Codage vidéo scalable un état de l'art

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

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

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

2.1 Codage vidéo hybride scalable

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

2.1.1 Schéma de principe d'un codeur vidéo hybride

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

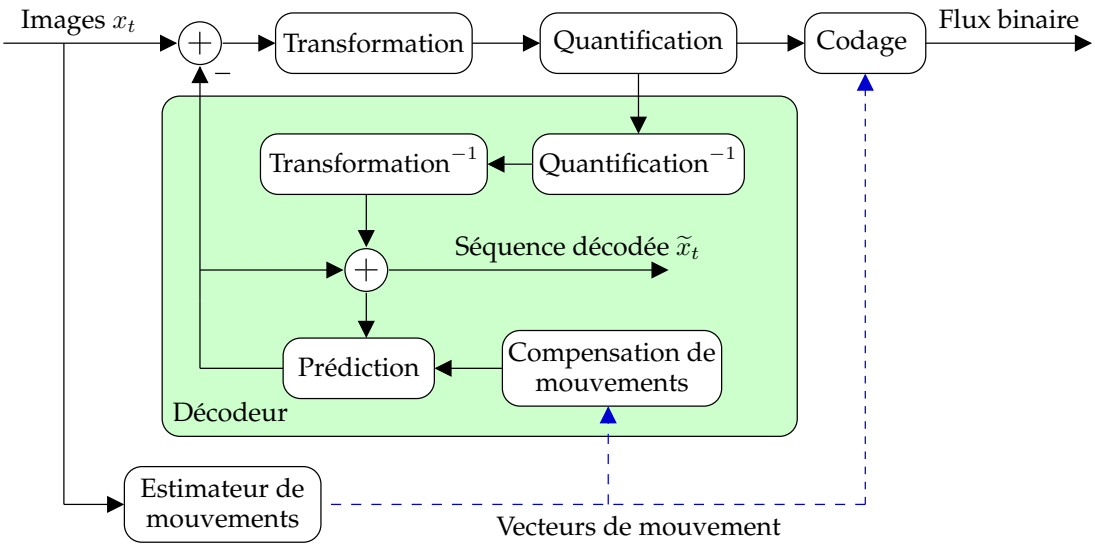

FIG. 2.1 – Schéma de principe d'un encodeur vidéo hybride avec boucle de rétroaction.

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

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

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

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

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

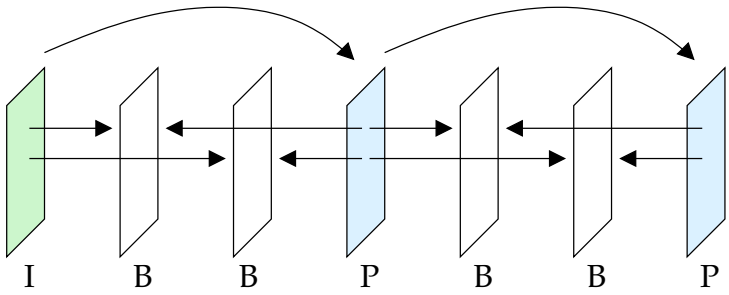

FIG. 2.2 – Agencement des modes de prédiction IBBPBBP d'un groupe d'images.

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

2.1.2 Panorama des codecs MPEG et H.26X

MPEG-1 et MPEG-2

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

MPEG-4 Part. 2

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

H.264/AVC MPEG-4 Part. 10

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

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

2.1.3 Scalabilité et l'extension MPEG-4 FGS

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

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

2.1.4 SVC ou l'extension scalable de H.264

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

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

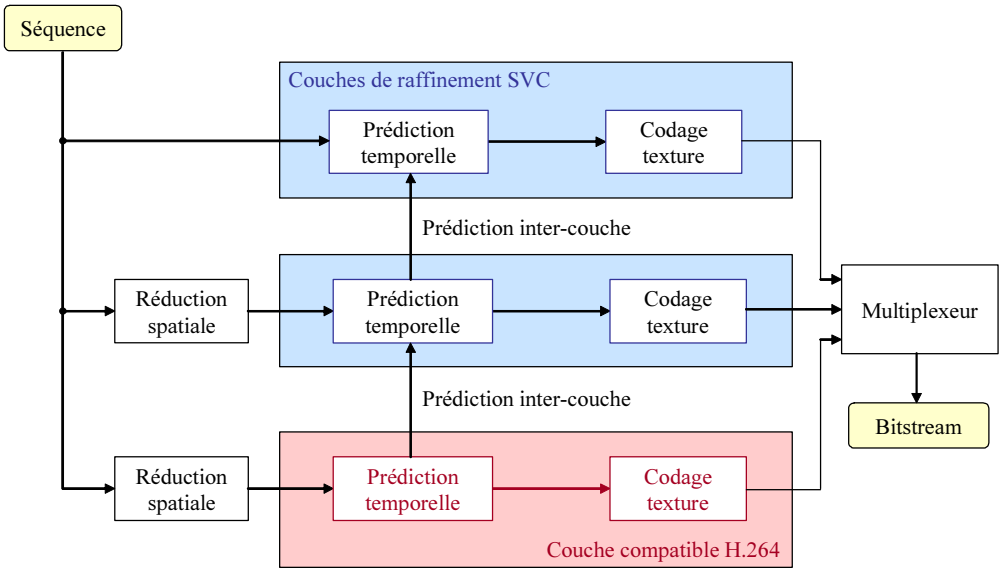

FIG. 2.3 – Structure de l'encodeur du schéma de codage SVC.

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

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

2.2 Codage vidéo scalable par ondelettes

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

2.2.1 Premières approches

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

2.2.2 Schéma de codage vidéo t+2D

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

Principe général

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

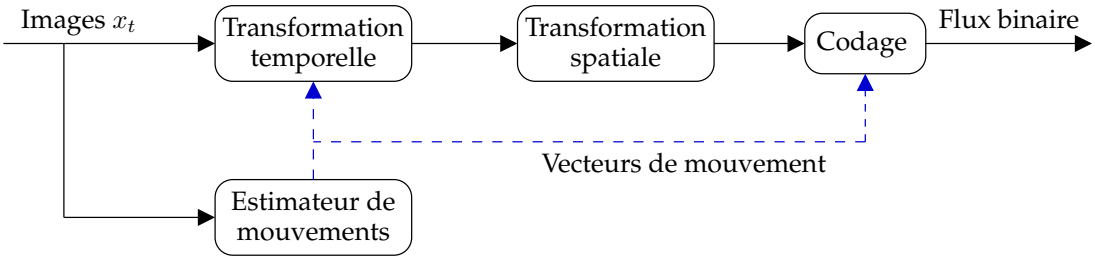

FIG. 2.4 – Schéma de principe d'un encodeur vidéo $t+2D$.

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

Extraction et scalabilité

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

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

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

2.2.3 Premiers filtres temporels

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

Schéma de codage fondateur de Ohm

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

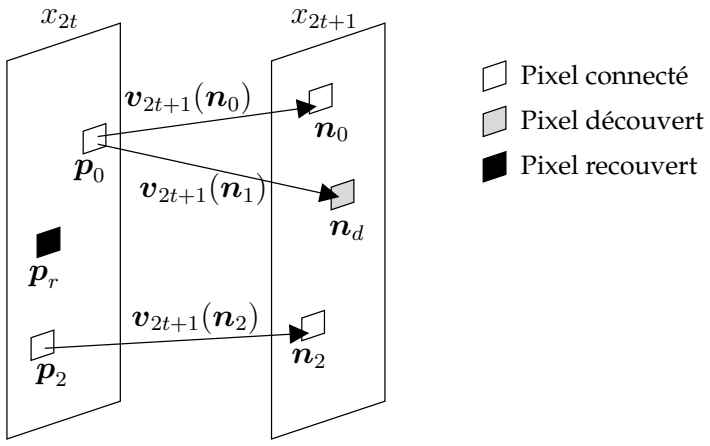

FIG. 2.5 – Filtrage de Haar compensé en mouvement selon la technique de Ohm.

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

de connexion des pixels, grâce aux équations suivantes :

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

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

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

Filtrage de Haar compensé en mouvement selon Choi et Woods

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

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

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

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

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

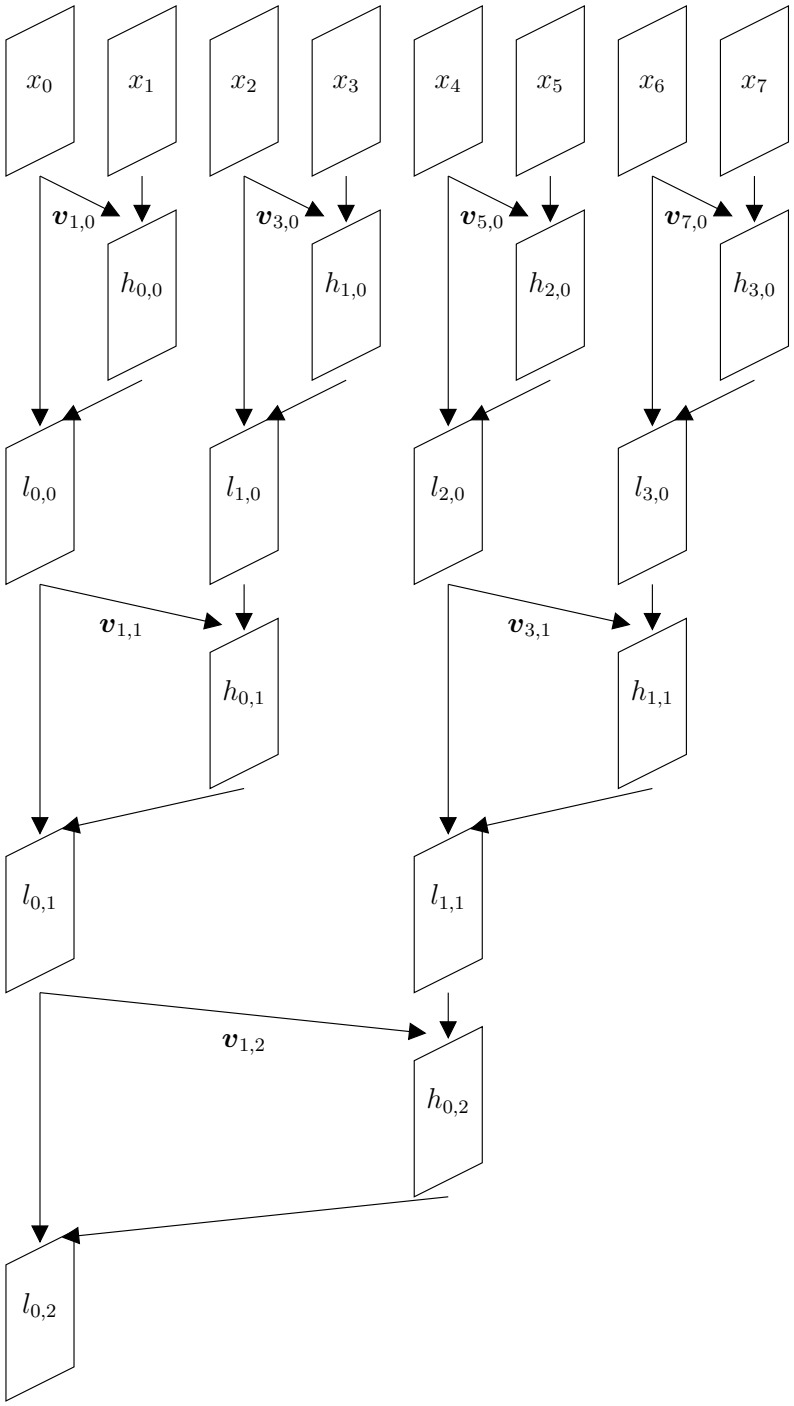

FIG. 2.6 – Décomposition temporelle de Haar d'un groupe d'images sur 3 niveaux.

Lifting temporel

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

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

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

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

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

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

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

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

Filtrage temporel

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

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

Estimation, élagage et codage des vecteurs de mouvement

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

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

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

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

Organisation du bitstream dans MC-EZBC

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

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

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

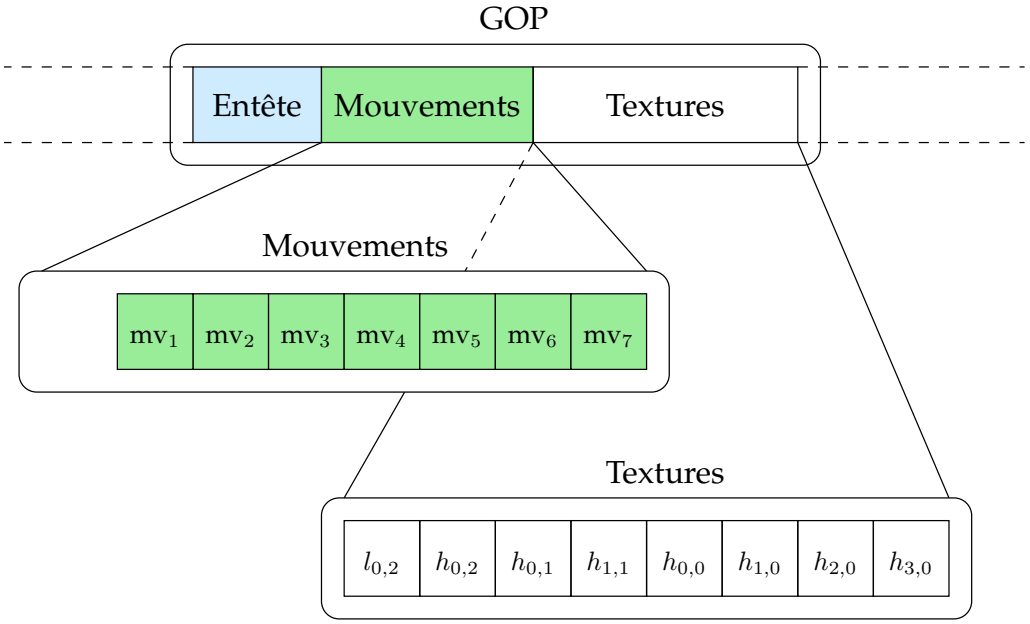

FIG. 2.7 – Organisation hiérarchique du flux vidéo *bitstream* compressé MC-EZBC.

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

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

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

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

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

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

Filtrage spatial et codage des textures

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

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

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

2.2.5 Améliorations apportées au schéma t+2D

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

Modèles de mouvement inversibles et grille triangulaire déformable

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

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

Diverses stratégies de mise à jour lors du filtrage temporel

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

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

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

Codage t+2D en boucle fermée

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

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

Utilisation de différents modes de prédiction temporelle

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

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

Autres filtres temporels

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

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

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

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

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

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

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

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

Scalabilité et codage des champs de mouvement

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

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

Schéma de codage Vidwav

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

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

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

2.3 Conclusion

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

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