Filtrage temporel

5SUR3

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

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

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

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

3.1 Filtrage temporel 5/3 compensé en mouvement

3.1.1 Notations

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

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

3.1.2 Lifting temporel

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

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

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

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

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

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

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

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

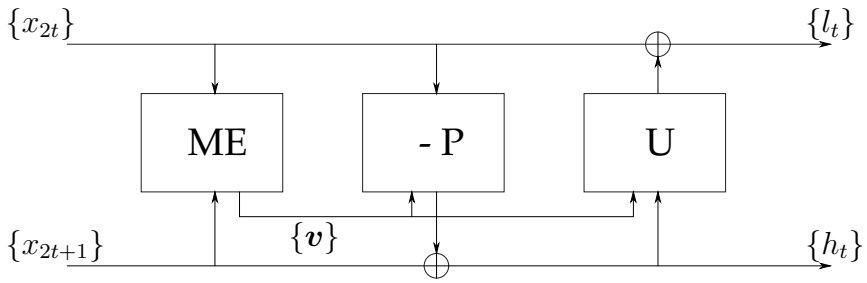

FIG. 3.1 – Structure lifting d'un filtre temporel compensé en mouvement.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

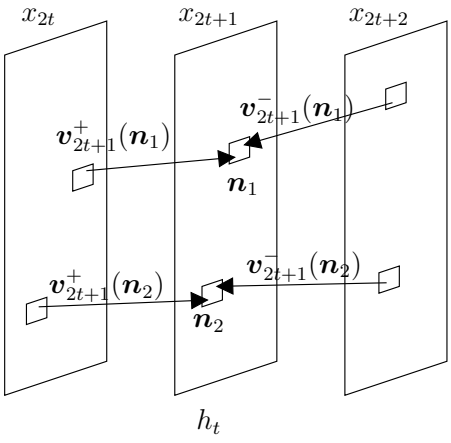

FIG. 3.2 – Opérateur de prédiction mis en jeu dans la transformée 5/3. Tous les pixels $\boldsymbol{n}_k$ sont connectés des deux côtés.

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

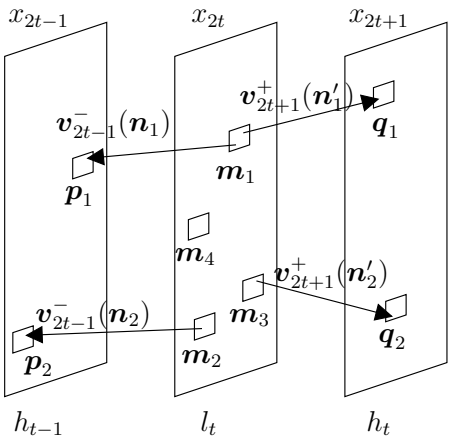

FIG. 3.3 – Opérateur de mise à jour utilisé dans la transformée 5/3. Le pixel $\boldsymbol{m}_1$ est connecté des deux côtés, les pixels $\boldsymbol{m}_2$ et $\boldsymbol{m}_3$ sont simplement connectés tandis que le pixel $\boldsymbol{m}_4$ n'est pas connecté.

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

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

Opérateur de mise à jour

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

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

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

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

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

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

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

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

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

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

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

Premier niveau temporel

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

Second niveau temporel

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

Troisième niveau temporel

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

Quatrième niveau temporel

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

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

Choix des paramètres

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

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

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

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

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

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

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

Filtrage temporel 5/3 compensé en mouvement : formulation finale

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

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

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

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

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

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

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

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

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

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

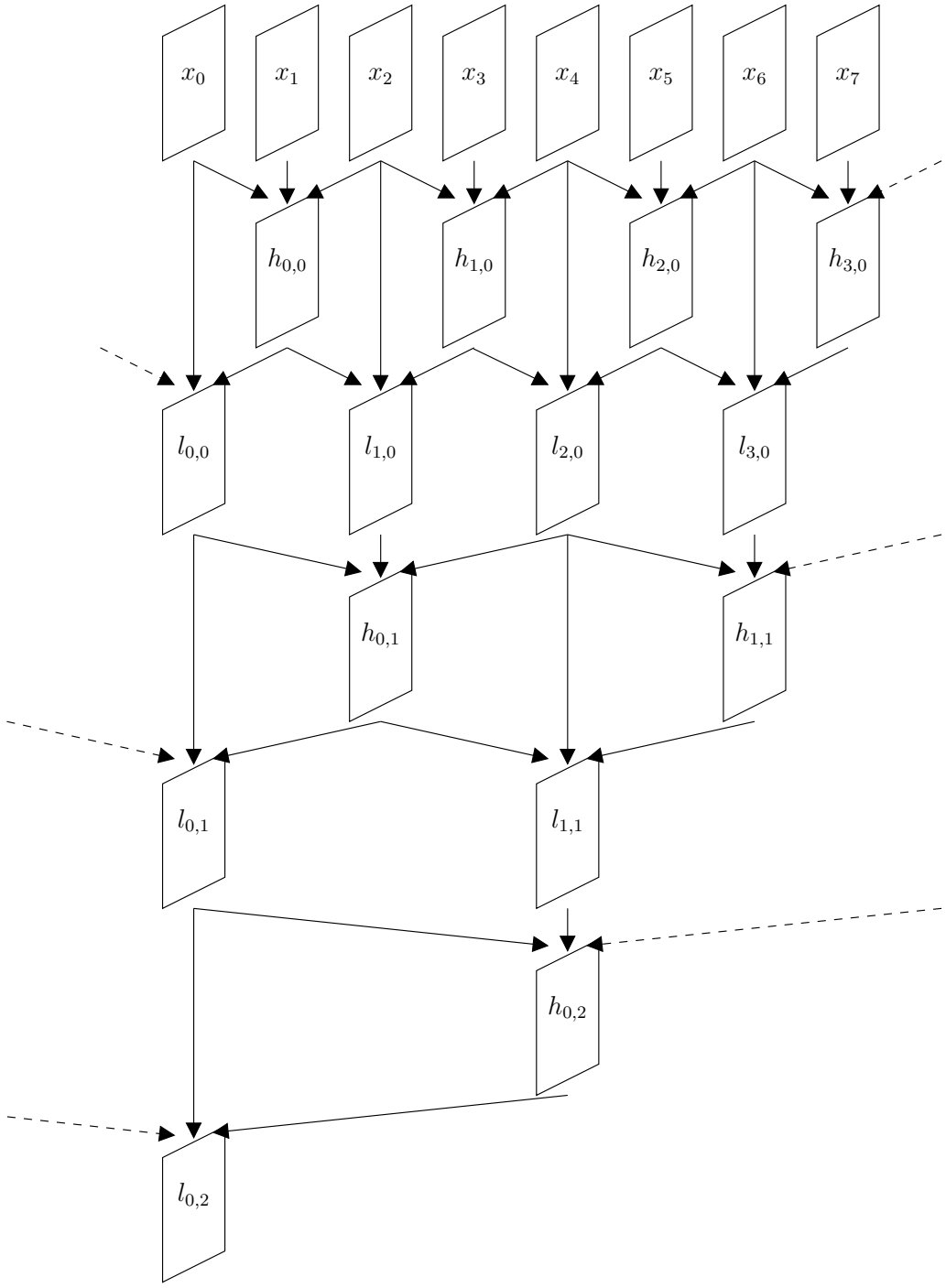

FIG. 3.4 – Analyse multirésolution temporelle 5/3 sur 3 niveaux d'une séquence vidéo.

3.1.4 Traitement au fil de l'eau

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

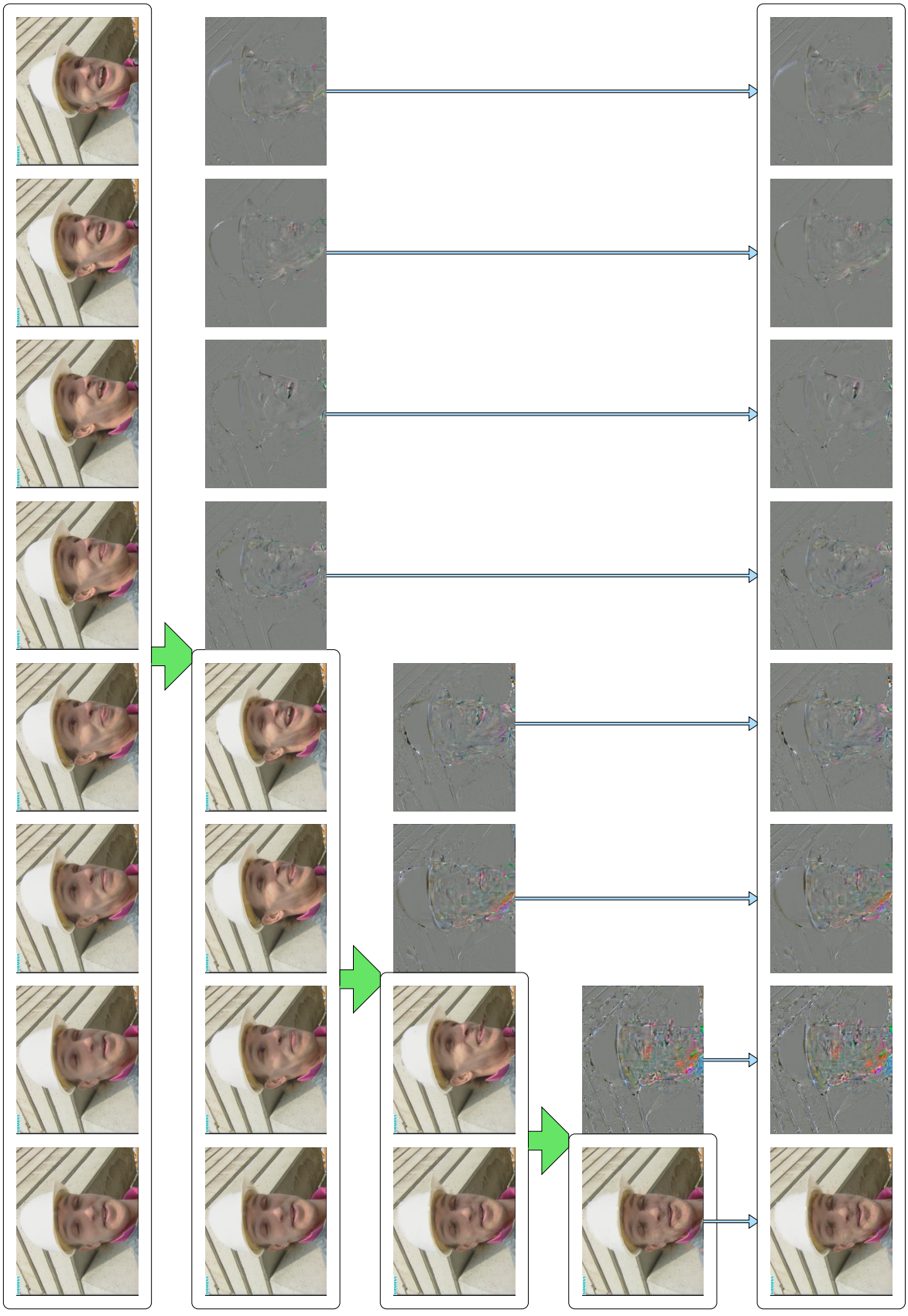

FIG. 3.5 – Analyse multirésolution temporelle 5/3 sur 3 niveaux d'un extrait de la séquence *Foreman*.

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

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

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

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

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

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

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

Algorithme de traitement au fil de l'eau

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

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

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

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

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

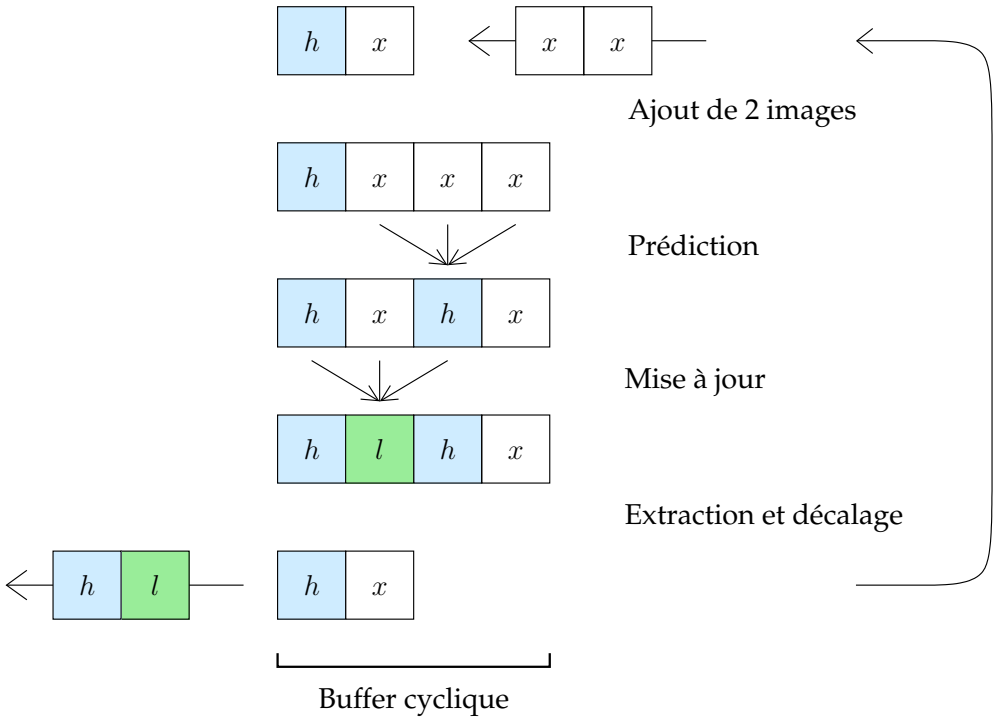

FIG. 3.6 – Schéma de fonctionnement du module de traitement au fil de l'eau de la transformée temporelle 5/3. On y observe l'évolution du buffer cyclique de filtrage.

3.2 Résultats expérimentaux

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

3.2.1 Efficacité de codage

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

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

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

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

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

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

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

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

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

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

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

3.2.2 Scalabilité temporelle

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

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

3.3 Conclusion

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

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

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