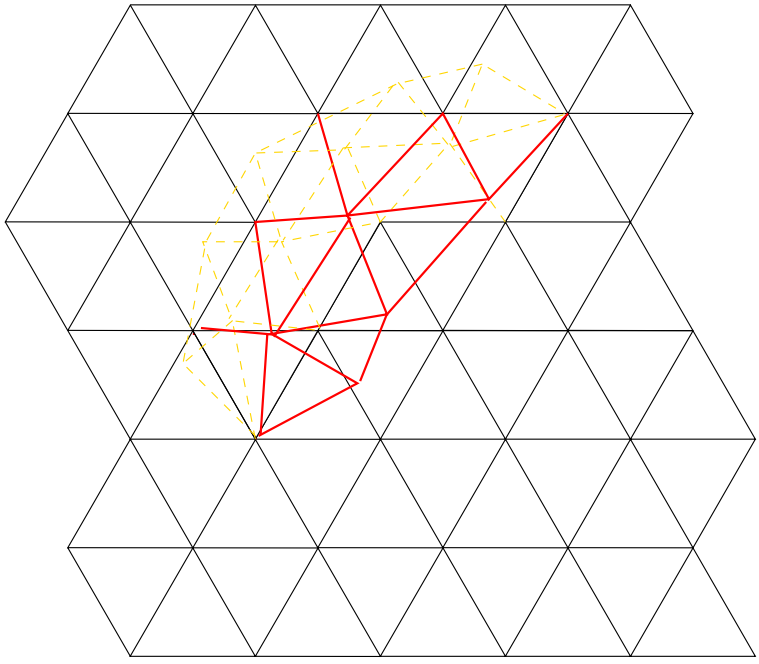


FIG. 7.20 – *Surface à paramétriser*

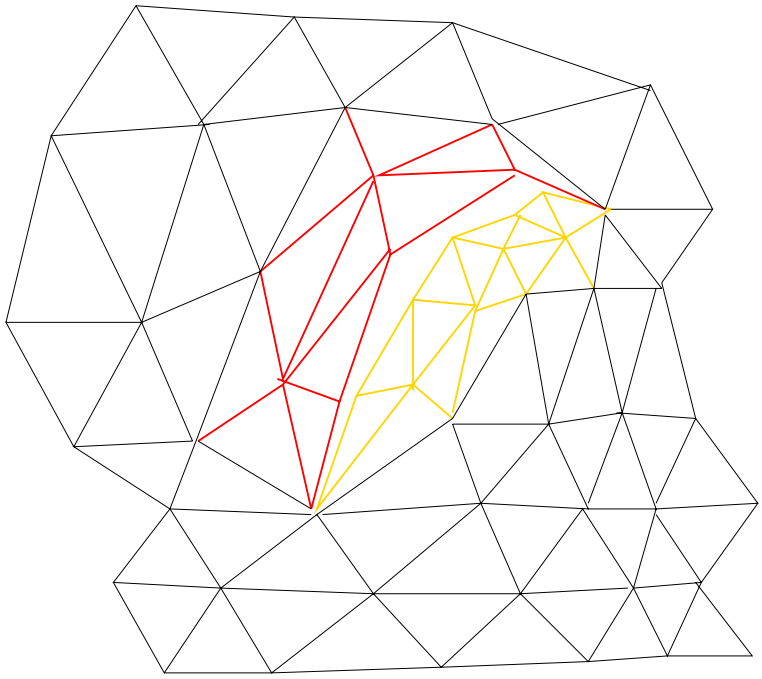


FIG. 7.21 – *Paramétrisation de la surface*

trouver des outils de codage qui s'appliquent à une structure de maillage et non plus à une structure image.
Dans le cadre de codage scalable, l'utilisation de maillage hiérarchique pour la représentation maillée permet de définir différents niveaux spatiaux ou en qualité de la texture.

7.3 Le problème des changements de résolution

7.3.1 Le problème des changements de résolution

Dans une séquence vidéo, les forts mouvements de zoom ou les mouvements de fortes amplitudes entrainent des changements importants de résolution de la texture. Ces mouvements se traduisent dans le maillage par un étirement ou une contraction des mailles. Le plaquage de la texture sur ces mailles entrainent un étalement ou une contraction de la texture. Ceci est illustré par la figure 7.22.
Le plaquage des textures sur des mailles déformées ne tient pas compte de ces changements de résolution et la texture à la résolution r est plaquée en tout pixel quelle que soit la résolution de ce pixel. Ceci influe sur la qualité de la prédiction des images.
Dans la suite de cette section, nous proposons quelques solutions afin de pallier ce problème des changements de résolution.

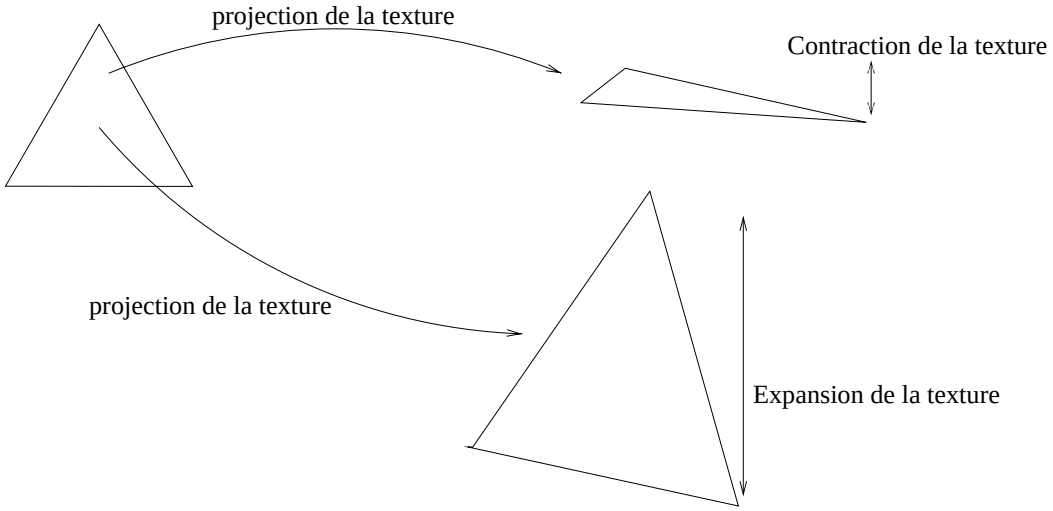


FIG. 7.22 – *Changements de résolution à l'intérieur d'une maille*

7.3.2 Solutions possibles

7.3.2.1 La résolution d'un pixel

La déformation des mailles entrainent des changements de résolution qui provoquent l'apparition de défauts visuels tels que l'aliasing ou un étalement des textures lors du plaquage de la texture sur une image. Ils influent également sur la qualité de la prédiction des images par compensation en mouvement.
Pour gérer ce problème de changements de résolution, nous proposons lors de la projection d'une texture sur la grille d'échantillonnage d'une image d'adapter la résolution de la texture à projeter à la résolution du pixel. Un pixel de l'image à reconstruire n'est alors plus défini seulement par sa position (x, y) dans le plan image mais aussi par sa résolution r.
La résolution d'un pixel peut être calculée par rapport à la déformation du triangle auquel il appartient. La déformation subie par les triangles du maillage est une déformation affine. Si un triangle $P_1P_2P_3$ se transforme en $Q_1Q_2Q_3$, le point P de $P_1P_2P_3$, avec $P = \sum_{i=1}^{3} w_i P_i$, se transforme en $Q = \sum_{i=1}^{3} w_i Q_i$. Pour calculer les changements de résolution en x et y d'un point P, on dérive Q par rapport à P, $\frac{\partial Q}{\partial P}$. La transformation affine est définie par:

$$Q = AP + B = \begin{bmatrix} a & b \\ d & e \end{bmatrix} P + \begin{bmatrix} c \\ f \end{bmatrix}$$

La dérivée $\frac{\partial Q}{\partial P}$ s'exprime alors $\frac{\partial Q}{\partial P} = A$. Les paramètres de la transformation affine de chaque triangle sont calculés à partir des coordonnées des nœuds de ce triangle. Une fois les paramètres affines calculés, on peut obtenir les résolutions en x et y de deux

manières ([Schilling 96, Williams 83, Gangnet 84]):

- $r_x = Sup(\frac{\partial x_Q}{\partial x}, \frac{\partial y_Q}{\partial x}) = Sup(a, d)$ et $r_y = Sup(\frac{\partial x_Q}{\partial y}, \frac{\partial y_Q}{\partial y}) = Sup(b, e)$
- $r_x = \sqrt{a^2 + d^2}$ et $r_y = \sqrt{b^2 + e^2}$

Les résolutions ainsi obtenues sont les résolutions des triangles déformés, pour obtenir les résolutions en chaque point de l'image, il nous faut calculer les résolutions aux nœuds de chaque triangle. Un noeud P appartenant à plusieurs triangles, sa résolution peut être calculée:

- soit par la moyenne des résolutions des triangles auxquels il appartient: $r_P = \frac{1}{N} \sum_{i=1}^{N} r_i$ avec N le nombre de triangles auxquels appartient P et r_i leur résolution respective.
- soit par une somme pondérée de ces résolutions: $r_P = \frac{1}{\sum_{i=1}^{N} A_i} \sum_{i=1}^{N} \frac{r_i}{A_i}$ avec A_i l'aire du triangle i.

Les résolutions en chaque point M sont alors interpolées par les poids associés aux nœuds: $r_M = \sum_{i=1}^{3} w_i r_i$.

7.3.2.2 Pyramide multirésolution: techniques existantes

L'adaptation de la résolution de la texture peut se faire au moment du plaquage en filtrant directement la texture à plaquer ou bien en pré-traitement. Dans ce dernier cas, la texture est alors préalablement filtrée à différentes résolutions et stockées dans une pyramide de type mip-map, lors du plaquage une interpolation bilinéaire entre les versions pré-filtrées de la texture permet d'adapter la résolution à la résolution du pixel. L'avantage du filtrage direct au moment du plaquage est de pouvoir filtrer dans la direction d'étirement de la maille [Heckbert 89, Feibush 80]. A cause du coût de stockage des textures filtrées, les techniques de pré-filtrage ne permettent de filtrer que suivant un nombre de directions et de résolutions restreint [Williams 83, Gangnet 84].
Les solutions proposées ci-dessus permettent de gérer le problème de changements de résolution dans le cas où ce changement intervient d'une résolution fine vers une résolution grossière. Dans le cas inverse, le problème est plus complexe car il s'agit alors d'affiner l'information de texture. Une solution à ce problème est donnée par l'utilisation de techniques de super-résolution.
La super-résolution permet de reconstruire une image super-résolue à partir de plusieurs vues d'une scène. Dans notre cas, une image super-résolue peut-être construite à partir des images de la séquence vidéo.
Une des techniques permettant de construire des images super-résolues est la technique par rétroprojection [Zomet 98]. Le problème consiste à minimiser une fonctionnelle portant sur la distorsion de reconstruction des images basses résolutions à partir de l'image super-résolue estimée. Cependant, les techniques de super-résolution sont très dépendantes de la qualité de l'estimation du mouvement entre les images de la vidéo et sont assez coûteuses en terme de complexité opératoire.

7.3.2.3 Pyramide multirésolution pour représenter la texture

Il s'agit ici de définir une pyramide de texture multirésolution dynamique et représentée par une grille unique d'échantillonnage. La pyramide de texture est dynamique car elle traduit les variations temporelles de la luminance au cours de la séquence vidéo et elle est construite par apprentissage au cours du temps.
Supposons que la pyramide de texture soit définie sur la grille d'échantillonnage de la première image de la séquence vidéo. Le mouvement est suivi à l'aide d'un maillage régulier positionné sur l'image à t=0. La texture à cet instant est à la résolution 1, les mailles ne sont pas déformées.
Le mouvement est estimé entre l'image à t=0 et l'image à t=t_k. Les mailles se déforment, la pyramide est complétée à l'aide de la projection de l'image à t_k sur la grille de l'image à t=0 en projetant chaque pixel dans la pyramide en fonction de sa résolution (figure 7.23).
Une fois, toutes les images de la séquence projetée dans la pyramide, celle-ci est complétée à l'aide d'une technique de remplissage spatio-temporel et à travers les résolutions (figure 7.24) comme celle présentée au chapitre 3, section 3.2.2.2.
Un parallèle entre la pyramide de décomposition ondelettes et la pyramide de texture multi-résolution peut être fait (figure 7.25). La pyramide multirésolution correspond aux basses fréquences successives de l'image de la pyramide à la plus grande résolution. Le remplissage spatio-temporel peut alors être effectué à l'aide de filtres d'interpolation et de décimation similaire aux filtres ondelettes. Son but est de trouver les sous-bandes hautes fréquences de chaque niveau à partir des sous-bandes basses fréquences données par la pyramide de texture.
L'analogie à la pyramide ondelettes est intéressant pour une application de codage de la pyramide de texture. En effet, la pyramide de texture peut alors être codée comme un ensemble de sous-bandes ondelettes.
De plus, dans le cadre de codage vidéo scalable, la représentation de la texture par pyramide multirésolution fournit une représentation naturellement scalable de la texture. La définition des niveaux de résolution permet de coder progressivement la texture et de ne décoder que l'information nécessaire au format de restitution. Enfin, la construction de la pyramide permet de générer des textures de plus haute résolution que la résolution d'origine et ainsi d'offrir une reconstruction très haute définition de la séquence.

L'utilisation d'une pyramide de texture multirésolution reste cohérente avec l'utilisation d'une représentation maillée de la texture comme présentée dans la section précédente. En effet, l'utilisation de maillage hiérarchique pour cette représentation maillée permet de définir la notion de texture multirésolution. A chaque niveau de hiérarchie de maillage correspond un niveau de résolution de la texture.

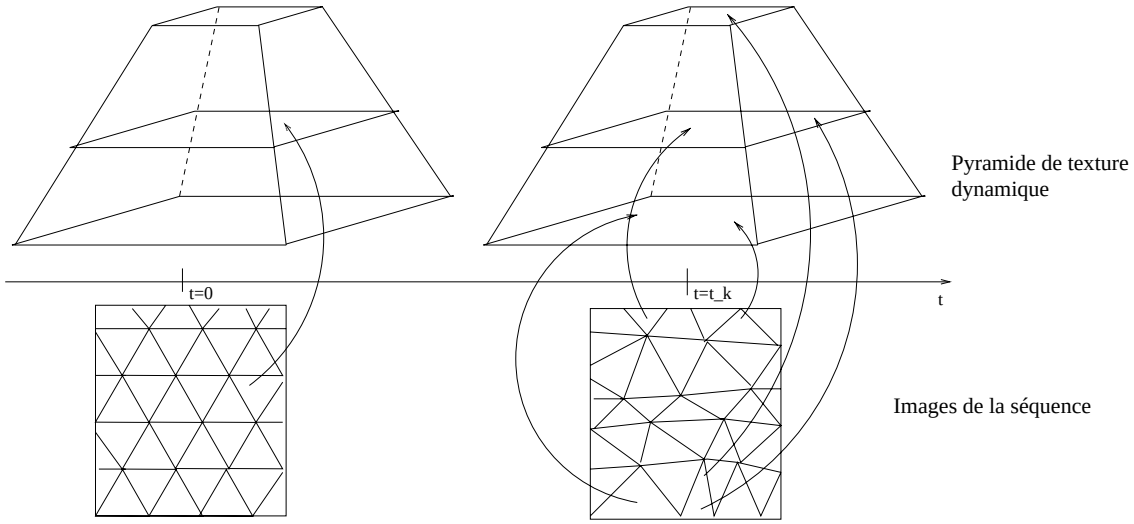


FIG. 7.23 – *Construction de la pyramide de texture à partir des images de la séquence vidéo et du mouvement estimé*

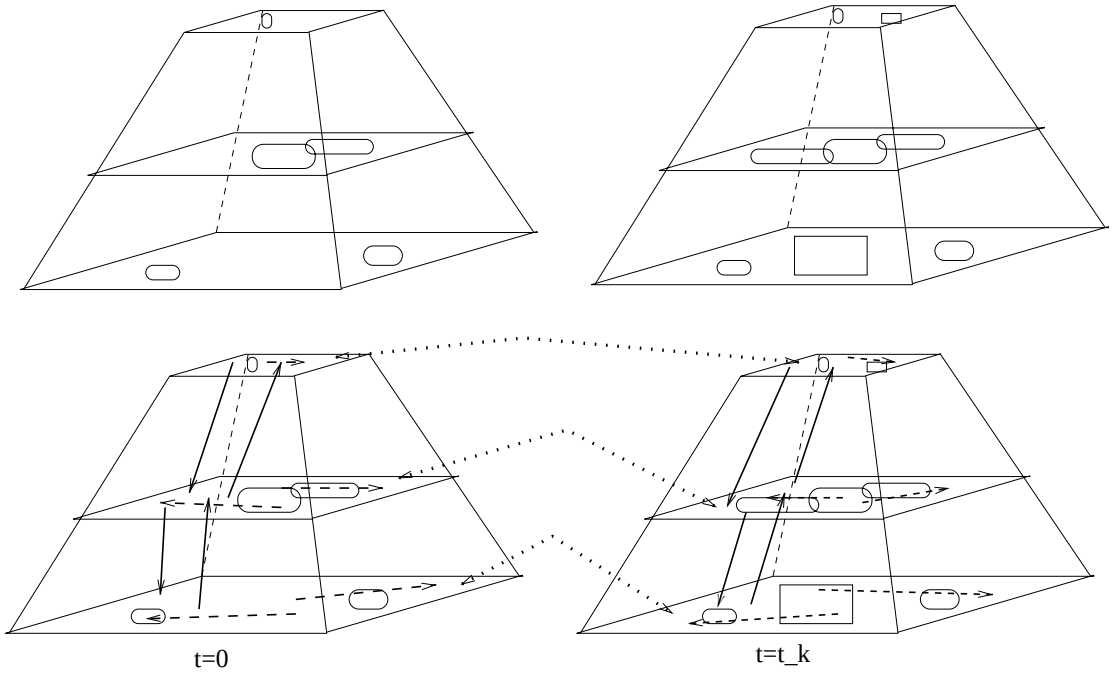


FIG. 7.24 – *Remplissage spatio-temporel et à travers les résolutions de la pyramide de texture dynamique*

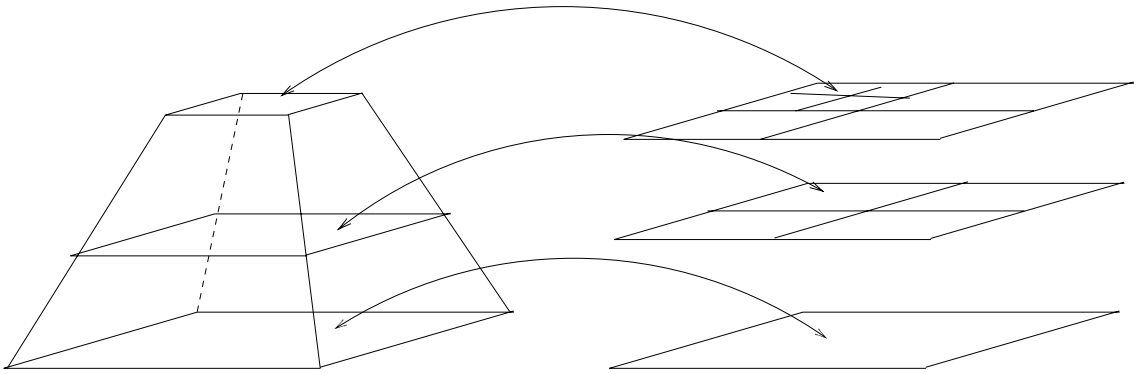


FIG. 7.25 – *Parallèle pyramide de texture et décomposition en ondelettes*

Conclusion

Cette thèse s'inscrit dans le cadre du codage vidéo scalable. Dans ce domaine, nous avons vu qu'il était nécessaire de définir de nouveaux standards de compression vidéo si l'on voulait pouvoir atteindre les mêmes performances en terme de compression qu'un codeur standard non scalable tout en fournissant un flux scalable.
Dans cette thèse, nous avons proposé un schéma de codage vidéo par transformée en ondelettes t+2D. L'utilisation des ondelettes offre une scalabilité naturelle au schéma de codage, de plus la décorrélation des informations par une transformée ondelettes est efficace dans le but de compresser ensuite le signal.
Dans les schémas de codage vidéo, l'exploitation du mouvement est le point crucial dont dépendent les performances en compression du codeur. Dans ce manuscrit, nous avons proposé un schéma de codage vidéo basé sur une approche analyse-synthèse. Cette approche permet de représenter indépendamment les informations de mouvement et de texture et de les coder séparément. Le mouvement est représenté par des maillages 2D déformables. Les maillages sont un bon outil pour l'estimation du mouvement et fournissent une texture continue bien adaptée pour une décorrélation par transformée ondelettes. Le suivi long terme de la texture par les maillages permet la projection des images de la vidéo sur des grilles de référence et permet ainsi de s'affranchir du mouvement dans la transformée temporelle. Au niveau du schéma de codage, les contributions ont portées sur l'optimisation des différentes briques du schéma de codage par analyse-synthèse:

- la transformée en ondelettes temporelles a été améliorée par l'utilisation d'une transformée ondelettes en version lifting. Ceci a permis d'assurer la réversibilité du schéma dans le cas de mouvements sous-pixelliques.
- le choix de deux grilles de référence lors du redressement des images dans la phase d'analyse a permis d'améliorer la qualité globale des images reconstruites de la séquence vidéo en diminuant la distance entre les images projetées et leur référence. Cette technique nécessite alors quelques compensations en mouvement dans la transformée temporelle, qui n'étaient pas nécessaires avec une projection sur une grille unique de référence. Cependant, d'une part le nombre de compensation est limitée, d'autre part, la scalabilité du mouvement n'est pas limitée dans cette méthode, elle n'influence pas les performances de la transformée car le mouvement nécessaire pour les compensations est codé dans la couche de base qui n'est pas scalable. Le mouvement scalable est codé dans la couche haute et n'est utilisé que lors de la synthèse des images reconstruites.

- l'utilisation d'une technique de remplissage spatio-temporelle a permis d'augmenter les performances en compression. Le remplissage est nécessaire sur les bords des images de texture car la projection des images est faite sur un support plus large afin de prendre en compte les zones entrantes dans la vidéo. L'utilisation d'informations de GOFs futurs a permis d'améliorer le codage des images de la couche de base et d'une manière plus générale la qualité de toutes les images de la séquence reconstruite.
- la scalabilité du mouvement a été mise en œuvre en tenant compte du format de restitution de la séquence et en adaptant la distorsion au décodage au format de restitution.
- concernant la mesure de qualité objective des performances du codeur proposé, le codage du mouvement avec pertes implique qu'une mesure telle que le PSNR entre la séquence originale et la séquence reconstruite est biaisé. Partant du fait que les erreurs sur le mouvement ne sont pas visibles à l'œil, nous avons montré que la qualité visuelle globale de la séquence est mieux représentée par un PSNR calculé dans le domaine texture (entre les images de texture codées-décodées).
- afin de rendre le codeur vidéo pleinement scalable, nous avons mis en œuvre les différentes scalabilités au sein de ce codeur (temporelle, spatiale, SNR).
- enfin, nous avons positionné notre approche d'une part par rapport à un codeur standard non scalable (H264/AVC) et d'autre part, par rapport à des codeurs scalables proposés en normalisation MPEG. En terme de compression, notre approche offre des performances proche de celles du codeur H264, voire parfois meilleures sur certaines séquences. En terme de performances comparées à d'autres codeurs scalables, notre approche offre des performances assez bonnes. Les performances des différents codeurs varient selon les débits et les formats restitués, de plus, la mesure objective de la qualité entre tous les codeurs reste problématique.

Au niveau des performances de l'approche, nous avons vu que la qualité de l'estimation du mouvement influe beaucoup sur les performances en compression du codeur. En effet, dans les zones de découvrement/recouvrement, l'estimation du mouvement échoue et rend la prédiction de la texture très mauvaise. Pour résoudre ce problème, nous avons proposé une nouvelle représentation du mouvement par maillages avec prise en compte de lignes de discontinuité locales. Les contributions mises en œuvre sur cette partie sont les suivantes:

- une réflexion a été menée afin de définir ce qu'est une zone d'occlusion dans le maillage, ce qu'est une ligne de discontinuité dans cette zone et où est placée cette ligne.
- une technique de détection de la zone de discontinuité dans le maillage a été mise en œuvre. La détection utilise la déformation des mailles obtenues après une première estimation du mouvement.
- la structure de maillage a été modifiée dans la zone d'occlusion au niveau fin de la hiérarchie du maillage, prenant en compte un contour défini sur les mailles de la zone de discontinuité.

- la discontinuité et le remaillage ont été remonté dans la hiérarchie vers les niveaux plus grossiers, jusqu'à la disparition de la ligne de discontinuité.
- une technique de z-order a été utilisée pour savoir quelles mailles servaient à la reconstruction d'une image au temps t.
- l'amélioration de l'estimation du mouvement grâce à l'utilisation de la nouvelle structure de maillage a été montrée sur certaines séquences présentant d'importants découvrements/recouvrements de texture.
- le problème de la représentation de la texture pour une application de la structure de maillage à la représentation du mouvement dans un codeur vidéo a été abordé et quelques idées de solutions ont été proposées.

Perspectives

Les perspectives de cette étude concernant le schéma de codage vidéo scalable portent sur l'étude de la brique de codage des sous-bandes spatio-temporelles. L'outil de codage utilisé est aujourd'hui le codeur EBCOT, nous avons vu que le codeur de sous-bandes EZBC donne également des résultats intéressants. Ces deux codeurs offrent des performances complémentaires. Une amélioration du codage des sous-bandes peut peut-être être obtenue par un codage hybride par ces deux codeurs. De plus, il serait également intéressant de prendre en compte la dimension 3D des sous-bandes à coder. Le codage des images d'erreurs peut, quant à lui, être amélioré par l'utilisation d'ondelettes de secondes générations.
Enfin, afin d'assurer une scalabilité complète spatio-temporelle, en qualité et en complexité, une amélioration du schéma porterait sur la définition d'une couche basse codée au format QCIF (ou au plus petit format souhaité au décodage). En effet, le codeur scalable aujourd'hui ne supporte pas la scalabilité en complexité sans dérive. La scalabilité spatiale et temporelle est obtenue à partir d'une scalabilité SNR sur le flux, le décodage est effectué au format d'encodage quelque soit le format de restitution demandé. La scalabilité en complexité n'est alors pas prise en compte, si elle l'était le décodage effectué au format de restitution (différent du format d'encodage) impliquerait une dérive. La définition d'une couche basse au format QCIF permettrait d'éviter cette dérive au décodage et d'offrir toutes les scalabilités.

Concernant la représentation du mouvement par maillages avec lignes de rupture, une perspective concerne l'automatisation de la détection de la ligne de discontinuité par l'utilisation de techniques comme les contours actifs. Dans notre cas, le problème de la détection de la ligne de discontinuité est celui de la segmentation. Nous disposons cependant de certaines contraintes qui nous permettent d'assurer une bonne segmentation. En effet, la zone à segmenter est restreinte à la zone définie par les mailles déformées et le contour à rechercher est a priori un contour de fort contraste.

Une autre perspective est l'extension du schéma au cas de plusieurs lignes de discontinuité dans la même zone, dans le cas où plus de deux objets ont créé une discontinuité.

Au niveau des perspectives globales de l'étude, la première concerne la représentation de la texture par des techniques de paramétrisation permettant l'application de la structure de maillage au codage vidéo scalable. La structure de maillage avec lignes de discontinuité ne permet pas de représenter la texture sur un plan 2D. Elle donne une représentation surface (3D) de la texture qui est difficilement exploitable pour une application directe au codage vidéo. Une phase de reparamétrisation de la surface est nécessaire afin de pouvoir appliquer les outils de codage vidéo connus.
Une autre représentation de la texture peut être obtenue à l'aide d'une représentation maillée de la surface. La surface peut alors être codée par des outils de codage de maillage. Cette surface peut être codée par exemple par des ondelettes de secondes générations.
Enfin, une dernière perspective concernant la représentation de la texture et permettant la gestion des changements de résolution dus à la représentation par maillages est la représentation à l'aide d'une pyramide multi-résolution. Les changements de résolutions dus à la déformation des mailles créent des motifs difficiles à coder et peuvent entrainer des artéfacts visuels tels que l'aliasing ou un étalement des textures. L'utilisation d'une pyramide multi-résolution pour représenter la texture permettrait de gérer ces changements de résolution et d'offrir une représentation de la texture analogue à une représentation ondelettes.

Annexe A

Le lifting

On considère l'analyse et la synthèse d'un signal X par une transformée ondelette selon la figure A.1. Les basses fréquences Bf sont obtenues par filtrage du signal X avec le filtre passe-bas $\tilde{h}$, puis décimation du signal filtré, les hautes fréquences Hf par filtrage avec $\tilde{g}$ et décimation. La reconstruction du signal est effectuée par filtrage avec les filtres g et h, on obtient le signal reconstruit Xr. La reconstruction parfaite est assurée si les conditions suivantes sont vérifiées: $\frac{1}{2}(\tilde{h}(z)h(z) + \tilde{g}(z)g(z)) = I$ et $h(z)\tilde{h}(-z) + g(z)\tilde{g}(-z) = 0$. C'est toujours le cas avec des filtres biorthogonaux ou duaux.
Nous allons développer plus explicitement le calcul des basses fréquences Bf à l'aide du filtre $\tilde{h}$:
$Bf_0 = x_0\tilde{h}_0 + x_1\tilde{h}_1 + x_2\tilde{h}_2 + x_3\tilde{h}_3 + ...$
$Bf_1 = x_1\tilde{h}_0 + x_2\tilde{h}_1 + x_3\tilde{h}_2 + x_4\tilde{h}_3 + ...$
$Bf_2 = x_2\tilde{h}_0 + x_3\tilde{h}_1 + x_4\tilde{h}_2 + x_5\tilde{h}_3 + ...$
$Bf_3 = x_3\tilde{h}_0 + x_4\tilde{h}_1 + x_5\tilde{h}_2 + x_6\tilde{h}_3 + ...$
Après filtrage du signal X par le passe-bas, nous décimons les coefficients transformés, les coefficients restant sont:
$Bf_0 = x_0\tilde{h}_0 + x_1\tilde{h}_1 + x_2\tilde{h}_2 + x_3\tilde{h}_3 + ...$

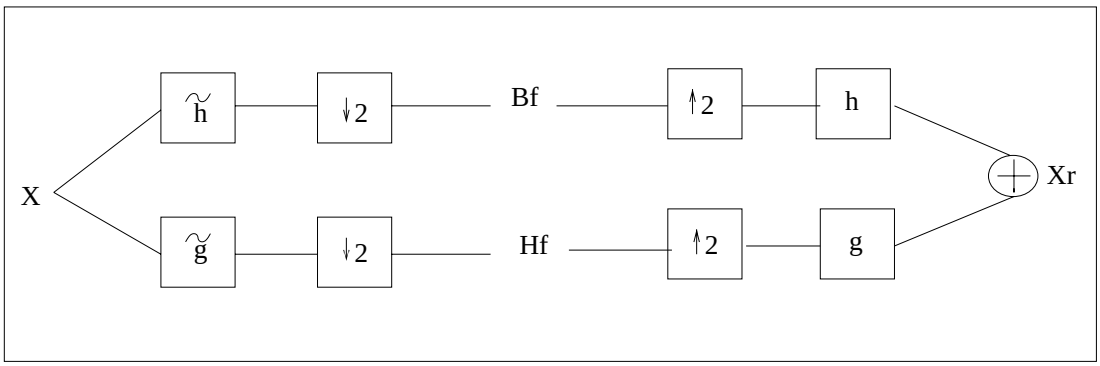


FIG. A.1 – *Analyse en ondelette et reconstruction du signal X*

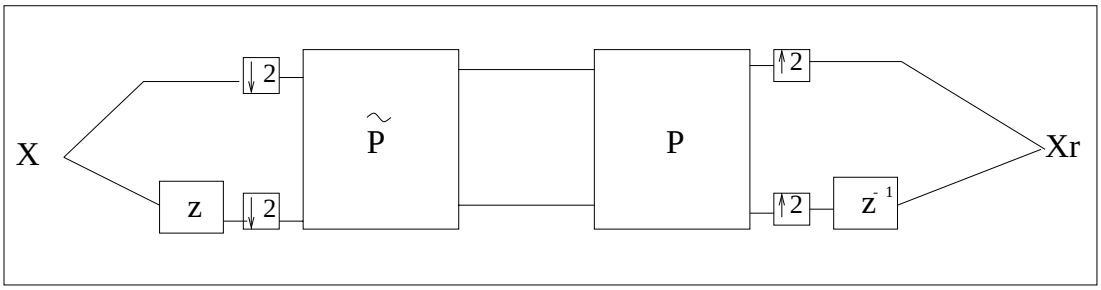


FIG. A.2 – *Analyse en ondelette et reconstruction du signal X avec matrices polyphases*

$Bf_2 = x_2\tilde{h}_0 + x_3\tilde{h}_1 + x_4\tilde{h}_2 + x_5\tilde{h}_3 + ...$
On remarque l'on peut réécrire le calcul des coefficients basses fréquences sous une forme polyphase, on sépare les coefficients d'indice pair et impair:
$Bf = \tilde{h}_e x_e + \tilde{h}_o x_o$, avec x_e et x_o les coefficients du signal d'indice pair et impair. De la même manière, on obtient les hautes fréquences, ainsi que le signal reconstruit. Les équations de l'analyse et de la synthèse sont alors:

Analyse
$$Bf = \tilde{h}_e x_e + \tilde{h}_o x_o$$
$$Hf = \tilde{g}_e x_e + \tilde{g}_o x_o$$
Synthèse
$$Xr_e = h_e(Bf) + g_e(Hf)$$
$$Xr_o = h_o(Bf) + g_o(Hf)$$

On peut ainsi définir deux matrices polyphases duales pour l'analyse et la synthèse du signal $\tilde{P}$ et P:

$$\tilde{P} = \begin{bmatrix} \tilde{h}_e & \tilde{h}_o \\ \tilde{g}_e & \tilde{g}_o \end{bmatrix}$$

et

$$P = \begin{bmatrix} h_e & g_e \\ h_o & g_o \end{bmatrix}$$

La figure A.2 illustre la transformée ondelette correspondant à celle définie précédemment avec les matrices polyphases. La reconstruction parfaite est assurée avec la condition suivante: $\tilde{P}(z^{-1})P(z) = I$.
On montre que l'on peut factoriser $\tilde{P}$ sous la forme:

$$\tilde{P} = \begin{pmatrix} K & 0 \\ 0 & \frac{1}{K} \end{pmatrix} \prod_{i=0}^{M} \begin{pmatrix} 1 & u_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -p_i & 1 \end{pmatrix}$$

La figure A.3 montre un étage de lifting et un étage de lifting dual. La transformée ondelette est une succession de ces étages.
La transformée inverse du lifting s'obtient facilement en remplaçant les additions par des soustractions (figure A.4).

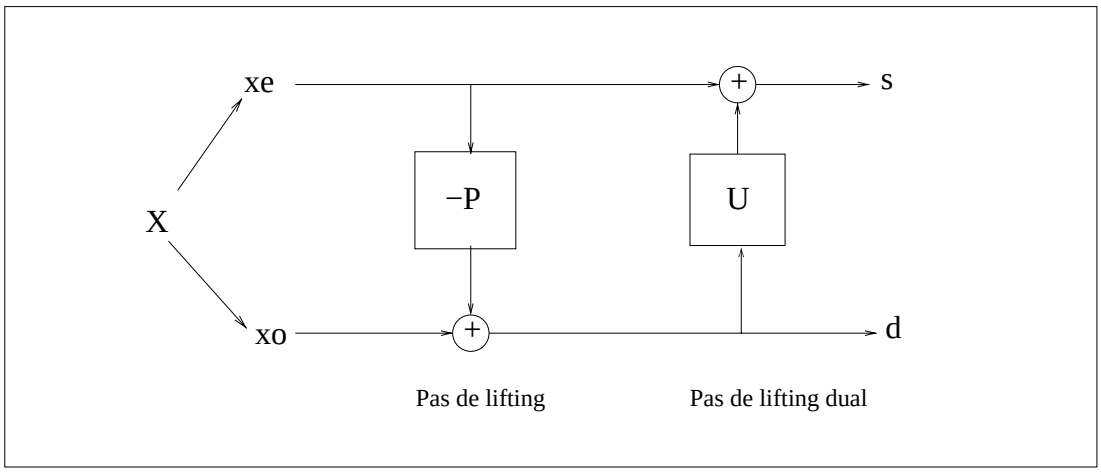


FIG. A.3 – *La transformée lifting*

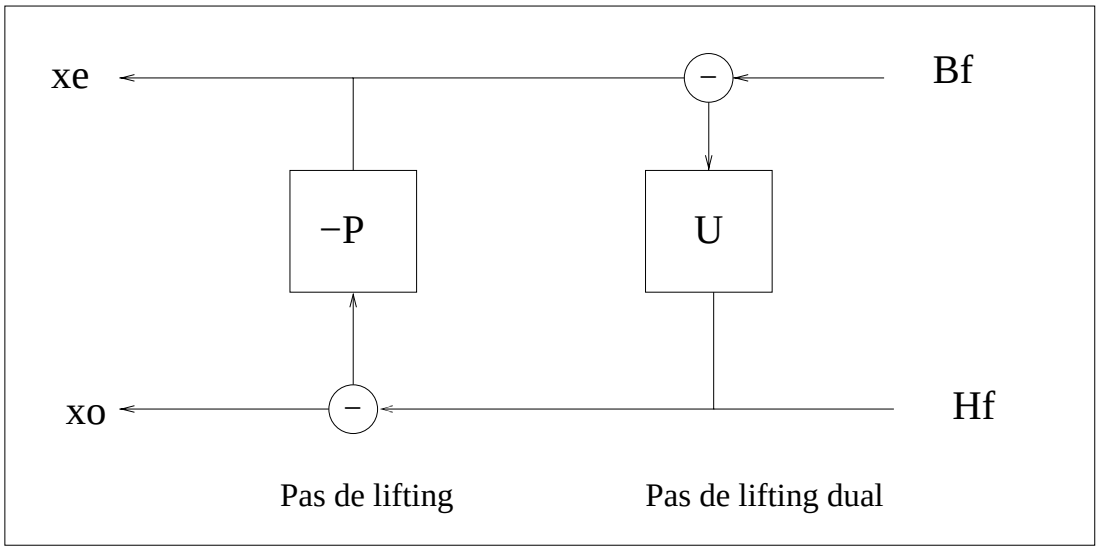


FIG. A.4 – *La transformée lifting inverse*

Bibliographie

[Altunbasak 97] Y. Altunbasak & A.M. Tekalp. *Occlusion-adaptive, content-based mesh design and forward tracking.* IEEE Transactions on image processing, vol. 6, no. 9, pages 1270–1280, Septembre 1997.

[Andre 04] T. Andre, M. Cagnazzo, M. Antonini, M. Barlaud, N. Bozinovic & J. Konrad. *(N,0) Motion-compensated lifting-based wavelet transform.* In International Conference on Acoustics, Speech and Signal Processing, ICASSP-2004, Septembre 2004.

[Andreopoulos 02] Y. Andreopoulos, A. Munteanu, G. van der Auwera, P. Schelkens & J. Cornelis. *Wavelet-based Fully-scalable Video Coding With In-band Prediction.* In Proc. Third IEEE Benelux Signal Processing Symposium (SPS-2002), Mars 2002.

[Andreopoulos 03] Y. Andreopoulos, M. van der Schaar, A. Munteanu, J. Barbarien, P. Schelkens & J. Cornelis. *Complete-to-overcomplete Discrete Wavelet Transforms for Scalable Video Coding With MCTF.* In Proceedings of SPIE, editeur, Visual Communications and Image Processing 2003, volume 5150, 2003.

[Blaszak 04] L. Blaszak, M. Domanski, R. Lange & A. Luczak. *Scalable AVC Codec.* ISO/IEC JTC1/SC29/WG11 MPEG04/M10569/S13, Munich, March 2004.

[Bottreau 01] V. Bottreau, M. Bénetière, B. Felts & B. Pesquet-Popescu. *A fully scalable 3D subband video codec.* In IEEE International Conference on Image Processing, ICIP'01, volume 2, pages 1017–1020, October 2001.

[Bottreau 02] V. Bottreau, E. Barrau & A. Bourge. *Architecture and features of fully scalable motion-compensated 3D subband codec.* Rapport technique, ISO/IEC JTC1/SC29/WG11 MPEG 2002/m7977, mars 2002.

[Bozinovic 04] N. Bozinovic & J. Konrad. *Mesh-based motion models for wavelet video coding.* In Proceedinds International Conference on Acoustics, Speech and Signal Processing, ICASSP-2004, Septembre 2004.

[Brangoulo 04] S. Brangoulo & P. Gioia. *An adaptive video coder using saliency and second generation wavelets.* In IASTED 6th Conference on Si-

gnal and Image Processing, pages 286–291, Honolulu, Hawaii U.S.A, August 2004.

[Buhan 98] C. Le Buhan. *Progressive geometrical compression of arbitrary shaped video objects*. PhD thesis, EPFL, 1998.

[Cammas 03] N. Cammas, S. Pateux & N. Laurent. *Procédés et dispositifs de codage et de décodage d'une séquence d'images par décomposition mouvement/texture et codage par ondelettes*. Brevet français n°03 03449, Mars 2003.

[cfp 03] *Call for proposals on Scalable Video coding Technology*. ISO/IEC JTC1/SC29/WG11 MPEG2003/N6193, December 2003.

[Chaumont 03] M. Chaumont. *Représentation en objets vidéo pour un codage progressif et concurrentiel des séquences d'images*. PhD thesis, Université de Rennes 1, Novembre 2003.

[Chen 02] P. Chen & J.W. Woods. *Video Coding for Digital Cinema*. In IEEE International Conference on Image Processing, ICIP'02, 2002.

[Chen 03a] P. Chen. *Fully scalable subband/wavelet coding*. PhD thesis, Graduate Faculty of Rensselaer Polytechnic Institute, Rensselaer Polytechnic Institute, Troy, New York, may 2003.

[Chen 03b] P. Chen, K. Hanke, T. Rusert & J. W. Woods. *Improvements to the MC-EZBC scalable video coder*. In International Conference on Image Processing 2003, ICIP'03, 2003.

[Choi 99] S-J. Choi & J.W. Woods. *Motion-Compensated 3-D Subband Coding of Video*. IEEE Transactions on Image Processing, vol. 8, no. 2, pages 155–167, february 1999.

[Chrysafis 99] Christos Chrysafis, David Taubman & Alex Drukarev. *Overview of JPEG2000*, 1999.

[Cohen 89] A. Cohen, I. Daubechies & J.C. Feauveau. *Biorthogonal Bases of Compactly Supported Wavelets*. Rapport technique TM 11217-900529-07, AT&T Bell Laboratories, 1989.

[Desbrun 02] M. Desbrun, M. Meyer & P. Alliez. *Intrinsic Parameterizations of Surface Meshes*. Eurographics 2002, vol. 21, no. 2, 2002.

[Dudon 95] M. Dudon, O. Avaro & C. Roux. *Triangle-based motion estimation and temporal estimation*. In IEEE Workshop on nonlinear signal processing, page 242, 1995.

[Dudon 96] M. Dudon. *Modélisation du mouvement par treillisactifs et méthodes d'estimation associées. Application au codage de séquences d'images.* PhD thesis, Université de Rennes 1, Décembre 1996.

[Dudon 97] M. Dudon, O. Avaro & C. Roux. *Triangular active mesh for motion estimation*. Signal Processing: Image Communication, vol. 10:21, no. 41, 1997.

[Eren 03] P. E. Eren & A. M. Tekalp. *Bi-directional 2-D mesh representation for video object rendering, editing and superresolution in the*

presence of occlusion. Signal Processing: Image Communication, vol. 18, pages 321–336, 2003.

[Feibush 80] E.A. Feibush, M. Levoy & R.L. Cook. *Synthetic Texturing Using Digital Filters*. In Computer Graphics SIGGRAPH'80 Proceedings, volume 14, pages 294–301, July 1980.

[Gangnet 84] M. Gangnet & D. Ghazanfarpour. *Comparaison de techniques de mise en perspective de textures planes*. In Actes du Premier Colloques International d'images électroniques, Biarritz, CESTA, pages 29–35, Mai 1984.

[Golwelkar 03] A. Golwelkar & J. W. Woods. *Scalable video compression using longer motion compensated temporal filters*. In Visual Communications and Image Processing 2003, VCIP2003, 2003.

[Han 98] S. Han & J.W. Woods. *Adaptive coding of moving objects for very low bit-rates*. IEEE Journal on Selected Areas in Communications, vol. 16, pages 56–70, 1998.

[Heckbert 89] Paul S. Heckbert. *Fundamentals of Texture Mapping and Image Warping*. PhD thesis, Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, June 1989.

[Hsiang 00] S-T. Hsiang & J.W. Woods. *Embedded Image Coding using Zeroblocks of Subband/Wavelet Coefficients and Context Modeling*. In IEEE International Symposium on Circuits and Systems, 2000.

[Hsiang 01] S.T. Hsiang & J.W. Woods. *Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank*. Signal Processing: Image Communication, vol. 16, pages 705–724, 2001.

[Hsiang 02] S-T. Hsiang. *Highly scalable subband/wavelet image and video coding*. PhD thesis, Graduate faculty of Rensselaer Polytechnic Institute, January 2002.

[Karlsson 88] G. Karlsson & M. Vetterli. *Three Dimensional Subband Coding of Video*. In IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'88, pages 1100–1103, 1988.

[Kim 97] B.J. Kim & W.A. Pearlman. *An Embedded Wavelet Video Coder Using Three-Dimensional Set Partitioning in Hierarchical Trees (SPIHT)*. In IEEE Data Compression Conference DCC'97, pages 221–260, 1997.

[Kim 00] B-J. Kim, Z. Xiong & W. A. Pearlman. *Low Bit-Rate, Scalable Video Coding with 3D Set Partitioning in Hierarchical Trees (3D SPIHT)*. IEEE Transactions on Circuits and Systems for Video Technology, vol. 10, no. 8, 2000.

[Laurent 98] N. Laurent, P. Lechat & H. Sanson. *Limitation of triangles overlapping in mesh-based motion estimation using augmented lagrangian*. In 5th IEEE International Conference on Image Processing, ICIP'98, volume 2, pages 223–227, Chicago, USA, octobre 1998.

[Laurent 00] N. Laurent. *Hierarchical mesh-based global motion estimation, including occlusion area detection.* In 7th IEEE International Conference on Image Processing, ICIP'00, Vancouver, Canada, septembre 2000.

[Li 01] W. Li. *Overview of Fine Granularity Scalability in MPEG-4 Video Standard.* IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 3, pages 301–317, March 2001.

[Luo 01] L. Luo, J.Li, S.Li, Z.Zhuang & Y-Q.Zhang. *Motion-Compensated Lifting Wavelet And Its Application In Video Coding.* In IEEE International Conference on Multimedia and Expo, August 2001.

[Luo 03] L. Luo, F. Wu, S. Li & Z. Zhuang. *Advanced lifting-based motion-threading technique for the 3D wavelet video coding.* In Visual Communications and Image Processing, VCIP 2003, 2003.

[Mallat 89] S. Mallat. *A theory for multiresolution signal decomposition: the wavelet representation.* IEEE Transactions on Pattern Analysis and Machine intelligence, vol. 11, no. 7, pages 674–693, july 1989.

[Marquant 00] Gwenaelle Marquant. *Représentation par maillage adaptatif déformable pour la manipulation et la communication d'objets vidéo.* PhD thesis, Université de rennes 1, 2000.

[Mertins 98] A. Mertins. *Optimized Biorthogonal Shape Adaptive Wavelets.* In Proc. ICIP'98, volume 3, pages 673–677, Chicago, USA, October 1998.

[Minami 01] G. Minami, Z. Xiong, A. Wang & S. Mehrotra. *3D wavelet coding of video with arbitrary regions of support.* IEEE Transactions on circuits and systems for video technology, vol. 11, no. 9, september 2001.

[N5559 03] N5559. *Call for Evidence on Scalable Video Coding Advances*, March 2003.

[N6025 03] N6025. *Requirements and Applications for Scalable Video Coding*, october 2003.

[N6373 04] N6373. *Description of Core Experiments in SVC.* ISO/IEC JTC1/SC29/WG11 N6373, Munich, March 2004.

[Odobez 94] J.M. Odobez. *Estimation, détection et segmentation du mouvement: une approche robuste et markovienne.* PhD thesis, Université de Rennes 1, 1994.

[Ohm 94] J.R Ohm. *Three-Dimensional Subband Coding with Motion Compensation.* IEEE Transactions on Image Processing, vol. 3, no. 5, pages 559–571, September 1994.

[Pateux 01] S. Pateux, G. Marquant & D. Chavira-Martinez. *Object mosaicking via meshes and cracklines technique. Application to low bit-rate video coding.* In Proceedings of Picture Coding Symposium 2001, Séoul, Korea, 2001.

[Podilchuk 95] C. I. Podilchuk, N.S. Jayant & N. Farvadin. *Three Dimensional Subband Coding of Video*. IEEE Transactions on Image Processing, vol. 4, no. 2, pages 125–139, February 1995.

[Reichel 03] J. Reichel & F. Ziliani. *Controlled Temporal Haar Transform for video coding*. In IEEE International Conference on Image Processing, ICIP'03, 2003.

[Said 96] A. Said & W.A. Pearlman. *A new fast and efficient image codec based on set partitioning in hierarchical trees*. IEEE Transactions on circuits and systems for video technology, vol. 6, June 1996.

[Schäfer 03] R. Schäfer, T. Wiegand & H. Schwarz. *The emerging H264/AVC standard*. EBU Technical Review- European Broadcasting Union - Union européenne de Radio-Télévision, no. 293, janvier 2003.

[Schilling 96] A. Schilling, G. Knittel & W. Strasser. *Texram: A smart memory for texturing*. IEEE Computer Graphics and Applications, vol. 16, no. 3, pages 32–41, 1996.

[Secker 01] A. Secker & D. Taubman. *Motion-compensated highly scalable video compression using an adaptative 3D wavelet transform based on lifting*. In IEEE International Conference on Image Processing, ICIP'01, 2001.

[Secker 02] A. Secker & D. Taubman. *Highly scalable video compression using a lifting-based 3D wavelet transform with deformable mesh motion compensation*. In IEEE International Conference on Image Processing, ICIP'02, 2002.

[Shapiro 93] J. M. Shapiro. *Embedded image coding using zerotrees of wavelet coefficients*. IEEE Transactions on signal processing, vol. 41, no. 12, pages 3445–3462, December 1993.

[Sun 04] X. Sun, Y. Zhou, Y. Wang, G. Sullivan, M.-C. Lee, F. Wu & S. Li. *Progressive Fine Granularity Scalable (PFGS) video coding*. ISO/IEC JTC1/SC29/WG11 MPEG04/M10569/S06, Munich, March 2004.

[Sweldens 95] W. Sweldens. *The Lifting Scheme: A Construction of Second Generation Wavelets*. Rapport technique 1995/6, Industrial Mathematics Initiative, Departement of Mathematics, university of South Carolina, 1995.

[Taubman 94] D. Taubman & A. Zakhor. *Multirate 3-D Subband Coding of Video*. IEEE Transactions on Image Processing, vol. 3, no. 5, pages 572–588, september 1994.

[Taubman 99] D. Taubman. *High performance scalable image compression with EBCOT*. IEEE International Conference on Image Processing, ICIP'99, 1999.

[Taubman 03] D. Taubman & A. Secker. *Highly scalable video compression with scalable motion coding*. In IEEE International Conference on Image Processing, ICIP'03, 2003.

[Toklu 96] C. Toklu, A.T. Erdem, M.I. Sezan & A.M. Tekalp. *Tracking motion and intensity variations using hierarchical 2-D mesh modeling for synthetic object transfiguration.* Graphical models and image processing, no. 0046, pages 553–573, 1996.

[Toklu 97] C. Toklu, A. M. Tekalp & T. Erdem. *2-D Triangular mesh-based mosaicking for object tracking in the presence of occlusion.* In SPIE, editeur, Proc. of SPIE Visual Communication and Image Processing, VCIP'97, pages 328–337, 1997.

[Turaga 02] D.S. Turaga & M. van der Schaar. *Unconstrained motion compensated temporal filtering.* International Organisation for Standardisation, ISO/IEC JTC1/ SC29/WG11 Coding of Moving Picture and Audio, mai 2002. M8338.

[van Beek 99] P. van Beek, A.M. Tekalp, N. Zhuang, I. Celasun & M. Xia. *Hierarchical 2-D mesh representation, tracking, and compression for object-based video.* IEEE Transactions on circuits and systems for video technology, vol. 9, no. 2, pages 353–369, mars 1999.

[Vieron 02] J. Vieron, C. Guillemot & S. Pateux. *Motion compensated 2D+t wavelet analysis for low rate FGS vido compression.* In International Thyrrhenian workshop on digital communications 2002 (invited paper), 2002.

[Vieron 04] J. Vieron, E. Francois, V. Bottreau, C. Guillemot, G. Marquant & G. Boisson. *Fully scalable video coding based on 2D+t wavelet technology.* ISO/IEC JTC1/SC29 WG11 MPEG04/M10569/S18, Munich, March 2004.

[Wang 94] Y. Wang & O. Lee. *Active mesh-A feature seeking and tracking image sequence representation scheme.* IEEE Transactions on image processing, vol. 3, no. 5, Septembre 1994.

[Wang 96] Y. Wang, O. Lee & A. Vetro. *Use of two-dimensional deformable mesh structures for video coding, part II- The analysis problem and a region-based coder employing an active mesh representation.* IEEE Transactions on circuits and systems for video technology, vol. 6, no. 6, pages 647–659, Décembre 1996.

[Wang 99] A. Wang, Z. Xiong, P.A. Chou & S. Mehrotra. *Three-dimensional wavelet coding of video with global motion compensation.* In IEEE Intl. Conf. on Data Compression, DCC'99, pages 404–413, March 1999.

[Williams 83] L. Williams. *Pyramidal Parametrics.* In Computer Graphics (SIGGRAPH '83 Proceedings), volume 17, pages 1–11, July 1983.

[Woods 02] J.W. Woods & P. Chen. *Improved MC-EZBC with Quarter-pixel Motion Vectors.* International Organisation for Standardisation, ISO/IEC JTC1/ SC29/WG11 Coding of Moving Picture and Audio, May 2002.

[Wu 04] Y. Wu, A. Golwelkar & J.W. Woods. *MC-EZBC video proposal from Rensselaer Polytechnic Institute*. ISO/IEC JTC1/SC29/WG11 MPEG04/M10569/S15, Munich, March 2004.

[Xiong 04] R. Xiong, F. Wu, S. Li, Z. Xiong & Y.-Q. Zhang. *Exploiting temporal correlation with adaptive block-size motion alignment for 3D wavelet coding*. In SPIE & IS&T, editeurs, Visual Communications and Image Processing 2004, VCIP'04, volume 5308, pages 144–155, 2004.

[Xu 00] J-Z. Xu, S. Li & Y-Q. Zhang. *Three-dimensional shape-adaptive discrete wavelet transforms for efficient object-based video coding*. In IEEE/SPIE Visual Communications and Image Processing, VCIP 2000, June 2000.

[Xu 01] J. Xu, Z. Xiong, S. Li & Y-Q. Zhang. *Three-Dimensionnal Embedded Subband Coding with Optimized Truncation (3-D ESCOT)*. Applied and Computational Harmonic Analysis, vol. 10, pages 290–315, 2001.

[Xu 04] J. Xu, R. Xiong, B. Feng, G. Sullivan, M.-C. Lee, F. Wu & S. Li. *3D sub-band video coding using Barbell lifting*. ISO/IEC JTC1/SC29/WG11 MPEG 2004/M10569/S05, March 2004.

[Zhan 02] Y. Zhan, M. Picard, B. Pesquet-Popescu & H. Heijmans. *Long temporal filters in lifting schemes for scalable video coding*. Rapport technique, ISO/IEC JTC1/SC29/WG11 MPEG02/M8680, July 2002.

[Zomet 98] Assaf Zomet & Shmuel Peleg. *Applying Super-Resolution to Panoramic Mosaics*, 1998.