Décompositions spatio-temporelles et allocation de débit utilisant les coupures des graphes pour le codage vidéo scalable

Maria Trocan

TELECOM PARIS
école nationale supérieure des télécommunications

Thèse

Présentée pour obtenir le grade de docteur
de l'École Nationale Supérieure des Télécommunications

Spécialité : **Signal et Images**

Maria TROCAN

Décompositions spatio-temporelles et allocation de débit utilisant les coupures des graphes pour le codage vidéo scalable

Soutenue le 15 octobre 2007 devant le jury composé de:

Dominique BARBA	Rapporteurs
Enis CETIN	
Vasile BUZULOIU	Examinateurs
Marc SIGELLE	
Nicolas MOREAU	
Béatrice PESQUET-POPESCU	Directeurs de thèse
James E. FOWLER	

Remerciements

Je tiens tout d'abord à exprimer ma gratitude à l'égard de ma directrice de thèse, Béatrice Pesquet-Popescu, pour m'avoir constamment guidée, encouragée et conseillée, tout en me laissant une grande liberté, et pour la confiance qu'elle m'a accordée tout au long de mes recherches. Je remercie tout autant mon co-directeur de thèse, James E. Fowler, pour sa gentillesse, sa disponibilité et ses conseils scientifiques avisés.

Mes remerciements vont également au Professeur Vasile Buzuloiu pour m'avoir fait l'honneur de présider mon jury lors de la soutenance et pour tous ses conseils durant ces trois années de thèse; au Professeur Dominique Barba et au Professeur Enis Cetin pour avoir accepté d'être rapporteurs; au Professeur Nicolas Moreau et au Professeur Marc Sigelle pour avoir gentiment accepté de faire partie du jury.

Je tiens à remercier fortement tous les gens du laboratoire qui m'ont chaleureusement accueillie et qui m'ont beaucoup soutenue tout au long de cette thèse : Christophe, Grégoire, Dora, Mathieu, Miguel, Yiol, Gemma, Lois, Lionel, Nicolas, Mihai, Clèo, Slim, Chloé, Ioana, Nancy, Valentin, Pierre, Reda, Caroline, Roland, Jean, Cédric, Cyril H., Aurélia, Julien, ainsi qu'aux plus jeunes : Brahim, Ismael, Thomas et Sarah qui m'ont beaucoup aidée pendant la dernière ligne droite. Je remercie tout particulièrement Seb et Cyril C. pour leur disponibilité, les lectures attentives du manuscrit et leur bons conseils; Laurence et Patricia pour leur gentillesse et leur efficacité lors des difficultés administratives, Sophie - Charlotte et Fabrice pour faire tout le possible que cette thèse arrive à son terme.

Je remercie également mes amis qui m'ont encouragée pendant cette période : Ioan, Paul, Dragos, Oana, Liviu. Malgré la distance que j'ai pu prendre en me consacrant à mes recherches, qu'ils soient assurés que leur amitié m'a toujours été plus précieuse que mes travaux.

Enfin, un grand merci à mes parents, Maria et Nicolae, et à Ionut, pour leur confiance et leur soutien fort durant ces années d'études. Je leur dédie donc mon travail afin de leur témoigner toute ma reconnaissance et ma gratitude. Je dédie également cette thèse à la mémoire du Professeur Stefan Bojinca.

Résumé

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

Grâce à la représentation multirésolution inhérente, les codeurs basés-ondelettes ont la capacité de fournir une description scalable d'un signal. Ceci représente la raison principale pour laquelle le choix du paradigme du codage lifting t+2D basé-ondelettes s'impose comme cadre conceptuel de développement pour les travaux dans cette thèse.

L'objectif de ces travaux consiste en l'analyse et la conception d'un système de codage vidéo scalable. Dans un premier temps, nous nous intéressons à la construction et l'optimisation de nouvelles transformées temporelles compensées en mouvement, dans le but d'augmenter l'efficacité objective et subjective du codage. En outre, nous décrivons une meilleure représentation pour les sous-bandes temporelles en utilisant des décompositions spatiales anisotropes. Enfin, nous proposons une methode d'amélioration du codage entropique en concevant une solution basée sur la théorie des graphes, afin d'optimiser la minimisation du Lagrangien débit-distorsion.

Abstract

The recent progress in wavelet-based video coding led to the emergence of a new generation of scalable video schemes, whose performance is comparable to that of the best hybrid codecs. The t+2D subband coding methods exploit the temporal interframe redundancy by applying an open-loop temporal wavelet transform over the frames of a video sequence. The temporally-filtered subband frames are further spatially decomposed and entropy coded.

Due to their inherent multiresolution signal representation, wavelet-based coding schemes have the potential to support temporal, spatial and SNR scalability. This is the main reason for chosing the scalable lifting-based wavelet-coding paradigm as the conceptual development framework for this thesis work.

The objective of this thesis consists of the analysis and design of an efficient scalable video-coding system. In a first time, we are interested in the construction and optimization of motion-compensated temporal coding schemes, in order to enhance both the objective and subjective coding quality. Moreover, we describe a better representation of the temporal subbands by using anisotropic spatial decompositions. Finally, we improve the entropy coding by designing a graph-cut solvable energy functional for the Lagrangian rate-distortion optimization problem.

Contents

Glossary

HD : High Definition.
4CIF : *Four CIF* – Resolution format made of 704×576 pixels.
CIF : *Common Interchange Format* – Resolution format made of 352×288 pixels.
QCIF : *Quarter CIF* – Resolution format made of 176×144 pixels.
DCT : *Discrete Cosine Transform*
Dyadic : Power of two.
EZBC : *Embedded Zeroblock coding based on Context modeling* – Image codec.
GOP/GOF : *Group of Frames* – Consecutive group of pictures/frames.
H.26X : Video coding algorithms normalized by ITU.
ISO : *International Organization for Standardization* – Standadization organism.
ITU : *International Telecommunications Union* – Standadization organism.
JPEG : *Joint Photographic Experts Group* – ISO expert group.
JPEG : Still image coding algorithm created by JPEG.
JPEG-2000 : Scalable still image coding algorithm created by JPEG.
JSVM : *Joint Scalable Video Model* – Reference software associated to SVC standardization.
Lifting : Invertible decomposition structure.
MC-EZBC : *Motion-Compensated EZBC* – Scalable video coding algorithm.
MCTF : *Motion Compensated Temporal Filtering*
MPEG : *Moving Picture Experts Group* – ISO expert group.
MPEG-X : Video coding algorithms normalized by MPEG.
PSNR : *Peak Signal Noise Ratio*
Wavelet : Function whose translations and dilatations provide a scalable representation of another function.
Scalable : Can be represented with different precision levels.
Subband : The result of a filter bank decomposition.
SVC : *Scalable Video Coding* – Scalable extension of H.264 standard.
Vidwav : Wavelet research group inside MPEG community (denote also the reference video coding algorithm developped by the Vidwav group).
YSNR : Signal to noise ratio of the luminance Y component of a frame in YUV format.

Synthèse des travaux exposés dans le manuscrit

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

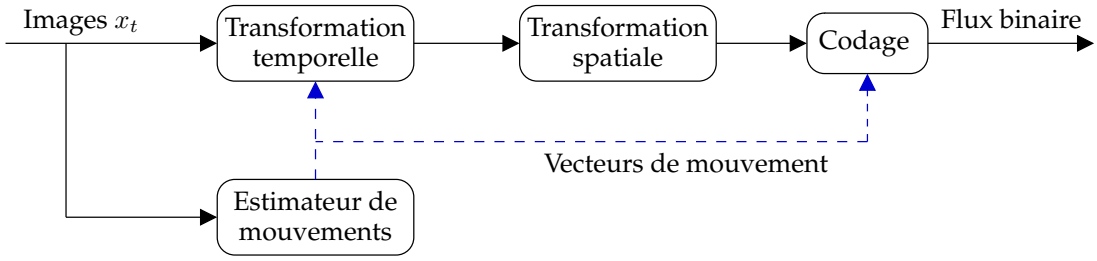


Figure 1: Schéma générique d'un encodeur vidéo $t + 2D$.

Grâce à la représentation multirésolution inhérente, les codeurs basés-ondelettes ont la capacité de fournir une description scalable d'un signal. Ceci représente la raison principale pour laquelle le choix du paradigme du codage *lifting* $t + 2D$ basé-ondelettes s'impose comme cadre conceptuel de développement pour les travaux dans cette thèse.

L'objectif de ces travaux consiste en l'analyse et la conception d'un système de codage vidéo scalable. Plus exactement, nous nous intéressons particulièrement à :

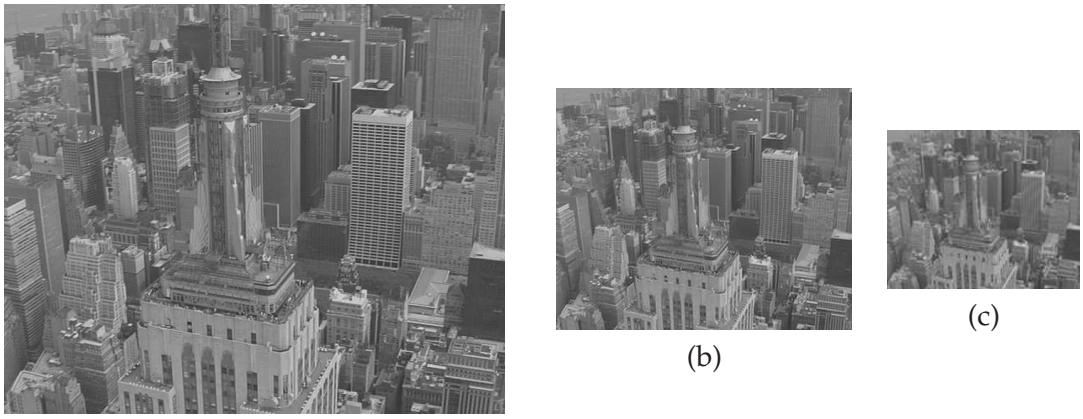
- ⋆ la construction et l'optimisation de nouvelles transformées temporelles compensées en mouvement, dans le but d'augmenter l'efficacité objective et subjective du codage;
- ⋆ une meilleure représentation des sous-bandes temporelles, en utilisant des décompositions spatiales anisotropes afin de capturer l'orientation spatiale de détails;
- ⋆ l'amélioration du codage entropique en concevant une solution basée sur la théorie des graphes afin d'optimiser la minimisation du Lagrangien débit-distorsion.

Cette thèse s'inscrit dans le développement d'un codec visuel 3D basé-ondelettes proposé par le groupe Vidwav dans le cadre MPEG [212]. Une partie de ce travail a été soutenue par le 6éme programme-cadre de la Commission Européenne dans le projet IST-FP6-507752 (Réseau d'Excellence MUSCLE).

Introduction aux représentations multirésolution

La scalabilité caractèrise le fait qu'un objet ou un signal soit représentable sur plusieurs niveaux de résolution ou de qualité. Une transformation sera ainsi dite scalable si elle est en mesure de représenter un signal sur plusieurs niveaux de résolution ou de qualité.

La notion de scalabilité est en fait très générale et il existe plusieurs types de scalabilité. Dans le cas d'un signal monodimensionnel, on parlera de scalabilité en résolution pour désigner le fait qu'un signal puisse être décrit par un nombre variable d'échantillons. Dans le cas d'une image, la scalabilité spatiale qualifie la propriété de pouvoir représenter une image sur plusieurs niveaux de résolution spatiale, comme illustré en Fig. 2.



(a) (b) (c)

Figure 2: Scalabilité spatiale. Exemples de facteurs de résolution dyadiques obtenus avec le codec scalable JPEG-2000.

Il est aussi possible de représenter un signal sur différents niveaux de qualité, où chaque échantillon ou coefficient peut être décrit avec une précision plus ou moins grande. On parlera dans ce cas de scalabilité en qualité. Il existe d'autres types de scalabilité : dans le cas d'une séquence vidéo, on parlera de scalabilité temporelle pour désigner la propriété de pouvoir la représenter à plusieurs cadences temporelles, exprimées en nombre d'images par seconde. D'autres types de scalabilité peuvent être définis comme la scalabilité en complexité ou la scalabilité en délai, scalabilité orientée-objet etc.

Pourquoi des représentations scalables?

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

Il existe de nombreux cas d'utilisation nécessitant une description scalable et parcimonieuse d'un contenu multimédia, relevant pour la plupart du domaine de l'adaptation de contenu. Par exemple, les images présentes sur Internet sont souvent disponibles sous deux voire trois résolutions (aperçu *thumbnail*, résolution moyenne et haute résolution) en fonction de la façon dont elles sont visualisées. De plus, il est souvent nécessaire de posséder un morceau de musique compressé à plusieurs débits, en fonction de la qualité désirée et de la place disponible. Enfin, les opérateurs commerciaux de diffusion

de contenus multimédia ont tout intérêt à utiliser un format scalable. Un opérateur de téléphonie mobile pourra ainsi diffuser un flux vidéo TV destiné à un parc hétérogène de récepteurs dont les écrans sont de tailles différentes.

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

L'analyse multirésolution et la transformée en ondelettes sont des outils mathématiques capables de fournir une telle représentation scalable d'un signal. Nous rappelons dans la section 1.2 les fondements mathématiques de ces outils. Il existe cependant d'autres transformations capables de fournir une représentation scalable d'un signal. La structure *lifting*, rappelée en section 1.3 permet d'étendre la théorie des ondelettes dans un cadre non-linéaire et autorise simplement la construction de transformées non-linéaires et inversibles.

Les travaux menés tout au long de cette thèse ont comme but la construction d'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 (voir chapitre 2).

Codage vidéo scalable

La majeure partie des codecs vidéos actuels, dont les célèbres MPEG-x 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. Les travaux sur les schémas de codage vidéo par ondelettes sont plus récents. Ces derniers sont intrinsèquement scalables et nous allons détailler la structure de codage la plus prometteuse : le schéma de codage $t+2D$, basé sur l'utilisation d'un filtrage temporel compensé en mouvement, et utilisé dans ces travaux de thèse.

Codage vidéo hybride scalable

Le schéma générique d'un encodeur vidéo hybride est donné en Fig. 3. 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'entrée x_t provenant d'une séquence vidéo sont lues et sont transformées suivant les étapes suivantes.

Estimation de mouvement Avant la transformation des images d'entrée, on procède à une estimation de mouvement. Cette estimation est généralement représentée par des champs de blocs de taille fixe ou variable, dont la précision peut être sous-pixellique. 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

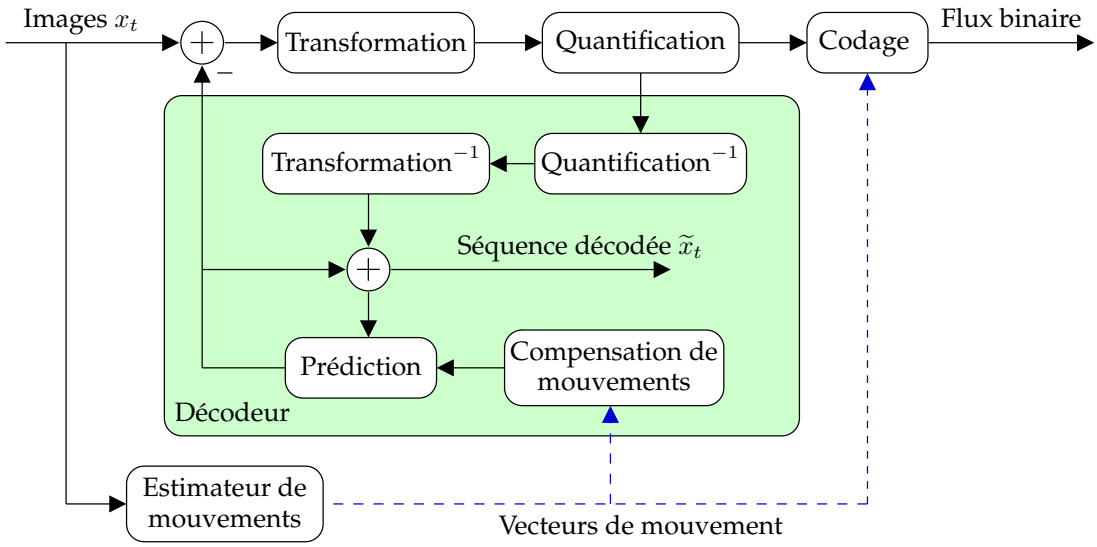


Figure 3: Schéma générique d'un encodeur vidéo hybride avec boucle de rétroaction.

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 codage des images dans une séquence vidéo. 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 (ou des) image(s) passée(s) et une (ou des) image(s) future(s), et sont encore plus concises. Les images d'une séquence vidéo sont généralement encodées par un motif de prédiction cyclique fixe, comme illustré en Fig 4.

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 les normes JPEG et MPEG ou ITU. Les coefficients résultants sont alors quantifiés par des tables, sous le contrôle d'un paramètre de qualité Q lié au pas de quantification.

Codage entropique Après quantification, les coefficients des images sont encodés en zig-zag, par un codeur de type RLE (*Run-Length Encoding*) et un codeur entropique, par example, codeur d'Huffman ou arithmétique. Les champs de mouvement sont quant à eux encodés sans perte au moyen de codes de longueur variable (VLC) (*Variable Length Coding*).

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 certaine résolution spatio-

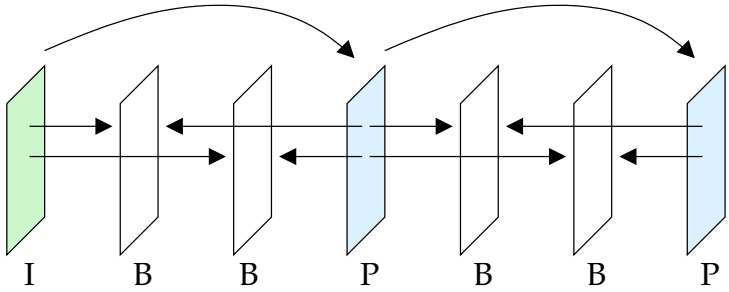


Figure 4: Agencement des modes de prédiction IBBPBBP d'un groupe d'images.

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 et moyens de calcul. De nombreuses stratégies ont cependant été mises au point [13, 140] pour diminuer sa complexité.

Codage vidéo scalable par ondelettes : schéma $t + 2D$

Parallèlement aux schémas de codage hybride, un nouveau paradigme de codage a été développé : le codage scalable par ondelettes. Il est basé sur deux technologies principales : le filtrage temporel compensé en mouvement (*Motion Compensated Temporal Filtering* - MCTF) et la transformation spatiale par ondelettes. Les schémas de compression basés-ondelettes sont devenus de plus en plus importants, un exemple étant le standard actuel de compression pour les images fixes JPEG2000 [8, 178].

En 1994, Taubman et Zakhor [177] ont proposé un schéma de codage vidéo par ondelettes dans lequel 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 [134] 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 et est à l'origine de nombreux travaux sur le codage vidéo par ondelettes. Nous décrirons dans la suite le principe général de ce schéma de codage et présenterons les filtres temporels les plus utilisés.

Principe général

Le principe du schéma de codage vidéo $t+2D$, illustré par la Fig. 1, repose sur l'utilisation d'un filtre temporel compensé en mouvement, 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.

En parallèle du traitement des images, un estimateur de mouvement est placé en amont du schéma et fournit les 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 2.2. Il n'y a ainsi pas de rétroaction d'un décodeur assujeti 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, une résolution spatiale et une 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 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 dans les 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.

Filtres temporels compensés en mouvement

Un des plus simples filtrages temporels est réalisé par le banc de filtres de Haar (Eq. 1). Les opérations de base pour obtenir les sous-bandes passe-haut et passe-bas dans la forme *lifting* sont les suivantes :

$$\begin{cases} H_t = x_{2t+1} - P(\left\{ x_{2(t-k)}, v_{2t+1}^{2(t-k)} \right\}_{k \in T_k^p}) \\ L_t = x_{2t} + U(\left\{ H_{t-k}, v_{2t}^{2(t-k)+1} \right\}_{k \in T_k^u}) \end{cases} \tag{1}$$

où v_i^j est le vecteur de mouvement utilisé pour prédire la trame courante i de la trame de référence j , T_k^p (respectivement T_k^u) étant le domaine de définition pour l'operateur de prediction (respectivement mise à jour) temporel.

Une vue d'ensemble des diverses structures de filtrage temporel compensé en mouvement utilisées dans le codage visuel scalable peut être trouvée dans [137], et, dans [150], quelques formulations intéressantes de ces décompositions temporelles en forme *lifting* sont présentées.

Le schéma *lifting* presenté dans Fig. 5 permet ainsi la construction de transformées temporelles plus longues, inversibles et dotées d'opérateurs bidirectionnels comme la transformée 5/3 compensée en mouvement, décrite dans Eq. 2.

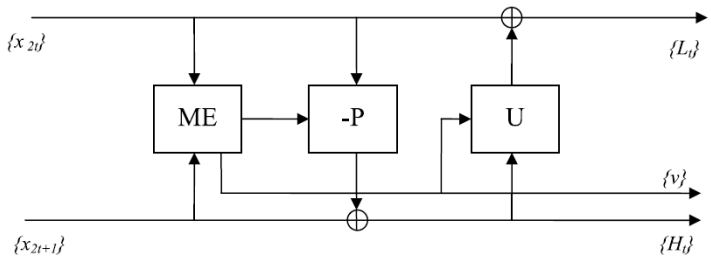


Figure 5: Structure *lifting* temporelle compensée en mouvement

$$\begin{cases} H_t(n) = x_{2t+1}(n) - \dfrac{1}{2}(x_{2t}(n - v_{2t+1}^{2t}) + x_{2t+2}(n - v_{2t+1}^{2t+2})) \\ L_t(p) = x_{2t}(p) + \dfrac{1}{4}(H_{t-1}(p + v_{2t}^{2t+1}) + H_t(p + v_{2t+2}^{2t+1})) \end{cases} \tag{2}$$

Les codeurs basés MCTF fournissent une flexibilité élevée, en terme de type de scalabilité, pour le flux binaire à travers différentes résolutions temporelles, spatiales et de qualité. En outre, elles fournissent une meilleure robustesse face aux erreurs que les codeurs hybrides conventionnels. En fait, les codeurs basés ondelettes peuvent mieux extraire l'information pertinente. Les sous-bandes temporelles passe-bas contiennent de l'information consistante qui permet de mieux exploiter la redondance temporelle, ce qui est non réalisable par les methodes hybrides classiques. Même si beaucoup d'éléments dans le cadre du filtrage temporel compensé en mouvement basé *lifting* peuvent être considérés comme des prolongements des techniques prouvées dans les codeurs hybrides, ce schéma de codage permet d'exposer un certain nombre d'options radicalement nouvelles dans le codage visuel. Cependant, quand une transformation par ondelettes est appliquée pour le codage des sous-bandes passe-bas et passe-haut résultant du processus de filtrage temporel, la ressemblance avec l'approche des méthodes de codage basé ondelette 2D est évidente. De nombreuses améliorations et optimisations ont ensuite été apportées pour améliorer l'efficacité du codage de ces structures : nos travaux s'inscrivent dans ces développements et sont détaillés dans les sections suivants.

Codage temporel

Comme cela a été dit dans la section précédente, les décompositions les plus utilisées pour la décorrélation temporelle sont dyadiques, c'est à-dire les bancs de filtres Haar et 5/3. En règle générale, le filtrage uniforme part de l'hypothèse que les trames sont fortement corrélées. Toutefois, cette hypothèse n'est plus vérifiée quand la vidéo rencontre des coupures de scène, comme dans le cas des films d'action, des vidéo clips etc. Dans ce cas, l'inefficacité de l'estimateur de mouvement conduit à une mauvaise prédiction/mise à jour, qui, combinée avec l'implantation avec une fenêtre glissante du filtrage temporel, conduit à la propagation des erreurs dans les niveaux de décomposition. Nous proposons alors un filtrage temporel compensé en mouvement modifié, capable de surmonter ce déficit en détectant et traitant les coupures de scènes qui peuvent survenir dans une séquence vidéo.

Traitement des coupures de scène dans le codage temporel compensé en mouvement

Une faiblesse des codecs vidéo $t + 2D$ est liée à la façon dont le filtrage temporel se comporte près des changements de scènes. Habituellement, la séquence vidéo est partitionnée en GOPs (*Group pf Pictures*) et filtrée temporellement sans vérifier la corrélation entre les trames du GOP. Lorsque le signal contient du mouvement de forte complexité et surtout des coupures de scène, cela peut se traduire par l'inefficacité de l'opération de prédiction/mise à jour, entraînant une dégradation de la qualité des résultats et également de la scalabilité temporelle.

Plusieurs tentatives pour éviter les artefacts liés à ces changements brusques ont déjà été proposées pour le codage hybride, comme la détection de scènes et l'échantillonnage basé sur le contenu de séquences vidéo [168] ou la segmentation en utilisant des données liées au coût d'encodage [61], qui améliorent les performances, mais ne résolvent pas complètement ce problème.

On a ainsi proposé de détécter les coupures de scènes, en utilisant un algorithme basé sur l'étude de la variation d'énergie du résidu de prédiction (*Displaced Frame Difference* - DFD), et d'encoder chaque série de trames entre deux scènes séparément, en adaptant le filtrage temporel pour pouvoir traiter un nombre arbitraire de trames dans une séquence vidéo. Si le résidu de prédiction entre deux trames consécutives est donné par :

$$d_t = DFD(x_t, x_{t+1}) = x_{t+1} - \mathcal{F}(x_t, \boldsymbol{v}_t)$$

alors la variation relative de l'énergie de la DFD est calculée comme :

$$\Delta_{2t} = \frac{d_{2t}^2}{d_{2t-1}^2}$$

où $\mathcal{F}$ est l'operateur de prédiction. Lorsque le signal d'entrée est fortement corrélé, la variation relative de l'énergie de la DFD tout au long de la séquence est presque constante (i.e., $\Delta \approx 1$). Nous affirmons qu'une coupure de scène est détectée lorsque la variation relative de l'énergie présente un changement rapide. Pour des paramètres τ_1 et τ_2 correctement choisis, nous affirmons que le changement de scène survient après la trame x_{2t+1} lorsque :

$$\begin{cases} |\Delta_{2t} - 1| < \tau_1 \\ |\Delta_{2t+1} - 1| > \tau_2 \end{cases}$$

On peut noter que tout autre algorithme de détection de changement de scène présents dans la littérature pourrait remplacer le critère DFD. Une fois la décision prise, nous passons à l'étape suivante, c'est à-dire l'encodage des trames précédant la coupure.

D'abord, le filtrage temporel doit être modifié afin de ne pas filtrer sur un changement de scène. Une modification est alors effectuée à l'encodage du dernier GOP avant la coupure de scène, où les opérateurs de prédiction et mise à jour doivent être modifiés près de la fin de la première scène, comme l'illustre la Fig. 6. Pour les séquences transformées temporellement de la meme manière, les sous-bandes résultant du codage MCTF sont codées par GOPs de 2^L trames, où L est le nombre de niveaux de décomposition temporelle. Quand un changement de scène se produit dans une séquence, le GOP juste avant le changement aura, en général, un nombre de trames different de 2^L. En notant

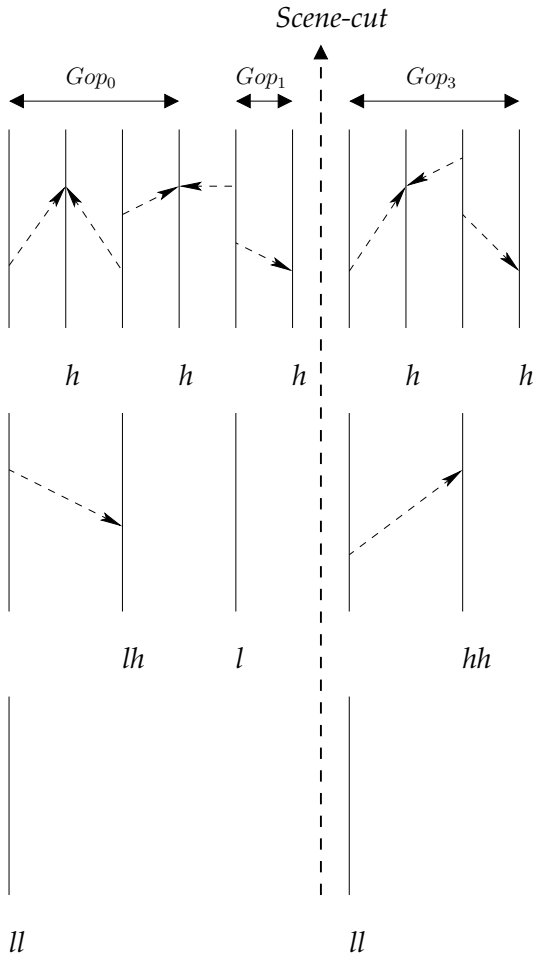


Figure 6: Traitement des coupures de scène sur 2 niveaux temporels.

A_n le nombre de ces trames et en exprimant ce nombre en représentation binaire :

$$A_n = (a_0 a_1 \dots a_{L-1})_2 = \sum_{l=0}^{L-1} a_l 2^l,$$

on décompose le GOP en petits GOPs, par ordre de taille décroissante : $a_l\, 2^l,$ où $l \in \{0, \dots, L-1\},$ et $a_l \in \{0, 1\}$, qui seront filtrés et codés séparément. Cela revient aussi à modifier le nombre de niveaux de décomposition temporelle et les opérations de *lifting* pour ces sous-GOPs. En fait, nous pouvons faire seulement l niveaux de filtrage temporel pour les sous-GOPs de taille 2^l, $l < L$. En outre, la prédiction à travers la coupure de scène n'est pas autorisée. Après le changement de scène, le filtrage normal avec fenêtre glissante (ou au fil de l'eau) est relancé.

Comme l'illustrent les résultats expérimentaux presentés dans Fig. 7, notre méthode donne un gain moyen d'environ 1.5 dB sur les séquences vidéo testées, et une meilleure qualité visuelle pour les trames proches de la coupure de scène.

En suivant les idées principales concernant l'efficacité du codage obtenue avec des filtres temporels plus longs proposés dans [218], nous passons à la présentation d'une nouvelle tranformée temporelle compensée en mouvement, spécialement adaptée pour

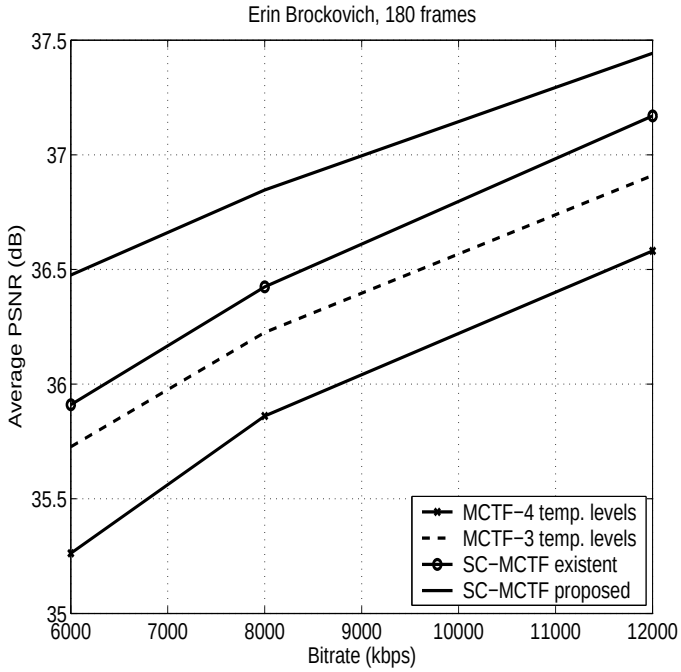


Figure 7: Performance débit-distorsion pour la séquence Erin Brockovich (HD 1920×1280, 60Hz) (*SC* dénote l'encodage avec traitement des changement de scène et *SC-MCTF existent* dénote les resultats obtenues avec la mèthode proposée dans [209, 38]).

le codage des séquences de faibles mouvements et ayant une application directe dans la vidéosurveillance.

Schéma 5-bandes compensé en mouvement

L'implantation *lifting* des codeurs basé ondelettes [173, 99] assure une complexité faible qui fait d'elle l'approche la plus largement adopté pour le codage $t+2D$ dans la littérature [44, 134, 94]. Comme mentionné précédemment, les schémas *lifting* MCTF ont été soumis à de nombreuses optimisations et améliorations, concernant, par exemple, les opérateurs de prédiction/mise à jour [196, 145, 143, 144, 184] ou la précision de l'estimation de mouvement [82, 94]. De plus, des schémas *lifting* M-bandes avec reconstruction parfaite [41, 89] ou des décompositions temporelles 3-bandes [181, 180] ont été proposées, permettant des facteurs de scalabilité flexibles, non-dyadiques.

L'intérêt pour plusieurs canaux peut être double. D'abord, pour permettre une liberté complète dans le choix du facteur de scalabilité (par exemple, permettant un échantillonnage temporel avec des facteurs 5, pourquoi pas 7, et des combinaisons de ces facteurs). Deuxièmement, cela permet la création des sous-bandes d'approximation en utilisant un nombre réduit de décompositions temporelles.

Nous avons ainsi proposé un schéma *lifting* compensé en mouvement qui permet un facteur de 5 en scalabilité temporelle. Selon les caractéristiques de la séquence vidéo, le modèle de mouvement etc., cette structure peut fournir un codage plus performant. En outre, selon la scalabilité temporelle désirée, cette structure particulière pourrait être plus efficace. Elle permet une meilleure représentation des sous-bandes temporelles d'approximation, et ainsi une meilleure scalabilité temporelle est obtenue.

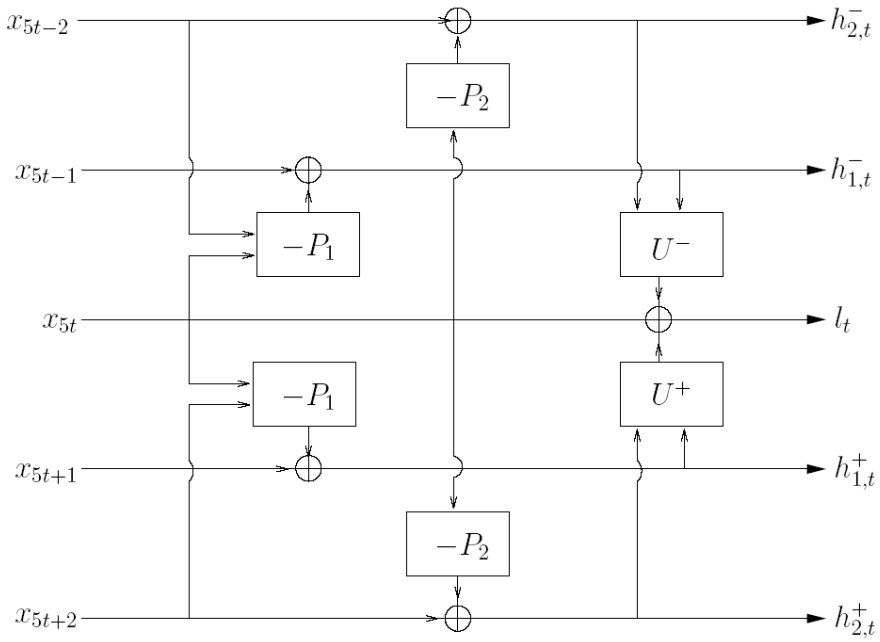


Figure 8: Schèma *lifting* 5-bandes avec opérateurs quelconques.

Par exemple, la structure *lifting*, donc inversible, du schèma 5-bandes entraîne une sous-bande d'approximation et 4 sous-bandes de détail (voir Fig. 8).

Les équations décrivant l'analyse sont :

$$\begin{cases} h^-_{1,t} = x_{5t-1} - P^-_1(\{x_{5t-2}, x_{5t}\}_t) \\ h^+_{1,t} = x_{5t+1} - P^+_1(\{(x_{5t+2}, x_{5t}\}_t) \\ h^-_{2,t} = x_{5t-2} - P^-_2(\{x_{5t}\}_t) \\ h^+_{2,t} = x_{5t+2} - P^+_2(\{x_{5t}\}_t) \\ \quad l_t = x_{5t} + U^-(\left\{h^-_{1,t}, h^-_{2,t}\right\}_t) + U^+(\left\{h^+_{1,t}, h^+_{2,t}\right\}_t). \end{cases}$$

Comme on peut remarquer, il y a quatre opérateurs de prédiction, P^-_1 / P^+_1 et P^-_2 / P^+_2, ainsi que deux opérateurs de mise à jour, U^- et U^+, utilisés pour obtenir les sous-bandes temporelles. En raison de la symétrie du schèma, nous avons proposé d'utiliser des opérateurs de prédiction symétriques pour générer les sous-bandes temporelles de détail $h^-_{1,t}/h^+_{1,t}$ et $h^-_{2,t}/h^+_{2,t}$:

$$\begin{aligned} P^-_1(\{x_{5t-2}, x_{5t}\}_t) &= \alpha x_{5t-2} + (1-\alpha)x_{5t}, \\ P^+_1(\{(x_{5t+2}, x_{5t}\}_t) &= \alpha x_{5t+2} + (1-\alpha)x_{5t}, \\ P^-_2(\{x_{5t}\}_t) &= \beta x_{5t} - (1-\beta)x_{5t-5} \\ P^+_2(\{x_{5t}\}_t) &= \beta x_{5t} - (1-\beta)x_{5t+5}. \end{aligned}$$

Dans un premier cas, nous avons consideré les opérateurs de prédiction les plus simples pour les trames encadrant le GOP, c'est à-dire des opérateurs identité pour obtenir

$h^-_{2,t}/h^+_{2,t}$ et des opérateurs de prédiction bidirectionels pour les autres sous-bandes de détail :

$$\begin{cases} h^-_{1,t} = x_{5t-1} - \frac{1}{2}(x_{5t-2} + x_{5t}) \\ h^+_{1,t} = x_{5t+1} - \frac{1}{2}(x_{5t+2} + x_{5t}) \\ h^-_{2,t} = x_{5t-2} - x_{5t} \\ h^+_{2,t} = x_{5t+2} - x_{5t}, \end{cases}$$

La sous-bande d'approximation est alors obtenue par :

$$l_t = (1-\gamma-2\delta)x_{5t} + \gamma x_{5t-1} + \gamma x_{5t+1} + (\delta - \frac{\gamma}{2})x_{5t-2} + (\delta - \frac{\gamma}{2})x_{5t+2},$$

où δ et γ sont des paramètres dans l'intervalle $(0,1)$ qui peuvent être adaptés afin de garantir l'existence du filtre passe-bas, c'est à dire $L(-1) = 0$, où L est la transformée en z de l_t.

Dans un deuxième cas, nous avons consideré une implantation avec une fenêtre glissante (ou *au fil de l'eau*) pour la structure 5-bandes, c'est à-dire avec des opérateurs de prédiction bidirectionnels pour toutes les trames de détail :

$$\begin{cases} h^-_{1,t} = x_{5t-1} - \frac{1}{2}(x_{5t-2} + x_{5t}) \\ h^+_{1,t} = x_{5t+1} - \frac{1}{2}(x_{5t+2} + x_{5t}) \\ h^-_{2,t} = x_{5t-2} - \frac{1}{2}(x_{5t} + x_{5t-5}) \\ h^+_{2,t} = x_{5t+2} - \frac{1}{2}(x_{5t} + x_{5t+5}), \end{cases}$$

où les valeurs $\alpha = \frac{1}{2}$ et $\beta = \frac{1}{2}$ correspondent ici au filtre passe-haut le plus selectif. La sous-bande passe-bas est alors obtenue comme :

$$l_t = (1-\gamma-\delta)x_{5t} + \gamma x_{5t-1} + \gamma x_{5t+1} + (\delta - \frac{\gamma}{2})x_{5t-2} + (\delta - \frac{\gamma}{2})x_{5t+2} - \frac{\delta}{2}x_{5t-5} - \frac{\delta}{2}x_{5t+5}.$$

La quantification étant appliquée de manière identique sur toutes les sous-bandes, on a renormalisé les différentes sous-bandes temporelles pour être aussi proche que possible de la situation orthonormale. Les filtres normalisés ont été obtenus de la manière suivante :

$$\begin{gathered} \widehat{l_t} = k_l l_t \\ \widehat{h}^-_{1,t} = k_{h1} h^-_{1,t}, \quad \widehat{h}^+_{1,t} = k_{h1} h^+_{1,t} \\ \widehat{h}^-_{2,t} = k_{h2} h^-_{2,t}, \quad \widehat{h}^+_{2,t} = k_{h2} h^+_{2,t} \end{gathered}$$

Remarquons que nous considérons la même normalisation pour $h^-_{1,t}/h^+_{1,t}$ et $h^-_{2,t}/h^+_{2,t}$, ceci provenant de la symétrie des schémas de prédiction. Deux approches ont été considerées pertinentes pour obtenir les constantes de normalisation pour les deux implantations du schèma 5-bandes. D'un côté, nous avons cherché à préserver la norme unitaire des réponses impulsionnelles des filtres intervenant dans la structure 5-bandes. D'un autre côté, on veut préserver l'énergie de la séquence d'entrée. En particulier, si nous considérons l'erreur de quantification de chaque sous-bande d'approximation et de détail

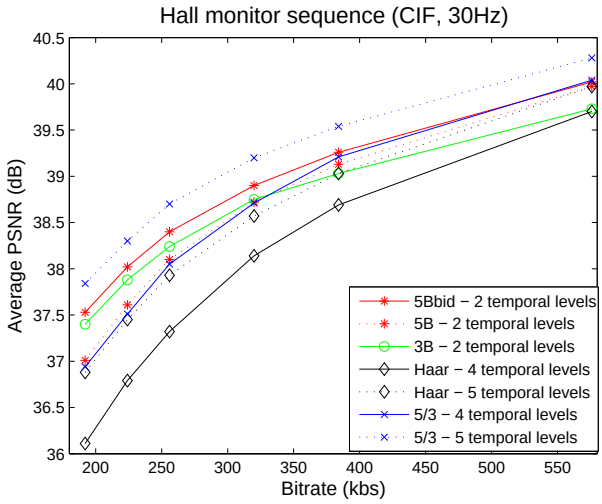


Figure 9: Courbes de débit-distorsion obtenues pour Hall monitor(CIF, 30Hz).

comme des variables indépendantes et identiquement distribuées, la somme des erreurs de reconstruction de cinq trames consécutives doit être égale à la somme des erreurs de quantification d'une sous-bande d'approximation et de quatre sous-bandes de détail :

$$\sigma^2_{x_{5t-2}} + \sigma^2_{x_{5t-1}} + \sigma^2_{x_{5t}} + \sigma^2_{x_{5t+1}} + \sigma^2_{x_{5t+2}} = \sigma^2_{\check{l}_t} + \sigma^2_{\widehat{h}^+_{1,t}} + \sigma^2_{\widehat{h}^-_{1,t}} + \sigma^2_{\widehat{h}^+_{2,t}} + \sigma^2_{\widehat{h}^-_{2,t}},$$

où σ^2_a designe la variance de la trame a.

Comme illustré dans Fig. 9 et Fig. 10, le schéma proposé a des résultats semblables aux bancs de filtres dyadiques de Haar et 5/3 et à la décomposition temporelle 3-bandes. En outre, il offre une meilleure efficacité de codage pour les sous-bandes temporelles, conduisant alors à une meilleure scalabilité temporelle. Le schéma 5-bandes peut être utilisé avec succès dans certaines applications, comme le codage des séquences de vidéosurveillance, où l'activité est faible dans la plupart des cas.

Toutes les transformations temporelles proposées jusqu'à présent sont fondées sur une approche linéaire du schéma *lifting*. Cependant, quand une séquence présente des transitions complexes, cette hypothèse de linéarité ne se vérifie plus. Dans la suite, nous allons présenter un schéma *lifting* de prédiction temporelle adaptative qui vise à atténuer ce problème.

Schéma de prédiction adaptative pour le codage vidéo scalable

L'étape clé pour réduire la redondance temporelle est l'estimation de mouvement qui se fait, généralement, par bloc. Même si une prédiction bidirectionnelle est appliquée, ou des algorithmes puissants comme *Hierarchical Variable Size Block Matching* (HVSBM) [44] sont utilisés, les effets de bloc sont encore présents. En outre, des effects de *ringing* sont lisibles à bas débit et des artefacts peuvent être présents aussi bien dans les sous-bandes passe-bas.

Afin d'éviter ces artefacts, des solutions ont été proposées pour la compensation en mouvement, comme une mise à jour par moyenne pondérée [184] ou compensation en mouvement par chevauchement des blocs [211]. Dans la suite nous proposons

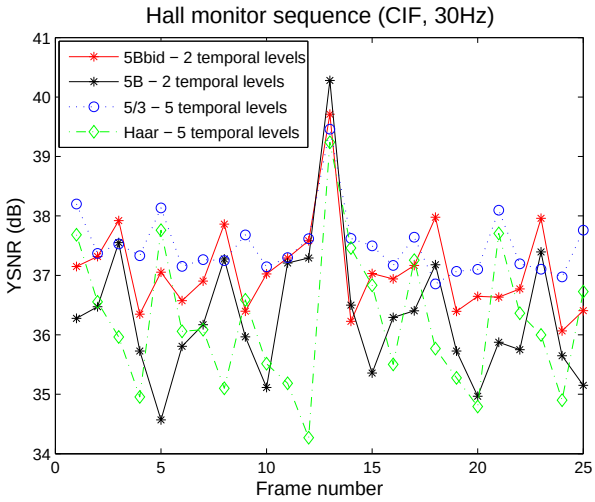


Figure 10: Variation de YSNR pour les trames 26-50 de Hall monitor (CIF, 30Hz) à 192 kbs.

d'améliorer la prédiction des sous-bandes temporelles passe-haut en utilisant un banc de filtres adaptatif.

Il y a divers filtres pour effectuer une decomposition adaptative des sous-bandes [206, 92, 20]. Nous utilisons le filtre adaptatif à réponse impulsionelle finie basé sur les moindres carrés (LMS) proposé dans [79]. Dans [79], le schéma d'adaptation est proposé pour la compression d'images 2D. Dans la suite, nous proposons d'étendre la méthode pour la prédiction temporelle compensée en mouvement dans un schéma de codage vidéo $t + 2D$.

Pour la trame x_{2t+1}, un estimateur à réponse impulsionelle finie peut être conçu en utilisant pour la prédiction un ensemble de pixels dans les trames voisines temporellement x_{2t} et x_{2t+2} (à noter qu'aucune compensation en mouvement n'est impliquée à ce point dans la prédiction) :

$$\hat{x}_{2t+1}(\mathbf{n}) = \sum_{\mathbf{k}\in\mathcal{S}} w_{2t,\mathbf{n},\mathbf{k}} x_{2t}(\mathbf{n}-\mathbf{k}) + \sum_{\mathbf{k}'\in\mathcal{S}'} w_{2t+2,\mathbf{n},\mathbf{k}} x_{2t+2}(\mathbf{n}-\mathbf{k}') \tag{3}$$

où les coefficients du filtre, w, sont trouvés de manière adaptative en utilisant l'algorithme LMS [53]. Dans l'Eq. (3), les sommes sont menées dans le voisinage approprié, $\mathcal{S}$ et $\mathcal{S}'$, dans les trames $2t$ et $(2t+2)$, respectivement. L'estimateur adaptatif pour la sous-bande de détail h_t est illustré dans Fig. 11.

En considerant une compensation en mouvement, l'Eq. (3) devient:

$$\hat{x}_{2t+1}(\mathbf{n}) = \sum_{\mathbf{k}} w_{2t,\mathbf{n},\mathbf{k}} x_{2t}(\mathbf{n}-\mathbf{k}-\mathbf{v}_t^+(\mathbf{n})) + \sum_{\mathbf{k}} w_{2t+2,\mathbf{n},\mathbf{k}} x_{2t+2}(\mathbf{n}-\mathbf{k}-\mathbf{v}_t^-(\mathbf{n})) \tag{4}$$

Une grande souplesse pour le schéma d'adaptation est atteint par la variation du nombre de pixels sélectionnés pour l'adaptation, comme l'illustre Fig. 12. Les résultats expérimentaux illustrés dans Fig. 13 montrent que même pour une adaptation avec deux pixels, la qualité objective donnée par le PSNR des trames reconstruites est améliorée. Un compromis entre l'efficacité de compression et la complexité supplémentaire venant d'une plus grande fenêtre d'adaptation peut être réalisé, selon l'application cible. Des

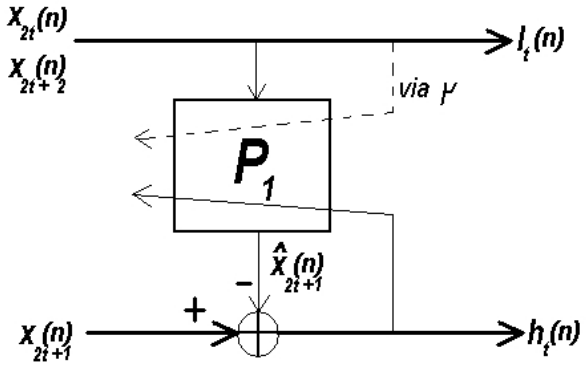


Figure 11: Estimateur adaptatif.

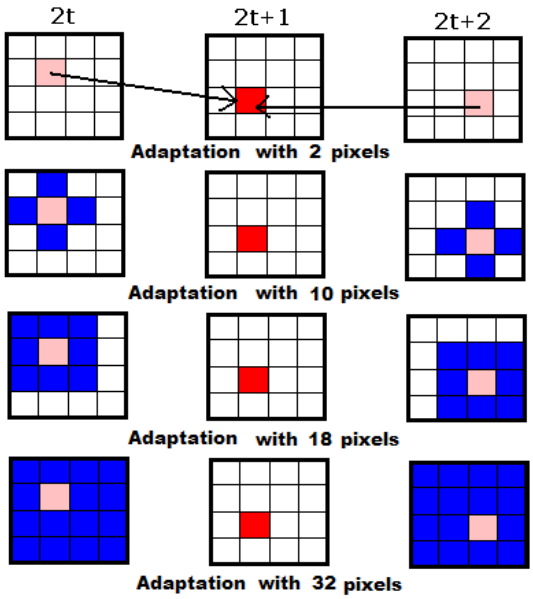


Figure 12: Schéma d'adaptation avec 2, 10, 18 et 32 pixels.

améliorations du PSNR ont été obtenues pour les séquences avec un contraste élevé entre divers segments dans la séquence et des conditions d'illumination variées.

Codage vidéo pour les séquences multitemporelles et multispectrales

On a vu dans les sections précédentes que les schémas de codage basé *lifting* sont très efficaces dans la compression des images [8] et de la vidéo [211]. Maintenant nous allons concentrer nos intérêts sur un autre type de données, les séquences satellitaires multispectrales et multitemporelles, comme actuellement, des capteurs optiques couvrent souvent une région localisée plusieurs fois par an. Ainsi, il est indispensable de fournir des outils qui soient capables de compresser ces séquences multispectrales et multitemporelles afin de stocker de grandes quantités de données. La compression des images hyperspectrales et multispectrales a été longement étudiée en utilisant plusieurs techniques, telles que le codage par ondelettes [21, 76], adaptatif [10], DCT [85], DPCM [51, 159] ou par modèles statistiques [66].

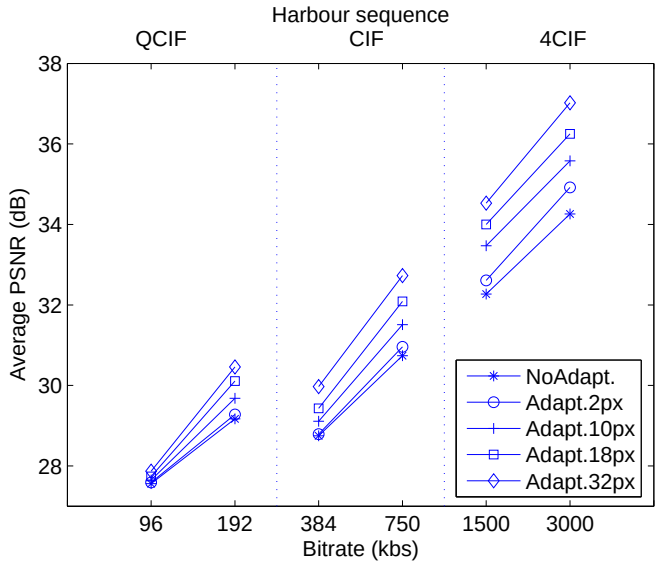


Figure 13: Courbes de débit-distorsion obtenues pour la séquence Harbour (4CIF, 60 Hz).

Comme dans le cas de la compression vidéo, où on a réduit la redondance dans le domaine temporel et spatial, les techniques de codage multispectral profitent de la présence de deux sources de redondance : la corrélation spatiale entre les pixels voisins dans la même bande spectrale et la corrélation spectrale entre les différentes bandes dans la même localisation spatiale. Plusieurs stratégies peuvent alors être adoptées pour exploiter séparément ou conjointement ces sources de redondance. La transformée de Karhunen-Loève (KLT) est généralement préférée comme une première étape, en raison de sa capacité de décorrelation, tandis que pour la deuxième étape, les transformées par ondelletes ou DCT ont été largement étudiées. En particulier, l'état-de-l'art JPEG2000 [8] peut être appliqué avec succès au codage intra-bande (spectrale), car il peut fortement compresser les images générées par la KLT avec une distorsion visuelle négligeable.

On a évalué la performance du codeur d'images JPEG2000 [8] et d'un codeur vidéo $t+2D$ basé ondelettes [210] pour la compression des séquences multispectrales et multitemporelles aquises par SPOT 1, 2 et 4, où chaque image est composée de 3 bandes spectrales. Une séquence a été construite avec des images de taille 3000×2000 pixels. Pour l'approche par codage vidéo, nous avons effectué une decomposition $t+2D$ dans le domaine spatio-temporel, suivi par une décorrelation KLT dans le domaine spectral. A noter que dans la dernière, même si aucun mouvement est estimé, l'algorithme de codage exploite la nature 3D de la décomposition, tant dans la répartition du débit, qu'au cours du codage entropique. Pour les expériences, nous avons utilisé pour le filtrage temporel le banc de filtres biorthogonal 5/3 et la DCT, pour leur efficacité de codage. Chaque pixel est codé sur 10 bits; par conséquent, la mesure de distorsion utilisée pour présenter les résultats est légèrement différente de celle utilisée dans la compression d'images ou vidéo, et on l'a défini comme suit :

$$PSNR = 10 \log_{10} \left(\frac{1024^2}{\text{MSE}} \right),$$

où MSE est l'erreur quadratique moyenne (*Mean Square Error*).

Le codage intra avec EZBC surpasse EBCOT en termes de gain de codage dans le cas

Débit (bpp)	JPEG2000 + WT	JPEG2000 + DCT	t+2D (EZBC)
PSNR (dB)	Lossless	> 70	> 60
Bande 1	3.22	3.78	2.46
Bande 2	3.55	4.07	2.84
Bande 3	4.82	4.42	4.42
Total	11.59	12.27	9.72

Table 1: Taux de compression avec une décorrelation temporelle.

de la compression sans perte. En outre, il a été mis en évidence que les décorrelations temporelles et spectrales peuvent être exploitées séparément ou conjointement afin d'augmenter l'éfficacité de compression. Toutefois, le progrès réalisé par ces méthodes de décorrelation ne diminue pas sensiblement le débit nécessaire pour encoder sans perte ces séquences.

Codage spatial

Dans un schéma de codage par ondelettes $t + 2D$, après la réduction de redondance temporelle par des techniques du type MCTF, les corrélations spatiales seront exploitées par des transformées en ondelettes. Les schémas de codage fondés sur les décompositions par ondelettes ont prouvé leur efficacité pour le codage spatial et ont été utilisées, par exemple, dans le codeur d'images JPEG2000 [8]. L'application des transformées en ondelettes pour le traitement spatial des sous-bandes temporelles est le plus souvent fondée sur une décomposition séparable [125].

Les lignes et les colonnes dans les sous bandes temporelles sont traitées indépendamment, et les fonctions de la base sont simplement des produits tensoriels des fonctions unidimensionnelles correspondantes. Ces décompositions sont simples à construire et ont une faible complexité de calcul, mais elles ne peuvent pas capturer toutes les propriétés géometriques d'une trame texturée.

L'inefficacité des transformées en ondelettes $2D$ classiques réside principalement dans leur isotropie spatiale. Cette isotropie se traduit par des opérations de filtrage et de sous-échantillonage effectuées de la même manière selon les directions verticales et horizontales à chaque échelle. En conséquence, les filtres obtenus par produit tensoriel de filtres $1D$ sont isotropiques à chaque échelle.

Selon une étude concernant la caractérisation des fréquences des sous- bandes temporelles, les spectres fréquentiels des détails temporels suivent une distribution approximativement uniforme, contrairement aux sous-bandes d'approximation. Comme le filtrage MCTF produit un nombre significatif de trames de détails en comparaison du nombre total de trames, une représentation parcimonieuse est importante pour l'efficacité globale du schéma de codage. Les paquets d'ondelettes constituent une solution possible pour la décorrelation des sous-bandes temporelles de détail. Cependant, le fait d'avoir une décomposition différente pour chaque trame peut être très coûteux lors de l'encodage, et également pour utiliser la décomposition dans le flux binaire compressé.

Nous proposons alors de trouver une représentation conjointe par paquets d'ondelettes pour un ensemble de plusieurs trames. De plus, nous avons créé un algorithme peu complexe pour le calcul de la meilleure base pour les décompositions biorthogonales.

Représentation conjointe par paquets d'ondelettes pour le codage vidéo

Une faiblesse du codage vidéo basé MCTF provient du filtrage spatial. En effet, la plupart des schémas de codage $t+2D$ ne distinguent pas les caractéristiques des approximations temporelles de celles des trames de détails et utilisent dans les deux cas une décomposition dyadique. Il a été montré que le banc de filtres 9/7 donne de bons résultats pour le codage d'images [19, 8], alors il paraît naturel de l'utiliser pour la décomposition spatiale des trames d'approximation. A cause de la grande quantité de fréquences hautes et intermédiaires, les trames de détails ne sont pas adaptées pour être décomposées par ondelettes.

Dans le paradigme du codage vidéo en sous-bandes, la proportion des trames de détails en rapport avec le nombre total de trames est significative. Par conséquent, une représentation spatiale efficace et parcimonieuse de ces trames est primordiale.

Les paquets d'ondelettes généralisent la décomposition dyadique utilisée dans les schémas de codage classiques en itérant les décompositions sur les sous-bandes de détails. Pour les images fortement texturées, ce type de décompositions adaptatives augmente les performances débit-distorsion. L'idée d'utiliser des paquets d'ondelettes $3D$ pour la compression vidéo a été développée dans des travaux de K. Ramchandran *et al.* [156] et T. Schell [162]. Un codeur vidéo hybride utilisant des décompositions par paquets d'ondelettes a été également proposé par Cheng, Li et Kuo [42], où les paquets sont utilisés pour décomposer la DFD, dont les caractéristiques sont proches de celles obtenues dans une décomposition temporelle. Nous proposons dans cette thèse une représentation *conjointe* pour plusieurs images.

Comme dans le cas des images individuelles, le critère entropique de Wickerhauser [208] peut être utilisé pour la sélection de la base de décomposition en paquets d'ondelettes d'un ensemble de trames, grâce à sa propriété d'additivité. Pour un groupe de trames $\mathcal{S}_j$, il faut d'abord calculer l'entropie de la représentation dans l'ensemble des bases du GOP. Dans le cas orthonormal, l'entropie associée peut simplement être trouvée par :

$$\mathcal{E}(\mathcal{S}_j) = \sum_{i=1}^{n_j} \mathcal{E}(f_{i,j}) \tag{5}$$

où $f_{i,j}$, $i \in \{1 \dots n_j\}$ représentent les n_j trames dans le GOP. Dans le cas biorthogonal, on peut profiter des valeurs de l'entropie déjà calculées afin d'obtenir une formule récursive pour le critère correspondant à l'union des bases :

$$\mathcal{E}(s, \bigcup_{j'>j,m'} \mathcal{B}^*_{j',m'}) = \frac{\sum_{j',m'} \mathcal{E}(s, \mathcal{B}^*_{j',m'})}{\sum_{j',m'} \mathcal{N}_{j',m'}} + \ln(\sum_{j',m'} \mathcal{N}_{j',m'}) \tag{6}$$

où $\mathcal{N}_{j',m'}$ est l'énergie des coefficients de la base $\mathcal{B}^*_{j',m'}$.

Une fois que l'entropie correspondant à $\mathcal{S}_j$ est obtenue, un algorithme *bottom-up* est utilisé pour la sélection de la meilleure base de décomposition [49]. Dans le cas dyadique, au niveau j, on compare le coût associé à la meilleure base du nœud courant, $\mathcal{B}_{j,m}$, avec celui de ses successeurs, $\mathcal{Q}(s, \mathcal{B}^*_{j+1,2m})$ et $\mathcal{Q}(s, \mathcal{B}^*_{j+1,2m+1})$. Comme la fonction de coût est considérée additive, si :

$$\mathcal{Q}(s, \mathcal{B}_{j,m}) \leq \mathcal{Q}(s, \mathcal{B}^*_{j+1,2m}) + \mathcal{Q}(s, \mathcal{B}^*_{j+1,2m+1}), \tag{7}$$

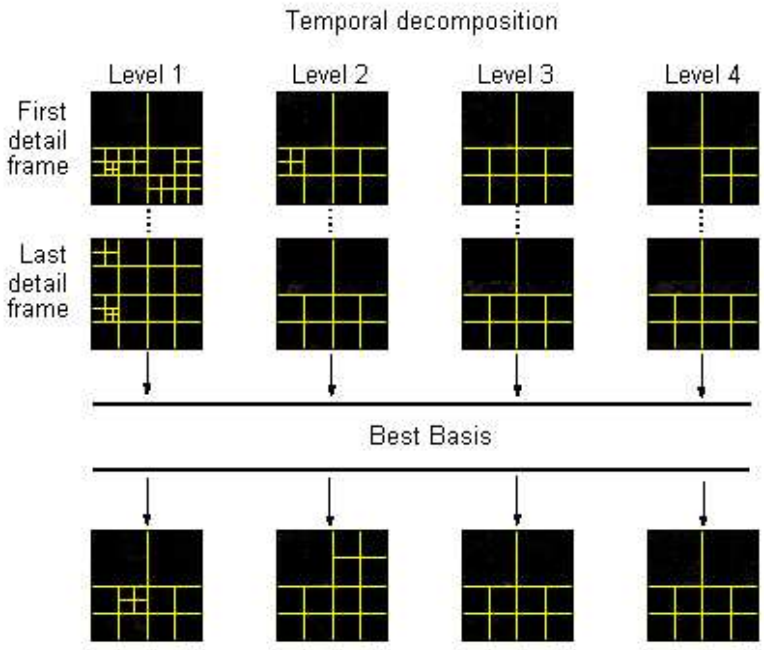


Figure 14: Décomposition conjointe en paquets d'ondelettes par sous-bande temporelle.

YSNR (dB) for Harbour (4CIF, 60Hz) sequence					
Bitrate (kbs)	1536	1780	2048	2560	3072
5-lev. dyadic Wavelet	30.834	31.228	31.726	32.489	33.035
Ortho.WP/TS	31.068	31.474	31.968	32.718	33.296
Ortho.WP/GOP	**31.111**	**31.476**	**31.969**	**32.733**	**33.312**
Biorthog.WP/TS	31.220	31.732	32.191	32.803	33.381
Biorthog.WP/GOP	**31.303**	**31.941**	**32.235**	**32.836**	**33.448**

Table 2: Tableau débit-distorsion pour la séquence Harbour(4CIF, 60Hz).

on peut alors conclure que la meilleure base pour le nœud (j, m) est $\mathcal{B}^*_{j,m} = \mathcal{B}_{j,m}$, et donc les nœuds $(j+1, 2m)$ et $(j+1, 2m+1)$ seront supprimés dans l'arbre de décomposition. En revanche, ils sont gardés dans cet arbre si l'Eq. (7) n'est pas vérifiée.

Nous avons proposé deux possibilités pour utiliser les paquets d'ondelettes dans la décomposition spatiale de sous-bandes temporelles de détail. La première approche est d'utiliser une décomposition conjointe par paquets d'ondelettes pour toutes les trames appartenant à un niveau de décomposition temporelle donné (comme illustré dans Fig. 14). Cette méthode provient de l'hypothèse que les trames appartenant à un certain niveau de décomposition ont, plus ou moins, les mêmes caractéristiques fréquentielles. La deuxième approche est d'avoir une description conjointe en paquets d'ondelettes pour chaque unité de codage ; c'est à dire, une base unique pour toutes les trames de détail dans un GOP (comme il peut être vu dans Fig. 15).

Il peut être facilement remarqué (Tab. 2) que dans tous les cas la décomposition des trames temporelles de détail en utilisant les paquets d'ondelettes donne des meilleurs résultats, en ayant un gain entre 0.1 et 0.9 dB sur le filtrage dyadique classique. En outre, on peut observer que l'utilisation d'une décomposition biorthogonale par paquets

d'ondelettes améliore légèrement les résultats par rapport à la base orthogonale (un gain moyen autour de 0.1 dB). Les petites différences entre les résultats obtenus avec la base de paquets d'ondelettes appliquée sur chaque niveau de sous-bandes temporelles et sur chaque groupe de trames dans un GOP peuvent être expliquées par le débit supplémentaire nécessaire dans le premier cas pour décrire l'arbre de décomposition à chaque niveau.

Toutefois, trouver la meilleure représentation en paquets d'ondelettes, même pour une seule base par groupe de trames, peut être une tâche coûteuse en complexité. Les décompositions spatiales en ondelettes et paquets d'ondelettes entièrement séparables qui seront présentées dans la section suivante préservent la simplicité de la transformée ondelette dyadique classique et fournissent une meilleure représentation des discontinuités dans les sous-bandes temporelles de détail.

L'application de la transformée en ondelettes et paquets d'ondelettes entièrement séparables dans le codage vidéo

Beaucoup de décompositions anisotropes ont été proposées comme solution au problème du codage des textures qui ont une répartition homogène d'énergie, comme les bandelettes [114], les wedgelettes [64], les curvelettes [63], les contourlettes [62] etc. Cependant, l'implantation de telles transformées nécessite souvent un suréchantillonage et a une plus grande complexité comparée aux transformées en ondelettes classiques. Nous avons proposé de séparer complétement les transformées verticales et horizontales de la transformée en ondelette classique, en obtenant donc un représentation fréquentielle fortement anisotrope qui permet une reconstruction parfaite.

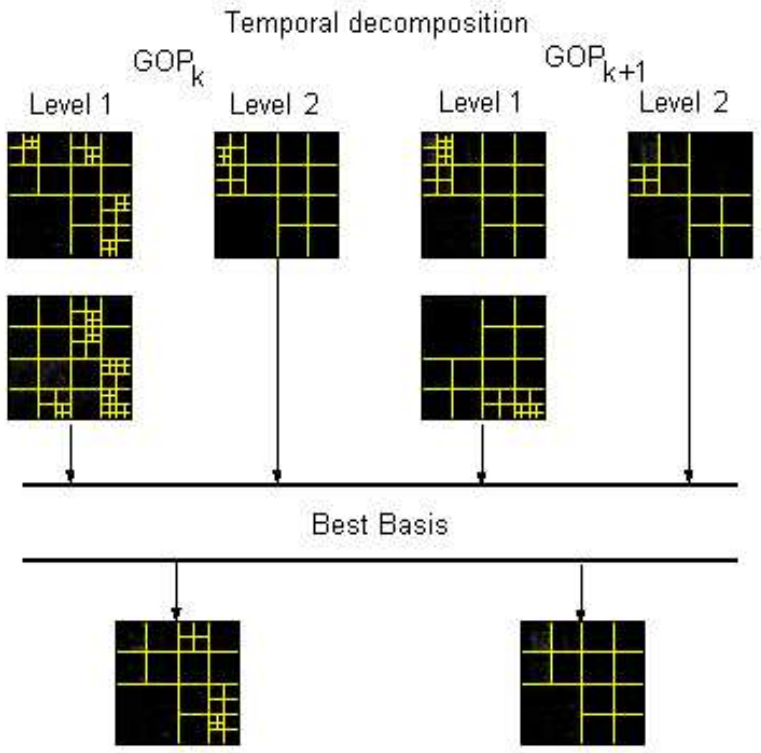


Figure 15: Décomposition conjointe en paquets d'ondelettes par GOP.

En règle générale, dans le codage des images, une décomposition 2D isotrope est utilisée. Cela résulte du produit tensoriel à chaque niveau de la décomposition de bases 1D ayant la forme :

$$\{\phi_{J,k}(t),\ k \in \mathbb{Z}\} \cup \bigcup_{j \leq J} \{\psi_{j,k}(t),\ k \in \mathbb{Z}\}\ , \tag{8}$$

où J est le nombre maximal de niveaux de décomposition. Cette alternance entre les décompositions horizontales et verticales à chaque niveau entraîne des sous-bandes *carrées*, c'est à dire la décomposition de Mallat [125]. Ainsi, pour une sous-bande, le nombre de niveaux de décomposition dans la direction horizontale est le même que le nombre de niveaux de décomposition dans la direction verticale. Ce processus est justifiée par les propriétés naturelles des images réelles : les caractéristiques de leur texture sont souvent très semblables dans toutes les directions.

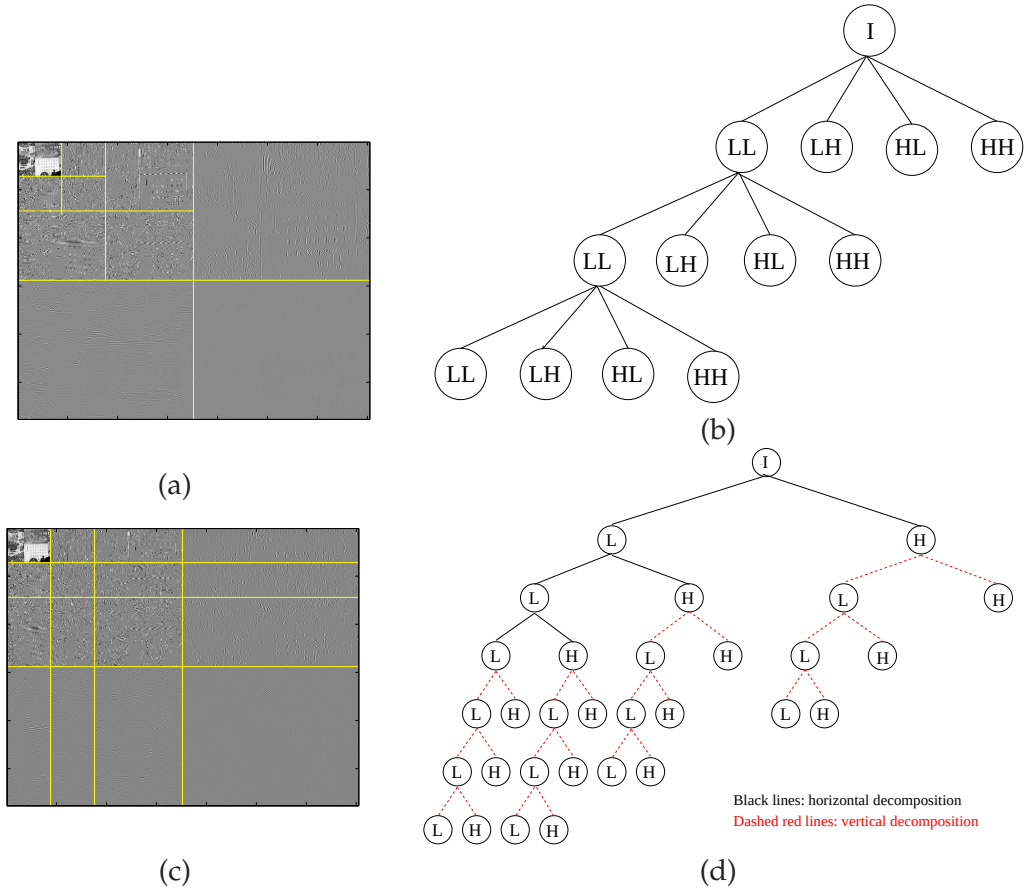


Figure 16: Trois niveaux de décomposition spatiale pour la séquence Mobile (CIF, 30Hz) : la transformée en ondelettes dyadique (WT) (a) et son arbre quaternaire (b); la transforméé en ondelettes entièrement séparable (FSWT) (c) et ses deux arbres binaires (la décomposition verticale est représenté par ligne continue et l'horizontale par ligne interrompue)(d).

La construction de la transformée en ondelettes dyadique 2D et son arbre quaternaire correspondant est illustrée en Fig. 16(a, b). Les sous-bandes résultant de la décomposition quaternaire sont notées à chaque niveau par LF, LH, HL et HH. Fig. 16(c, d) représente

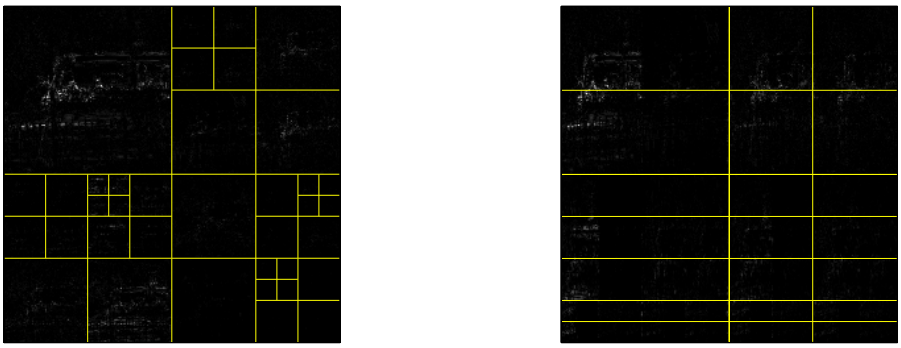
la décomposition en ondelettes entièrement séparable (*Fully Separable Wavelet Transform* - FSWT) et son arbre binaire, correspondant à la concaténation des arbres de décompositions horizontales et verticales. Les sous-bandes passe-bas et passe-haut résultant de la transformée en ondelettes 1D (filtrage horizontal ou vertical) sont notées par L et H, respectivement.

Une base d'ondelettes 2D entièrement séparable ($\mathcal{B}^{FS}$) est donnée par les produits tensoriels de *toutes* les paires d'ondelettes et fonctions d'échelle :

$$\begin{aligned}\mathcal{B}^{\mathcal{FS}} &= \left(\{\phi_{J,k}(x)\}_k \cup \{\psi_{j,k}(x)\}_{j\leq J,k}\right) \otimes \left(\{\phi_{J,k}(y)\}_k \cup \{\psi_{j,k}(y)\}_{j\leq J,k}\right) \\ &= \{\phi_{J,k_1}(x)\phi_{J,k_2}(y)\}_{k_1,k_2} \cup \bigcup_{j_1\leq J} \{\psi_{j_1,k_1}(x)\phi_{J,k_2}(y)\}_{k_1,k_2} \cup \\ &\bigcup_{j_2\leq J} \{\phi_{J,k_1}(x)\psi_{j_2,k_2}(y)\}_{k_1,k_2} \cup \bigcup_{j_1,j_2\leq J} \{\psi_{j_1,k_1}(x)\psi_{j_2,k_2}(y)\}_{k_1,k_2}\end{aligned} \tag{9}$$

Cela signifie qu'on peut trouver la FSWT simplement par l'exploitation de chaque axe séparément en utilisant une transformée unidimensionnelle. En revanche, une base d'ondelettes ($\mathcal{B}$) est donnée par les produits tensoriels de toutes les paires d'ondelettes et fonctions d'échelle *à la même échelle* :

$$\mathcal{B} = \{\phi_{J,k_1}(x)\phi_{J,k_2}(y)\}_{k_1,k_2} \cup \bigcup_{j\leq J} \{\psi_{j,k_1}(x)\phi_{j,k_2}(y), \phi_{j,k_1}(x)\psi_{j,k_2}(y), \psi_{j,k_1}(x)\psi_{j,k_2}(y)\}_{k_1,k_2} \tag{10}$$



(a) (b)

Figure 17: Quatre niveaux de décomposition spatiale pour les trames de détail dans la séquence Bus(CIF, 30Hz) : (a) la transformée en paquets d'ondelettes (WPT); (b) la transformée en paquets d'ondelettes entièrement séparable (FSWPT).

La sélection de la meilleure base de décomposition en paquets d'ondelettes entièrement séparables est réalisée à l'aide d'un algorithme *bottom-up*. Après une décomposition de l'image en suivant les deux orientations spatiales, l'algorithme *bottom-up* est d'abord appliqué sur la direction horizontale et la sélection de la meilleure base pour chaque nœud est faite en comparant les fonctions de coût dans l'arbre binaire de la décomposition H, entre le nœud courant et ses descendants :

$$\mathcal{Q}(f, \mathcal{B}^{\mathcal{FS}}{}_{j_1,j_2,k_1,k_2}) \leq \mathcal{Q}(f, \widetilde{\mathcal{B}}^{FS}_{j_1+1,j_2,2k_1,k_2} \cup \widetilde{\mathcal{B}}^{FS}_{j_1+1,j_2,2k_1+1,k_2}), \forall j_2 \tag{11}$$

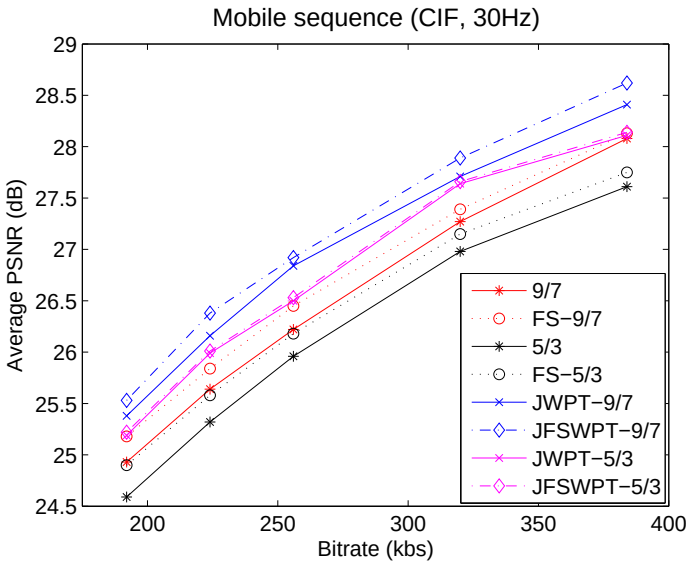


Figure 18: Courbes débit-distorsion pour la séquence Mobile (CIF, 30Hz) avec deux niveaux de décomposition spatiale.

Une fois que la meilleure base pour la direction horizontale $\widetilde{\mathcal{B}}^{FS}$ est obtenue, on passe à la sélection de la meilleure base pour la direction verticale, tout en gardant fixe l'arbre horizontal optimal et en utilisant le même critère de coût pour l'arbre vertical. Ainsi si :

$$\mathcal{Q}(f, \widetilde{\mathcal{B}}^{FS}_{j_1,j_2,k_1,k_2}) \leq \mathcal{Q}(f, \widetilde{\widetilde{\mathcal{B}}}^{FS}_{j_1,j_2+1,k_1,2k_2} \cup \widetilde{\widetilde{\mathcal{B}}}^{FS}_{j_1,j_2+1,k_1,2k_2+1}), \forall j_1 \tag{12}$$

on peut conclure que la meilleure base pour le nœud (j_1, j_2, k_1, k_2) est $\widetilde{\mathcal{B}}^{FS} = \widetilde{\widetilde{\mathcal{B}}}^{FS}$.

Comme montré par les résultats experimentaux de la Fig. 18, la séparation fréquencielle 2D la plus fine donnée par les transformées entièrement séparables permet de mieux saisir l'orientation des détails spatiaux, tout en entraînant une meilleure représentation de la texture vidéo, en comparaison avec les décompositions dyadiques classiques.

Optimisation débit-distorsion en utilisant des coupures de graphes

Les algorithmes de coupure minimale sur graphes sont apparus comme un outil de plus en plus utile pour résoudre des problèmes d'optimisation dans le traitement du signal. Habituellement, l'utilisation de coupures de graphes est due à l'une des deux raisons suivantes. Premièrement, les coupures de graphe permettent une interprétation géométrique : dans certaines conditions une coupure sur un graphe peut être vue comme une hypersurface dans l'espace N×D. Ainsi, de nombreuses applications dans la vision par ordinateur utilisent les algorithmes de coupure minimale comme des outils pour le calcul des hypersurfaces optimales. Deuxièmement, les coupures de graphe sont également utilisées comme un outil puissant dans les problèmes de minimisation d'énergie pour une classe assez large d'énergies binaires et non-binaires qui surviennent fréquemment en vision. Dans certains cas, les coupures de graphe produisent des solutions globalement optimales.

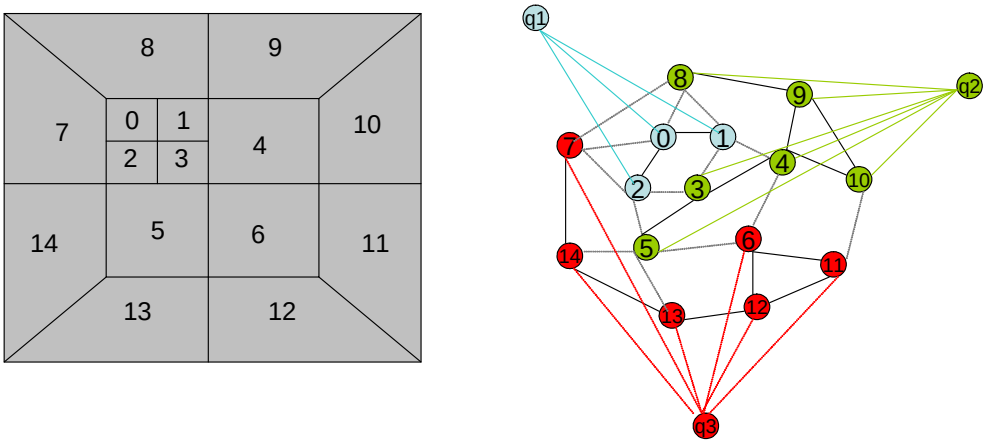


Figure 19: Trois niveaux de décomposition en contourlets (a) et la répartition des quantificateurs par sous-bandes donnée par la coupure de graphe (b).

Pour optimiser l'allocation de débit dans les problemes de compression d'images, nous avons modélisé une fonctionelle lagrangienne de débit-distorsion qui peut être minimisée par des coupures de graphe. Pour la minimisation d'une fonctionnelle correspondant à des décompositions non-orthogonales, nous avons étudié trois solutions possibles, par la modélisation de plusieurs aspects des interactions de l'énergie. Les résultats sont présentés dans le contexte de la compression d'images par sous-bandes, mais la méthode d'optimisation pourrait également être utilisée conjointement avec les algorithmes de codage vidéo.

Minimisation d'énergie en utilisant les coupures de graphes

Comme décrit dans [107, 30, 31], quelques problèmes dans le traitement du signal peuvent être exprimés naturellement en termes de minimisation d'énergie. Chacune de ces méthodes consiste à modéliser un graphe pour un type d'énergie, telles que la coupure minimale minimise globalement ou localement cette fonctionnelle d'énergie. Généralement, ces constructions graphiques sont denses et complexes, et modélisent la fonction d'énergie au niveau du pixel. Par exemple, dans [34, 29], la coupure de graphe fournit une formulation propre, souple, pour la segmentation des images. Le graphe fournit donc une manière commode de représenter les décisions locales pour la segmentation et permet un ensemble de mécanismes informatiques puissants pour extraire la segmentation globale de ces similitudes simples, locales entre les pixels. Des bons résultats dans l'optimisation d'énergie en utilisant les coupures de graphes ont été également obtenus dans la restauration des images [155, 55, 56], ainsi que dans le traitement de la vidéo stéréo [32], la segmentation du mouvement [163], la synthèse des textures des images et de la vidéo [112, 219] etc.

Nous proposons d'utiliser le mécanisme de coupures de graphes pour la minimisation de la fonction de débit-distorsion. Dans ce but, nous avons conçu un graphe spécialisé capable de représenter une décomposition multirésolution et prendre en considération les corrélations entre les sous-bandes dans une réprésentation pas forcement orthogonale.

Généralement, pour un graphe $G = (V, E, W)$, où $V/E/W$ répresente la série de sommets/arêtes et W représente les capacités des arêtes, et qui a 2 sommets spéciaux

(terminaux), $q_1, q_2 \in V$, une coupure $q_1 - q_2$ est définie comme la partition de sommets de V dans deux sous-ensembles disjoints Q_1 et Q_2 tels que $q_1 \in Q_1$ et $q_2 \in Q_2$. Le coût de la coupure est donnée par la somme des capacités w de toutes les arêtes reliant Q_1 à Q_2, i.e. :

$$C(Q_1, Q_2) = \sum_{u \in Q_1, v \in Q_2, (u,v) \in E} w(u, v) \tag{13}$$

La coupure minimale est ainsi trouvée comme la coupure ayant le coût minimal. Il y a des méthodes d'implantation rapides, polynomiales en temps, pour trouver la coupure minimale, notamment l'algorithme de Ford et Fulkerson [75].

On va considérer maintenant le graphe $G = (V, E, W)$ avec les capacités W, mais qui n'a pas seulement deux, mais un ensemble de nœuds (sommets) terminaux, $Q \in V$. Un sous-ensemble d'arêtes $\mathcal{E}_C \in E$ est appelé une coupure multiple (*multiway cut*) si les nœuds terminaux sont complètement séparés dans le graphe induit $G(\mathcal{E}_C) = (V, E - \mathcal{E}_C, W)$ et aucun sous-ensemble de $\mathcal{E}_C$ ne sépare les terminaux en $\mathcal{E}_C$. Si C est le coût de la coupure multiple, trouver la coupure minimale multiple est équivalent à trouver la coupure avec le coût minimal. En [33], Y. Boykov *et al.* trouvent la coupure minimale dans un graphe avec terminaux multiples en trouvant succesivement la coupure minimale entre chaque terminal et le reste des terminaux. Cette méthode d'approximation garantit une minimisation locale de l'énergie, très proche de la solution optimale pour les énergies concaves, et donne une solution globale pour la minimisation des fonctionnelles convexes. Comme le Lagrangien débit-distorsion réside sur une courbe convexe (c'est à dire $D(R)$), nous proposons d'utiliser cette méthode pour son optimisation.

Minimisation du Lagrangien débit-distorsion pour la compression d'images

Prenons maintenant le problème de codage d'une image à un débit donné R_{max}, avec une distorsion minimale D. Chaque image est constituée d'un nombre fixe d'unités de codage, X (par exemple, sous-bandes spatiales ou les blocs dans les sous-bandes), chacun d'entre eux codé avec un quantificateur q_i, $q_i \in Q$, où Q est l'ensemble de quantificateurs. On note par $D_i(q_i)$ la distorsion de la sous-bande i quand celle-là est quantifiée avec q_i, et par $R_i(q_i)$ le nombre de bits requis pour son codage. Le problème peut maintenant être formulé comme : trouver $\min \sum_i D_i(q_i)$, tels que $\sum_i R_i(q_i) = R \leq R_{max}$. Dans la formulation lagrangienne, ce problème d'optimisation peut être écrit dans la forme équivalente :

$$\min \sum_{i=1}^{X} \left(D_i(q_i) + \lambda R_i(q_i) \right), \quad R \leq R_{max} \tag{14}$$

où le choix de λ mesure l'importance relative de la distorsion, respectivement du débit pour l'optimisation, et dont une valeur optimale peut être déterminée par une recherche binaire. L'avantage de la formulation du problème dans Eq. (14) est que la somme et l'opérateur minimum peuvent être échangés :

$$\sum_{i=1}^{X} \min \left(D_i(q_i) + \lambda R_i(q_i) \right), \quad R \leq R_{max} \tag{15}$$

Cette formulation révèle évidemment que l'optimisation globale peut maintenant être menée indépendamment pour chaque unité de codage, en rendant possible une mise en œuvre efficace.

Modélisation du graphe avec une distorsion de prémier ordre

La distorsion D entre l'image originale x, et l'image quantifiée $\widehat{x}$ peut être écrite comme la norme L^2, c'est à dire : $D = \|\mathbf{x} - \widehat{\mathbf{x}}\|^2$. Pour les transformées orthonormales, cette norme peut être estimée de manière équivalente dans le domaine de la transformée. Toutefois, pour une transformée arbitraire (biorthogonale, redondante, non-linéaire etc.) cette propriété n'est plus vérifiée. Dans la suite, on va montrer comment la distorsion peut être approximée, et ensuite estimée dans le domaine spatial, tout en permettant une modélisation graphique des interactions entre les sous-bandes. Si, dans l'image reconstruite $\widehat{x}$, on souligne la contribution de chaque sous-bande, $\widehat{x} = \sum_i \widehat{x}_i$, où $\widehat{x}_i$ est la contribution par sous-bande (i.e., l'image reconstruite où seule la i-ème sous-bande est quantifiée et les autres sous-bandes sont mises à zéro), alors nous pouvons aussi écrire l'image de la même façon, $x = \sum_i x_i$. Toutefois, ici x_i est complètement arbitraire. Pour une base linéaire, il peut devenir $x_i = \sum_k \langle x, \widetilde{e}_{k,i} \rangle \, e_{k,i}$, où $\widetilde{e}_{k,i}$, $e_{k,i}$ sont les éléments d'analyse, respectivement de synthèse, de la base biorthogonale. Alors, on a :

$$D = \left\| \sum_i (\widehat{x}_i - x_i) \right\|^2 = \sum_{i,i'} \langle \widehat{x}_i - x_i, \widehat{x}_{i'} - x_{i'} \rangle \tag{16}$$

Dans une première approche, on peut se limiter aux termes diagonaux, i.e. :

$$D_I \cong \sum_i \|x_i - \widehat{x}_i\|^2 = \sum_i D_i(q_i) \tag{17}$$

ce qui revient à estimer la distorsion entre la contribution à l'image et à l'image quantifiée de seulement la i-ème sous-bande. Cela signifie que nous pouvons reconstruire l'image à partir des seuls coefficients de la i-ème sous-bande (les autres étant fixés à zéro) pour obtenir x_i, et des seuls coefficients quantifiés de la i-ème sous-bande pour obtenir $\widehat{x}_i$.

Dans [33] sont présentés deux algorithmes utilisant les coupures des graphes et capables de minimiser une fonctionelle d'énergie ayant la forme :

$$E(f) = E_{data}(f) + E_{smooth}(f) \tag{18}$$

où E_{smooth} est une contrainte de lissage, tandis que E_{data} mesure la distorsion introduite par la répartition des quantificateurs f entre les sous-bandes. Parce que E_{data} peut être arbitrairement choisi, avec seulement une contrainte de positivité, on va le définir comme:

$$E_{data} = \sum_i (D_i(q_i) + \lambda R_i(q_i)). \tag{19}$$

On peut définir :

$$E_{smooth} = \sum_{n_1, n_2 \in \mathcal{N}} \mathcal{V}_{n_1,n_2}(q_{n_1}, q_{n_2}) \tag{20}$$

où $\mathcal{N}$ représente le système de voisinage des nœuds, et $\mathcal{V}_{n_1,n_2}(q_{n_1}, q_{n_2})$ mesure le coût d'attribution des quantificateurs q_{n_1}, q_{n_2} aux nœuds adjacents n_1, n_2. On définit $\mathcal{V}$ comme la pénalité d'interaction de Potts, i.e. :

$$\mathcal{V} = \beta T(q_{n_1} \neq q_{n_2}) \tag{21}$$

où T est un opérateur booléen (par exemple, sa valeur équivaut 1 si son argument est vrai et 0 autrement), et β est une constante qui enforce ou diminue le lissage. Comme

il peut être vu, la définition de E_{smooth} est consistante, comme pour deux sous-bandes fortement corrélées le même quantificateur est imposé.

Un modèle simple de graphe $G = (V, E)$ pour cette fonctionnelle d'énergie peut être obtenu si on définit les sommets normaux (réguliers) comme les sous-bandes de la décomposition (X), et qui sont reliés entre eux en fonction de leur position géométrique dans le plan 2D (donc $(E - XQ)$-liens réguliers). Chaque nœud terminal $q \in Q$ peut être lié à tous les sommets non-terminaux (donc (XQ)-liens terminaux potentiels) (Fig. 19 montre l'attribution des quantificateurs d'après la coupure, c'est à dire, la sortie de l'algorithme). Pour un lien terminal (i.e., un lien entre un quantificateur q et un sommet régulier i correspondant à une unité de codage), le coût (la capacité) est donné par la distorsion apportée par ce quantificateur à la distorsion totale de l'image et le nombre de bits nécessaires pour transmettre la sous-bande quantifiée i, i.e. : $D_i(q_i) + \lambda R_i(q_i)$. Pour un lien régulier (i.e., un lien entre deux sommets réguliers voisins), le coût est 0 si les deux nœuds sont quantifiés avec le même quantificateur ou β autrement. En outre, ce coût est dynamiquement calculé pour chaque partitionnement possible f du graphe.

Modélisation du graphe avec une distorsion de corrélation croisée

Dans une première approche, nous avons consideré que les terms diagonaux dans l'approximation de la distorsion D entre l'image originale, x, et l'image quantifiée, $\widehat{x}$:

$$D \cong D_I = \sum_i \|x_i - \widehat{x}_i\|^2 . \tag{22}$$

Dans une deuxième approche, on peut aussi considérer les termes de la *corrélation croisée*, i.e. :

$$D \cong D_I + \sum_i \sum_{i' \in \mathcal{N}(i)} \langle \widehat{x}_i - x_i, \widehat{x}_{i'} - x_{i'} \rangle \tag{23}$$

où $\mathcal{N}(i)$ est le voisinage de la sous-bande i, contenant les sous-bandes fortement corrélées avec la sous-bande i. En effet, étant donné le support fini des ondelettes, plus les sous-bandes sont proches en amplitude et en fréquence, plus la corrélation entre elles est forte. Eq. (23) peut s'écrire comme :

$$D = \sum_i \underbrace{\|x_i - \widehat{x}_i\|^2}_{D_i} + \sum_i \sum_{i' \in \mathcal{N}(i)} \underbrace{\langle \widehat{x}_i - x_i, \widehat{x}_{i'} - x_{i'} \rangle}_{D_{i,i'}} \tag{24}$$

Le deuxième terme est le plus complexe (la transformée inverse plus les produits entre les images d'erreur), qui peut toutefois être divisé par deux, notant que $D_{i,i'} = D_{i',i}$ et donc :

$$D \cong \sum_i \left(D_i + 2 \sum_{i > i'} D_{i,i'} \right) \tag{25}$$

Pour $D_{i,i'}$ on a besoin de calculer l'erreur entre l'image reconstruite à partir de la i-ème sous-bande (x_i) et celle reconstruite à partir de la i-ème sous-bande quantifiée ($\widehat{x}_i$) (de même pour la sous-bande voisine i') et ensuite calculer le produit croisé.

La minimisation de la fonction d'énergie définie ci-dessus est équivalente à trouver la meilleure répartition des quantificateurs par sous-bandes. Le graphe que nous avons conçu pour résoudre ce problème a comme sommets réguliers l'ensemble des sous-bandes spatiales et l'ensemble des quantificateurs en tant que nœuds terminaux, où les

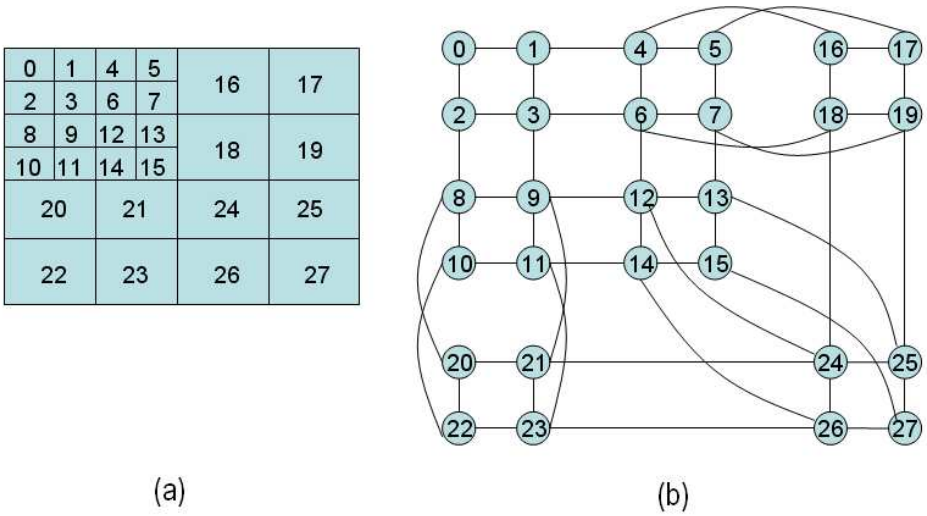


Figure 20: Modélisation de graphe par blocs : (a) deux niveaux de décomposition spatiale en ondelettes avec quatre blocs par sous-bande et (b) modélisation du réseau de nœuds réguliers.

sous-bandes sont liées par le système de voisinage $\mathcal{N}$. Chaque nœud terminal peut être lié à tous les nœuds non-terminaux, vu toutes les possibilités de quantification pour les sous-bandes spatiales (c'est à dire, $G = (V, E)$, où $V = X \cup Q$ et $E = E_N \cup E_Q$, E_N dénotant les arêtes régulieres entre les sommets réguliers dans le système de voisinage $\mathcal{N}$ et E_Q les liaisons entre les nœuds réguliers et les quantificateurs). On peut distinguer deux types de connexions : E_N et E_Q. On définit les capacités pour les liens des quantificateurs E_Q en termes de coût débit-distorsion; ainsi, le coût associé à l'arête reliant la sous-bande x au quantificateur q est défini comme $w_{x,q} = D_x(q) + R_x(q)$. Pour un lien appartenant à l'ensemble E_N, le coût associé est donné par la distorsion de corrélation croisée, c.-à-d. : $w_{x_i,x_{i'}} = \langle \widehat{x}_i - x_i, \widehat{x}_{i'} - x_{i'} \rangle$, $i' \in \mathcal{N}(i)$.

Ainsi la fonction que nous voulons minimiser peut être écrite sous la forme :

$$\min \sum_i \left(\underbrace{\|x_i - \widehat{x}_i\|^2 + \lambda R(i)}_{E_{data}(i)} + \underbrace{\sum_{i' \in \mathcal{N}(i)} \langle \widehat{x}_i - x_i, \widehat{x}_{i'} - x_{i'} \rangle}_{E_{smooth(i)}} \right) \tag{26}$$

Maintenant on établit la correspondance entre notre graphe et la coupure multiple. Dans la Fig. 19 est illustré le graphe induit $G(\mathcal{E}_C) = (V, E - \mathcal{E}_C)$ correspondant à une coupure multiple $\mathcal{E}_C$ sur G. On peut remarquer que dans le graphe induit, il devrait y avoir exactement un lien terminal pour chaque nœud régulier d'une sous-bande.

Modélisation du graphe avec une distorsion de corrélation croisée au niveau des blocs

Dans la suite, nous proposons d'estimer la distorsion au niveau des blocs (Fig. 20). Ce changement vient naturellement, en sachant que plus l'unité de codage est petite, plus

ses coefficients sont corrélés en amplitude. Au niveau des blocs, Eq. (26) devient :

$$\min \sum_{i=1}^{X} \sum_{j=1}^{N_b} \underbrace{\|x_{i,j} - \widehat{x}_{i,j}\|^2 + \lambda R(i,j)}_{E_{data}(i,j)} + \underbrace{\sum_{i' \in \mathcal{N}(i)} \left\langle \widehat{x}_{i,j} - x_{i,j}, \widehat{x}_{i',j} - x_{i',j} \right\rangle + \sum_{j' \in \mathcal{N}(j)} \left\langle \widehat{x}_{i,j} - x_{i,j}, \widehat{x}_{i,j'} - x_{i,j'} \right\rangle}_{E_{smooth}(i,j)} \quad (27)$$

où X, respectivement N_b représentent le nombre de sous-bandes, respectivement les blocs dans chaque sous-bande; $x_{i,j}$ est l'image reconstruite en considerant uniquement le bloc j de la sous-bande i; $\left\langle \widehat{x}_{i,j} - x_{i,j}, \widehat{x}_{i',j} - x_{i',j} \right\rangle$ représente la distorsion de corrélation croisée induite par le bloc j dans les sous-bandes voisines $i' \in \mathcal{N}(i)$; $\left\langle \widehat{x}_{i,j} - x_{i,j}, \widehat{x}_{i,j'} - x_{i,j'} \right\rangle$ mesure la corrélation entre les blocs voisins dans une sous-bande donnée i, avec $j' \in \mathcal{N}(j)$.

Notre graphe aura cette fois $B = X \times N_b$ sommets réguliers. Le système de voisinage, $\mathcal{N}$, contient maintenant deux types de liens réguliers entre les blocs : E_{N_M} (c'est à dire, liens de corrélation entre les mêmes blocs placés dans des sous-bandes voisines dans la décomposition multirésolution) et E_{N_G} (c'est à dire, liens entre les blocs voisins dans la même sous-bande). Le modèle géométrique peut être décrit comme : $G = (V, E)$ où $V = B \cup Q$ et $E = E_N \cup E_Q$, $E_N = E_{N_M} \cup E_{N_G}$ et Q/E_Q représente l'ensemble des quantificateurs/ les liens entre les nœuds réguliers donnés par les blocs et les quantificateurs. Pour les liens terminaux, E_Q, les capacités sont données par les coûts directs en termes de distorsion et de débit introduits par la quantification (c'est à dire, le lien entre le bloc b et le quantificateur q, (b, q), a le coût associé $w_{b,q} = D_b(q) + R_b(q)$). La capacité entre deux blocs réguliers voisins ($(b_i, b_{i'}) \in E_{N_M}$ ou $(b_j, b_{j'}) \in E_{N_G}$) est définie par la distorsion de corrélation croisée induite par la quantification courante de ces blocs.

Application à la compression d'images par sous-bandes

Les modèles de minimisation en utilisant les coupures de graphe proposés pour l'optimisation du Lagrangien débit-distorsion ont été appliqués à la compression d'images par sous-bandes.

Notre méthode semble faire mieux face à des décompositions biorthogonales telles que le schéma de pondération classique, basé sur les caractéristiques du banc de filtres de synthèse (Fig. 21), effectué par JPEG2000. Les résultats expérimentaux montrent que l'algorithme d'allocation débit-distorsion basé sur les coupures de graphes peut coder efficacement les coefficients de la transformée contourlet à bas débit (Fig. 22), tout en améliorant la qualité visuelle et numérique (PSNR). Par ailleurs, la méthode proposée peut être aussi employée avec des quantificateurs vectoriels.

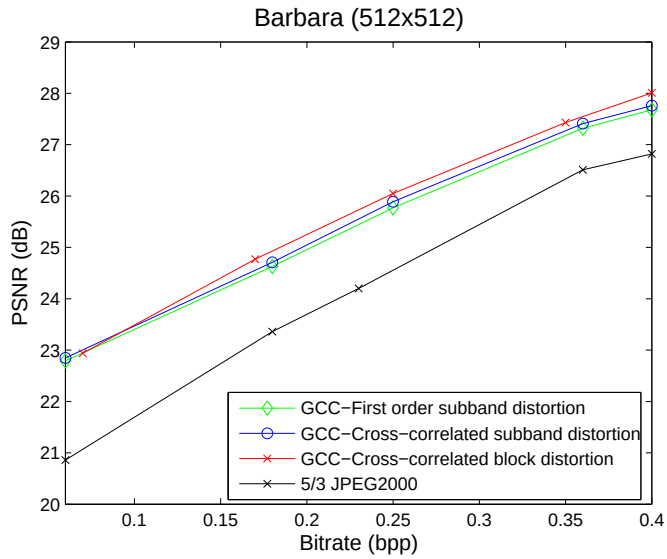


Figure 21: Courbes de débit-distorsion pour Barbara avec une décomposition en ondelettes $5/3$.

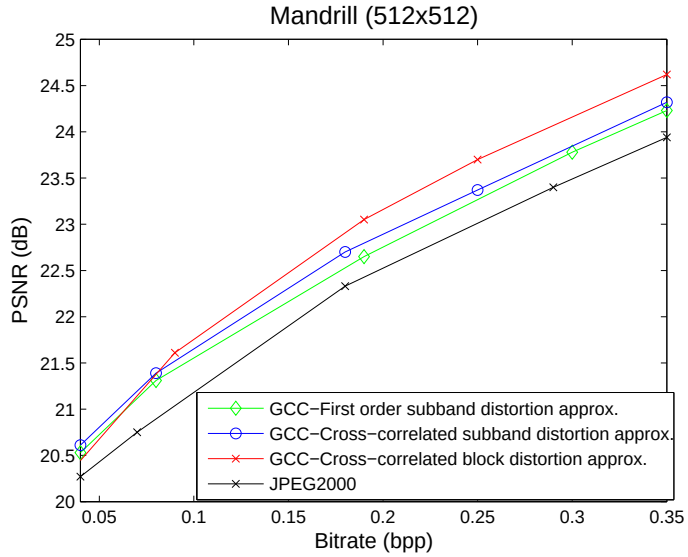


Figure 22: Courbes de débit-distorsion pour Mandrill avec une décomposition en contourlets.

Conclusion et perspectives

Cette thèse s'est inscrite dans le développement d'un codec vidéo $t+2D$ basé ondelettes. Plus spécifiquement, notre recherche a été focalisée sur la construction et l'optimisation des transformées temporelles compensées en mouvement, l'analyse de différentes méthodes de décomposition spatiale pour mieux représenter les sous-bandes temporelles afin d'augmenter la qualité objective et subjective des trames reconstruites, et enfin, l'amélioration du codage entropique en concevant des fonctions d'énergie minimisables par des coupures de graphe visant à optimiser l'allocation débit-distorsion dans la compression.

Synthèse des travaux

I—Optimisation et nouvelles méthodes pour le filtrage temporel

Traitement des coupures de scène dans le codage temporel compensé en mouvement

Généralement, les décompositions classiques le plus souvent utilisées pour la décorrélation temporelle sont les bancs de filtres dyadiques de Haar et 5/3. Le filtrage temporel est donc fait sur l'hypothèse que les trames sont fortement corrélées. Cependant, cette supposition n'est plus vérifiée quand la vidéo implique des changements de scènes, comme c'est le cas dans les films d'action, les clips vidéo etc. L'inefficacité de l'estimateur de mouvement mène à de mauvaises prédictions/mises à jour, qui, combinées avec l'implantation avec fenêtre glissante de ces filtres temporels, propage les erreurs de prédiction / mise à jour au travers les niveaux de décomposition.

Nous avons donc proposé une optimisation du filtrage temporel compensé en mouvement, capable de détecter et traiter les coupures de scène qui apparaissent dans les séquences visuelles. La structure *lifting* des filtres a été modifiée telle que le filtrage ne chevauche pas le changement de scène. D'ailleurs, les unités de codage ont été réduites près de la coupure de scène pour mieux s'adapter à ce changement. Sa mise en place au sein du codec vidéo MC-EZBC [210] permet d'atteindre des gain en PSNR atteignant environ 1.5 dB par rapport au filtrage classique.

Schéma 5-bandes compensé en mouvement

Grâce à leur efficacité de codage, les filtres plus longs sont préférés pour la décomposition temporelle. Cependant, quand les filtres sont trop longs, il est très probable qu'ils chevauchent plusieurs (et donc différentes) scènes et qu'ils perdent, ainsi, leur efficacité de décorrélation. D'ailleurs, plus le filtre temporel est long, plus le nombre de niveaux de décomposition temporelle nécessaires pour obtenir les trames clés d'approximation (si utiles pour la recherche de la vidéo dans les bases de données) est faible.

Nous avons ainsi proposé une structure *lifting* compensé en mouvement 5-bandes, qui permet d'obtenir des facteurs de scalabilité temporelle flexibles dans un codeur vidéo basé MCTF. Pour la structure proposée, nous avons développé deux schémas d'implantation : un premier utilisant la méthode de réflexion (et donc le filtre 5-bandes ne chevauche pas les autres GOPs) et un deuxième utilisant la technique de fenêtre glissante. Les deux implantations 5-bandes ont été intégrées au sein du codec vidéo basé ondelettes 3D MSRA [212]. Le banc de filtres proposé permet d'obtenir des resultats semblables à ceux obtenus avec les filtres dyadiques de Haar et 5/3, et à ceux donnés par le schéma 3-bandes compensé en mouvement. En outre, la décomposition 5-bandes donne une meilleure efficacité de codage pour les sous-bandes temporelles d'approximation, tout en améliorant la scalabilité temporelle. Ce schéma peut être utilisé avec succès dans certaines applications, telles que le codage des séquences de vidéo-surveillance, où le mouvement est faible dans la plupart des cas.

Schéma de prédiction adaptative

Dans le codage vidéo, la méthode la plus utilisée pour l'estimation du mouvement est faite par blocs, et même si une prédiction temporelle bidirectionnelle est employée, les artefacts des blocs sont encore présents. Afin d'éviter de tels artefacts, des solutions pour

la compensation en mouvement, comme une mise à jour par moyenne pondérée [184] ou une compensation en mouvement par chevauchement des blocs [211] ont été proposées.

Nous avons donc proposé ici une prédiction adaptative basée sur les moindres carrés (LMS) pour l'étape temporelle de prédiction dans un schéma *lifting* de codage vidéo scalable. Les pixels des trames temporelles sont prédits de façon optimale en utilisant un ensemble de pixels dans les trames temporellement voisines. On a illustré notre proposition sur un schéma *lifting* bidirectionnel, mais l'ensemble de pixels utilisés dans l'adaptation peut être choisi selon un nombre variable de trames impliquées dans une prédiction avec un support temporel plus long. La méthode proposée a été développée et mise en place dans le codeur vidéo MSRA. Les résultats expérimentaux montrent que, même pour une adaptation avec seulement deux pixels, la qualité visuelle des trames reconstruites est améliorée. Un compromis entre l'efficacité de compression et la complexité additionnelle venant d'une plus grande fenêtre d'adaptation peut être fait, selon l'application cible. Un gain significatif en PSNR a été obtenu pour les séquences avec un contraste élevé entre les divers segments et dans des conditions d'illumination variables.

Codage vidéo des séquences multitemporelles et multispectrales

L'efficacité de codage basé MCTF est fortement liée à la corrélation des données traitées. Nous avons donc testé les principes de codage vidéo $t+2D$ sur les séquences multitemporelles et multispectrales. Les séquences multitemporelles et multispectrales profitent de la présence de deux sources de redondance : la corrélation spatiale entre les pixels dans la même bande spectrale et la corrélation temporelle entre les différentes bandes à la même localisation spatiale.

Nous avons ainsi proposé d'évaluer les performances de compression sur les séquences SPOT (1, 2 et 4) du codeur d'images JPEG2000 et des techniques de codage vidéo $t+2D$ basé-ondelettes. On a observé que le codage intra avec EZBC [210] surpasse EBCOT [8] en termes de gain en débit dans le cas de la compression sans perte. En outre, nous avons montré que les décorrelations temporelles et spectrales peuvent être exploitées séparément ou conjointement afin d'améliorer l'efficacité de codage.

II—Optimisation et nouvelles méthodes pour le filtrage spatial

Représentation conjointe par paquets d'ondelettes

Les propriétés spectrales des sous-bandes temporelles de détail ont l'information spectrale répartie presque uniformément dans toutes les sous-bandes. Ce fait suggère que les ondelettes dyadiques, qui sont très éfficace dans le codage spatial des images naturelles ou des sous-bandes temporelles d'approximation, ne représentent pas la meilleure solution pour la décorrélation spatiale des sous-bandes temporelles de détail.

Nous avons ainsi proposé une méthode de représentation conjointe en paquets d'ondelettes pour des groupes de trames. Le choix de la meilleure-base de décomposition a été adapté dans ce but et la méthode a été dévélopée pour le codage basé MCTF. La méthode a été mise en place au sein du codeur vidéo MSRA, les meilleurs résultats étant obtenus quand une seule décomposition en paquets d'ondelettes est considérée pour la décorrelation spatiale de toutes les sous-bandes de détail dans un GOP. Nous avons également proposé et mis en œuvre les modifications algorithmiques pour le choix de la meilleure base pour les décompositions biorthogonales.

La transformée en ondelettes et paquets d'ondelettes entièrement séparables pour le codage vidéo

Trouver la meilleure représentation conjointe en paquets d'ondelettes, même pour des GOPs, peut être un procès complexe. Nous avons ainsi présenté une évaluation de la transformée en ondelettes et paquets d'ondelettes entièrement séparables pour la représentation des textures 2D dans le codage vidéo compensé en mouvement. La séparation fréquentielle 2D plus fine donnée par les décompositions entièrement séparables permet une meilleure capture de l'orientation pour les détails spatiaux, ayant comme résultat une meilleure représentation visuelle de la texture par rapport aux décompositions dyadiques classiques.

III—Optimisation de l'allocation débit-distorsion en utilisant les coupures de graphe

Beaucoup de problèmes d'optimisation dans le traitement du signal peuvent être formulés en termes de minimisation d'énergie. Nous avons ainsi conçu une fonctionnelle d'énergie minimisable par des coupures de graphes afin d'optimiser l'allocation de débit dans le codage par sous-bandes des systèmes non nécessairement orthogonaux. Nous avons présenté trois solutions possibles, en modelisant plusieurs aspects des interactions d'énergie pour la minimisation d'une fonctionnelle correspondant à une décomposition non-orthogonale. La méthode présentée a une forte efficacité de codage particulièrement à bas débits, tout en améliorant la qualité visuelle et PSNR des images reconstruites.

Perspectives

Dans cette thèse, nous avons étudié et proposé plusieurs opérateurs spatio-temporels capables de donner une représentation multirésolution parcimonieuse pour les séquences vidéo. Néanmoins, on peut identifier un certain nombre de sujets qui exigent davantage de recherche, et qui peuvent améliorer le schéma de codage $t + 2D$. Ceux-ci incluent :

- ⋆ Les opérateurs temporels adaptatifs, tels que la mise à jour adaptative, où le système visuel humain est employé pour évaluer l'impact en termes de qualité visuelle aux sous-bandes passe-bas, puisque, dans le MCTF, le résidu de la compensation en mouvement est encore employé dans l'étape de mise à jour des trames temporelles d'approximation et peut donc causer des artefacts fantôme si l'estimation de mouvement est imprécise.

- ⋆ L'application et l'optimisation de la compensation en mouvement par pixel pour le codage temporel adaptatif. Un algorithme pel-récursif d'estimation de mouvement peut être employé ainsi que la prédiction temporelle adaptative basée LMS pour obtenir un flux de mouvement plus cohérent au niveau des pixels et donc réduire plus d'artefacts sur les contours des objets.

- ⋆ Un schéma *lifting* avec la mise à jour comme première étape devrait être considéré pour la prédiction adaptative basée-LMS. Puisque l'étape adaptative a lieu à la prédiction, et les sous-bandes passe-haut sont encore employées dans le schéma *lifting* pour renforcer le signal dans les sous-bandes passe-bas, certaines erreurs de prédiction peuvent passer facilement d'un niveau de décomposition à l'autre. En

changant l'ordre des étapes du *lifting*, les erreurs de prédiction ne se propagent plus dans l'arbre de décomposition.

De plus, la description conjointe par paquets d'ondelettes qui caractérise les propriétes spatio-temporelles d'un GOP peut être exploitée comme critére dans la classification et la recherche des séquences vidéo dans les bases de données.

Egalement, la méthode d'allocation de débit en utilisant les coupures de graphe pourrait être employée conjointement avec les algorithmes déjà existants de codage entropique dans un codeur vidéo $t + 2D$. D'ailleurs, la méthode d'évaluation de débit pourrait être remplacée par un algorithme entropique plus évolué ou avec une vraie évaluation du débit, ainsi une amélioration des résultats pour les systèmes orthogonaux peut être envisagée.

Néanmoins, des algorithmes basés sur les coupures de graphe peuvent être envisagables pour l'optimisation des champs de vecteurs de mouvement, où le meilleur vecteur de mouvement est obtenu par la minimisation en termes de distorsion et de débit (comme classiquement fait), mais également par la corrélation avec le voisinage, qui pourrait être donnée par : la distance entre des vecteurs de mouvement adjacents, la distorsion introduite par les artefacts ou la différence de débit nécessaire pour coder les vecteurs de mouvement.

Introduction

The transmission of multimedia content over IP networks such as the Internet and wireless networks has been growing steadily over the past years. Moreover, multimedia streaming and the set of applications that rely on streaming are expected to continue growing. Meanwhile, the current quality of streamed multimedia content in general, and video in particular, still needs a great deal of improvement before IP-based video streaming can be widely deployed. To achieve this level of acceptability and proliferation of IP video, there are many technical challenges that have to be addressed in the areas of video coding and networking. A framework that addresses both video coding and networking challenges associated with IP-based video streaming is *scalability*. From a video-coding point of view, scalability plays a crucial role in delivering the best possible video quality over unpredictable, best-effort[1] networks. Bandwidth variation is one of the primary characteristics of best-effort networks, and current IP networks are a prime example of such networks. Therefore, video scalability enables an application to adapt the streamed video quality to changing network conditions (and specifically to bandwidth variation) and device complexities [197]. From the networking point of view, scalability is needed to enable a large number of users to view any desired video stream, at anytime and from anywhere.

Scalable techniques try to avoid simulcast solutions in which several encoders are run in parallel. The simulcast solution would require the knowledge about the network and decoder capabilities in order to select in advance the optimum encoding parameters. Since different streams are multiplexed, this may lead to network overload and restriction to a small number of possible bitstreams for multiplexing. Even though point-to-multipoint connections are enabled by the simulcast solution, there is a clear loss in efficiency. Consequently, any scalable video-coding solution has to enable a very simple and flexible streaming framework; hence, it must meet the following requirements:

- ⋆ The solution must enable a streaming server to perform minimal real-time processing and rate control when outputting a very large number of simultaneous unicast (on-demand) streams.

- ⋆ The scalable video-coding approach over IP networks has to be highly adaptable to unpredictable bandwidth variations due to heterogeneous access technologies of the receivers (e.g., analog modem, cable modem, xDSL, wireless mobile and wireless LANs, etc.) or due to dynamic changes in network conditions (e.g., congestion events).

- ⋆ The video-coding solution must enable low-complexity decoding and low-memory

[1] In a *best-effort* network, all users obtain best-effort service, meaning that they obtain unspecified variable bit rate and delivery time, depending on the current traffic load.

requirements to provide to common receivers (e.g., set-top boxes, digital televisions or mobile devices), in addition to powerful computers, the opportunity to stream and decode any desired Internet video content.

- ⋆ The streaming framework and the related scalable video-coding approach should be able to support both multicast and unicast applications. Generally, this eliminates the need for coding the content in different formats to serve different types of applications.
- ⋆ The scalable bitstream must be resilient to packet-loss events which are quite common over IP networks.

The above requirements were the primary drivers behind the design of the existing and emerging scalable video-coding schemes. There are three key factors which play an important role in the achievement of the above requirements for video coding; these are spatial, temporal, and quality (or SNR) scalabilities, allowing full/partial decoding. In a spatial-scalable scheme, full decoding leads to high spatial resolution, while partial decoding leads to reduced spatial resolutions (reduction of the format). In a temporal-scalable scheme, partial decoding provides lower decoded frame rates (temporal resolutions). In an SNR-scalable scheme, temporal and spatial resolutions could be fixed and the video quality (SNR) varies upon how much of the bitstream is decoded. This is the main reason for chosing the scalable lifting-based wavelet-coding paradigm as the conceptual development framework for the contributions of this thesis.

The objectives of this thesis consist of the analysis and design of new and efficient spatio-temporal-SNR scalable video-coding systems. More exactly, our research interests have been focused on:

- ⋆ construction and optimization of motion-compensated temporal-filtering (MCTF) schemes in order to enhance both the objective and subjective coding quality;
- ⋆ better representation of the temporal subbands by using anisotropic spatial decompositions in order to capture the orientation of spatial details.
- ⋆ improvement of entropy coding by designing a graph-cut solvable energy functional for the Lagrangian rate distortion optimization problem.

During this thesis, we have contributed to the development of the wavelet-based video codec proposed by the AdHoc Vidwav MPEG group [212]. Part of this work was supported by the 6th Framework Programme of European Commission under the grant number IST-FP6-507752 (MUSCLE Network of Excellence) and has been realized in collaboration with Bilkent University.

Thesis organization

This thesis is organized into two parts: the first part is an overview of wavelet basics and scalable video-coding strategies. The second part encapsulates the thesis work by presenting the contributions following the above-mentioned research directions.

Wavelet basics and scalable video coding overview

***Chapter 1*: Wavelet basics** This chapter reviews the relevant wavelet issues required in the development of this thesis work and introduces the lifting mechanism used for wavelet construction and implementation.

***Chapter 2*: Scalable video coding** This chapter presents scalability requirements and the general structure of a video codec to finally present an overview of scalability techniques from predictive to wavelet-based coding strategies.

Design of a scalable video coder

***Chapter 3*: Temporal video processing** This chapter presents contributions to the MCTF stage in video coding. It starts by describing a modified MCTF scheme able to detect and correctly process scene-cuts that may occur in a video sequence. A 5-band temporal-lifting scheme, which has been designed for low-motion video sequences and which has a direct application to video surveillance, is presented. Moreover, an adaptive temporal predictor, able to remove motion estimation artefacts to thereby enhance both the visual and SNR quality of the decoded sequence, is described. Further, knowing the coding efficiency of the MCTF-based schemes for video sequences, a comparative study of the 3D-wavelet coding approach and JPEG2000 on multi-temporal and multispectral satellite sequences ends this chapter.

***Chapter 4*: Spatial video processing** Several research studies regarding the properties of the temporal-decomposition subbands have concluded that the separable wavelet decompositions are not the most appropriate for representing the residual temporal subbands. In this chapter, a joint wavelet-packet representation for the detail temporal subbands is presented, as well as an improved and efficient best-basis algorithm for biorthogonal wavelets. Moreover, fully separable wavelet and wavelet-packet transforms have been studied for the spatial decorrelation of the temporal subbands, these decompositions leading to a better video-texture representation due to the capture of the orientation of spatial details.

***Chapter 5*: Rate-distortion optimization using graph-cuts** The geometric features of images such as edges are difficult to represent. When a redundant transform is used for their extraction, the compression challenge is even more difficult. This chapter presents the design of a graph-cut-solvable energy functional for Lagrangian rate-distortion optimization in subband image coding. It applies to any kind of decomposition, including biorthogonal or redundant transforms. Three graph-based solutions are described, by modeling several aspects of energy interactions for the minimization of a non-orthogonal system. In this way optimal rate-distortion truncation of scalable streams is achieved, including trade-offs at various rates.

***Appendix A*: Joint source-channel coding** This appendix presents thesis work on joint source-channel coding, published in two journal papers, which is slightly outsite the principal focus of this dissertation. A robust joint source-channel coding scheme for the transmission of video sequences over Gaussian channels using uncoded and coded index assignment via Reed-Muller codes is decribed in a first part, continued by the presentation of a coding system designed for the video transmission over flat Rayleigh-fading channels. Our contribution to these papers has been the design and implementation of a subband-based unequal protection scheme with RCPC codes for the prominent MCTF-based MC-EZBC codec [210].

Part I

Wavelet basics and scalable video coding overview

Chapter 1

Wavelet basics

Wavelets and wavelet theory have generated much interest during the last decades. Moreover, wavelet-based compression schemes have become increasingly important and gained widespread acceptance, an example being the JPEG2000 still image compression standard [8, 178]. In this chapter we review the relevant wavelet issues required in the development of this thesis work, and we introduce the lifting mechanism used for wavelets construction and implementation. Section 1.2 starts with an introduction to multiresolution analysis and the discrete wavelet transform. The connection between wavelets, filter banks and subband coding is presented. Section 1.3 introduces the lifting scheme and describes its use and advantages in coding algorithms.

1.1 Introduction to wavelet theory

The main idea behind wavelet analysis is to decompose a signal f using a basis of functions ψ_i:

$$f = \sum_i a_i \psi_i$$

To have an efficient representation of the signal f using only a few coefficients a_i, it is very important to use a suitable family of functions ψ_i. The functions ψ_i should match the features of the data we want to represent. Usually, signals have the following features: they are both limited in time [1] and in frequency. What we need is a compromise between the pure time-limited and band-limited basis functions, a compromise that combines the best of both worlds: wavelets.

The main feature of wavelets which is important for coding applications is good decorrelation. Wavelets are localized in both the space/time and scale/frequency domains. Hence they can easily detect local features in a signal. Moreover, wavelets are based on a multiresolution analysis. A wavelet decomposition allows thus the analysis of a signal at different resolution levels (or scales). Also, as we will show in 1.2, wavelets are smooth, which can be characterized by their number of vanishing moments. A function defined on the interval $[a, b]$ has n vanishing moments if:

$$\int_a^b f(x) x^i dx = 0, \quad \forall i \in \{0 \dots n-1\}$$

[1] Space-limited in the case of images

As one can observe, the higher the number of vanishing moments, the higher the degree of polynomials whose wavelet coefficients are null.

We will give in the following a brief presentation of discrete wavelet functions, by firstly introducting multiresolution analysis.

1.2 Multiresolution analysis

Consider the L^2 space, which is the Hilbert space of square integrable functions in $\mathbb{R}$:

$$L^2 = \left\{ f : \int_{-\infty}^{+\infty} f^2(x)dx < \infty \right\}.$$

In a multiresolution analysis [125], the L^2 space is decomposed in nested subspaces V_j, i.e.:

$$\ldots \subseteq V_2 \subseteq V_1 \subseteq V_0 \subseteq V_{-1} \subseteq V_{-2} \subseteq \ldots$$

such that the closure of their union is L^2:

$$\overline{\bigcup_{j=-\infty}^{+\infty} V_j} = L^2$$

and their intersection contains only the zero-function:

$$\bigcap_{j=-\infty}^{+\infty} V_j = \{0\}.$$

In the dyadic case[1], a function $f(x)$ that belongs to one of these subspaces V_j has the following properties:

$$f(x) \in V_j \quad \Leftrightarrow \quad dilation \quad f(2x) \in V_{j-1} \tag{1.1}$$

$$f(x) \in V_0 \quad \Leftrightarrow \quad translation \quad f(x+1) \in V_0 \tag{1.2}$$

If we can find a function $\phi(x) \in V_0$ such that the set of functions including $\phi(x)$ and its integer translates $\{\phi(x-k)\}_{k\in\mathbb{Z}}$ form a basis for the V_0 space, we will call it a *scaling function*. For the other subspaces V_j ($j \neq 0$), we define:

$$\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k)$$

Because the subspaces V_j are nested, i.e.: $V_j \in V_{j-1}$, we can decompose V_{j-1} in V_j and W_j, the orthonormal complement of V_j in V_{j-1}:

$$V_j \oplus W_j = V_{j-1}, \quad W_j \perp V_j.$$

The direct sum of subspaces W_j is equal to L^2:

$$\overline{\bigcup_{j=-\infty}^{+\infty} V_j} = \overline{\bigoplus_{j=-\infty}^{+\infty} W_j} = L^2$$

[1] In the dyadic case, each subspace V_j is twice as large as V_{j+1}

This means that V_j is a *coarse-resolution* representation of V_{j-1}, while W_j carries the *high-resolution* difference information between V_{j-1} and V_j.

Now, if we can find a function $\psi(x) \in W_0$ that obeys the translation property (1.2), i.e.:

$$\psi(x) \in W_0 \quad \Leftrightarrow \quad translation \quad \psi(x+1) \in W_0 \tag{1.3}$$

such that the set of functions consisting of $\psi(x)$ and its integer translates $\{\psi(x-k)\}_{k\in\mathbb{Z}}$ form a basis for the W_0 space, we will call it a *wavelet function*. For the other subspaces W_j ($j \neq 0$) we define:

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k).$$

Because both V_0 and W_0 are subspaces of V_{-1}, i.e. $V_0 \subset V_{-1}$ and $W_0 \subset V_{-1}$, we can express $\phi(x)$ and $\psi(x)$ in terms of the basis functions of V_{-1}:

$$\begin{cases} \phi(x) = 2\sum_k h_k \phi(2x-k), \\ \psi(x) = 2\sum_k g_k \phi(2x-k). \end{cases}$$

Due to multiresolution analysis, these relations are also valid between V_{j-1}, V_j, and W_j for arbitrary j. We will call thus h_k and g_k the filter coefficients that uniquely define the scaling function $\phi(x)$ and the wavelet function $\psi(x)$.

Since $V_{j-1} = V_j \oplus W_j$, we can thus express a function $f(x)$, which is initially written in terms of the basis functions of V_{j-1}, in terms of the basis functions of V_j and W_j:

$$f(x) = \sum_k \lambda_{j-1,k}\phi_{j-1,k}(x) = \sum_l \lambda_{j,l}\phi_{j,l}(x) + \sum_l \gamma_{j,l}\psi_{j,l}(x)$$

where the transform coefficients $\lambda_{j,l}$ and $\gamma_{j,l}$ are defined as:

$$\begin{aligned} \lambda_{j,l} &= \sqrt{2}\sum_k h_{k-2l}\lambda_{j-1,k}, \\ \gamma_{j,l} &= \sqrt{2}\sum_k g_{k-2l}\lambda_{j-1,k}. \end{aligned}$$

This operation is known as the *Fast Wavelet Transform* (FWT). It has a linear complexity, i.e. $O(n)$, the amount of work being proportional to the signal length. The filter h_k is a low-pass filter, while g_k is a high-pass filter. The inverse wavelet transform is obtained in a similar manner. A 2-tap one-level filter-bank decomposition is shown in Fig. 1.1.

The subspaces V_j are nested, and each of them can be split into two subspaces, V_{j+1} and W_{j+1}. If we start with the highest available resolution subspace V_0, and we recursively perform the following decompositions:

$$\begin{aligned} V_0 &= V_1 \oplus W_1, \\ &= V_2 \oplus W_2 \oplus W_1, \\ &= V_3 \oplus W_3 \oplus W_2 \oplus W_1, \\ &\vdots \end{aligned}$$

we obtain a decomposition tree as shown in Fig. 1.2. As one can observe, the amount

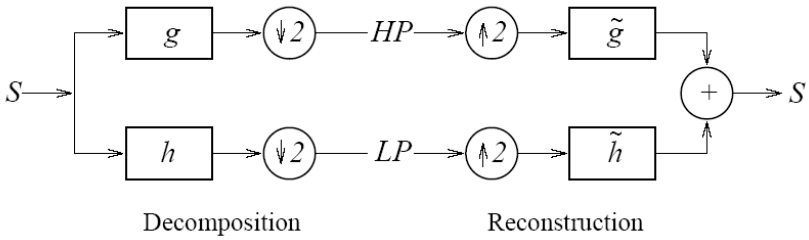


Figure 1.1: The filter-bank algorithm for orthogonal wavelets: the signal S is filtered and downsampled to get a low-pass signal (LP) and a high-pass signal (HP). It can be reconstructed by upsampling and filtering with the correct filters.

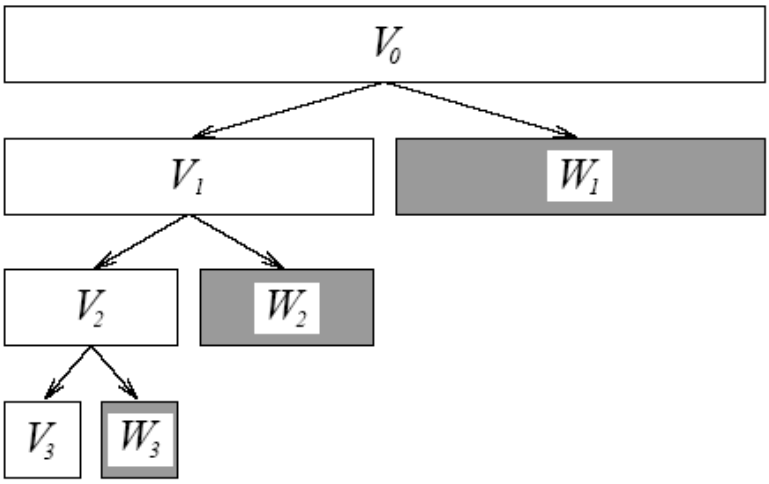


Figure 1.2: Example of wavelet decomposition tree.

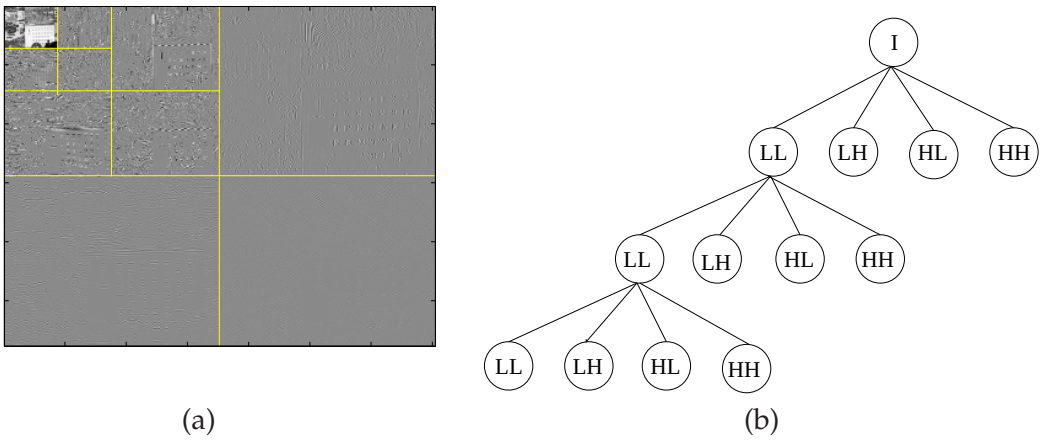


Figure 1.3: Three-level wavelet spatial decomposition for Mobile(CIF, 30Hz) sequence: wavelet transform (WT) (a) and its decomposition quadtree (b).

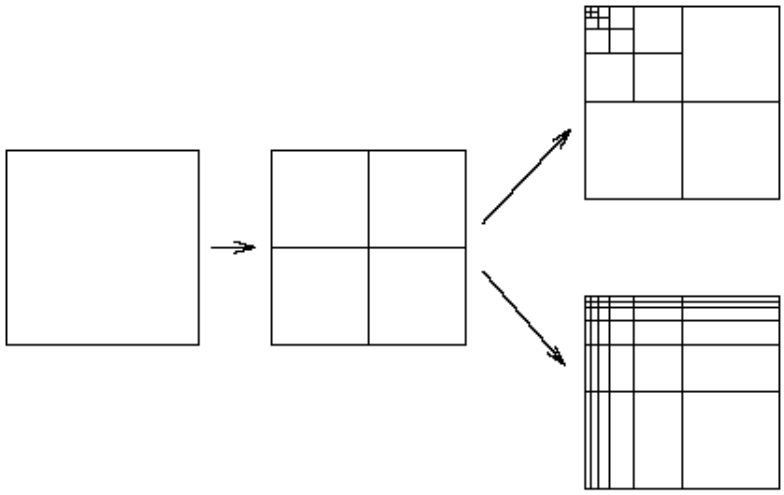


Figure 1.4: The two-dimensional wavelet transform: separable and fully separable approach.

of data to transform is halved at each resolution level, the total complexity of the full wavelet decomposition still being $O(n)$.

The above wavelets were defined on $\mathbb{R}$, i.e. on a one-dimensional domain. In the 2D case, we get a decomposition as shown in Fig. 1.3 for three spatial wavelet-filtering levels. In order to create wavelets on higher-dimentional domains, we can perform the wavelet transform independently on each dimension. Because the wavelet transform can be written as a multiplication with wavelet transform matrix, and, due to the associativity of the matrix product, the exact order is not an issue. Moreover, we can alternate the wavelet transform in each dimension at each resolution level or fully decorrelate in one dimension at a time (see Fig. 1.4) obtaining thus the so called separable [125, 123, 124] and fully separable wavelet transforms [188, 147]. The fully separable wavelet transform will be developed in Section 4.3, where we will use it for spatial decorrelation in a subband wavelet video-coding approach.

Due to the subsampling in the filter-bank approach, a wavelet transform is not translation invariant; that is, if a signal is delayed or advanced, its wavelet transform is not simply a delayed or advanced version of the wavelet transform of the original signal. Only if the delay is a multiple of 2^n, where n is the number of transform levels, the wavelet transform will be a delayed or advanced version of the original transformed signal, as shown in Fig. 1.5. In the 2D case, this condition needs to be true in both horizontal and vertical directions.

1.2.1 Orthogonal wavelets

If $\phi_{j,k}(x)$ and $\psi_{j,k}(x)$ are orthonormal, that is :

$$\begin{aligned} V_j &\perp W_j, \\ \left\langle \phi_{j,l}, \phi_{j,l'} \right\rangle &= \delta_{l-l'}, \\ \left\langle \psi_{j,l}, \psi_{j',l'} \right\rangle &= \delta_{j-j'}\delta_{l-l'}, \end{aligned}$$

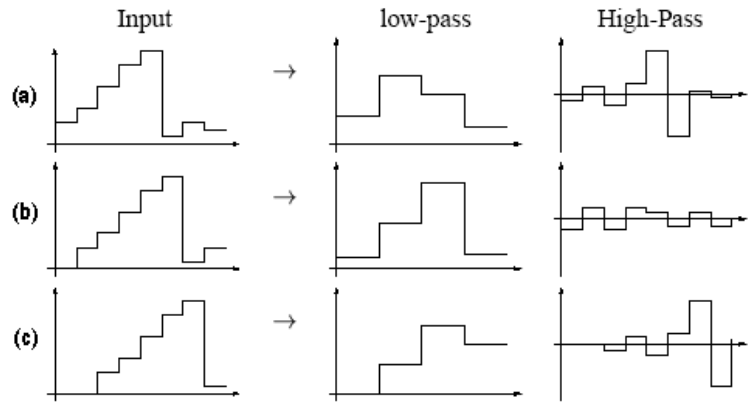


Figure 1.5: Translation variance of wavelet transform: when delaying the input signal (a) by one step, the resulting wavelet transform is completely different (b), while a delay of the input signal with two steps (c) results in a delayed version (by one step at the coarser level) of the original transformed signal.

where δ_i is the Kronecker symbol, i.e.:

$$\delta_i = \begin{cases} 1, & i = 0 \\ 0, & otherwise, \end{cases}$$

then we can calculate the coefficients of the decomposition:

$$f(x) = \sum_l \lambda_{j,l}\phi_{j,l}(x) + \sum_l \gamma_{j,l}\psi_{j,l}(x)$$

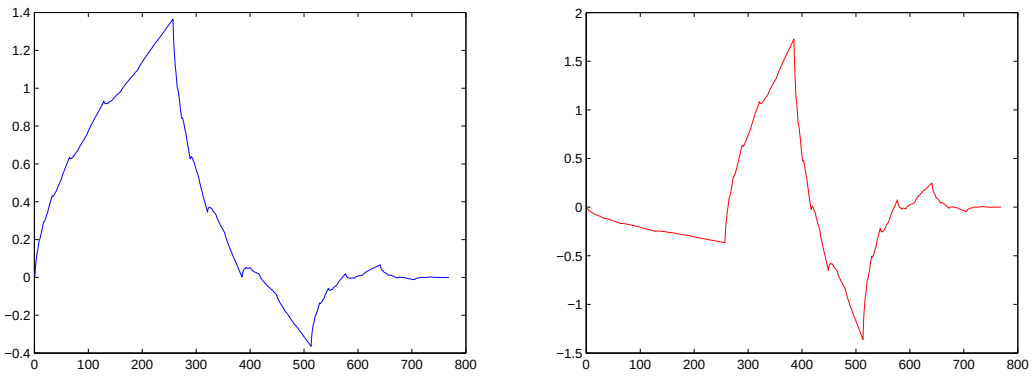
by taking the inner products with the scaling and wavelet functions, i.e.:

$$\lambda_{j,l} = \langle f, \phi_{j,l} \rangle ,$$
$$\gamma_{j,l} = \langle f, \psi_{j,l} \rangle .$$

The decomposition into the wavelet basis is guaranteed to be stable: if the function $f(x)$ is slightly changed, there will be only a slight change of the coefficients $\lambda_{j,l}$ and $\gamma_{j,l}$. For an orthonormal decomposition, the energy of the original signal is preserved in the transform domain, i.e. Parseval's identity holds:

$$\|f\|^2 = \sum_k \lambda^2_{j-1,k} = \sum_l \lambda^2_{j,l} + \sum_l \gamma^2_{j,l}. \tag{1.4}$$

An example of orthogonal wavelets is the family of orthogonal wavelets constructed by Daubechies [59]. The scaling function $\phi(x)$ and the wavelet function $\psi(x)$ with two vanishing moments (also known as *D4* because the corresponding wavelet filters h and g have 4 taps) are shown in Fig. 1.6.



(a) (b)

Figure 1.6: Daubechies orthogonal (a) scaling $\phi(x)$ and (b) wavelet $\psi(x)$ functions with two vanishing moments.

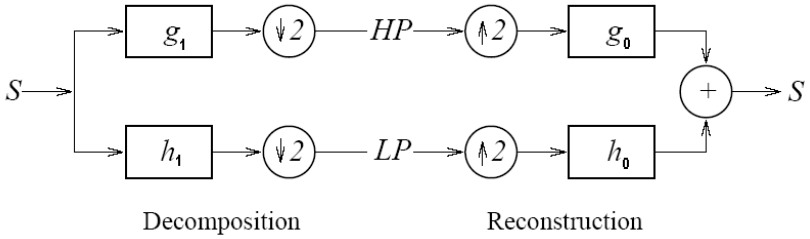


Figure 1.7: Two-band perfect-reconstruction biorthogonal filter-bank.

1.2.2 Biorthogonal wavelets

Recall that the dilations and translations of the scaling function $\phi_{j,k}$ constitute a basis for V_j and, similarly, $\psi_{j,k}$ for W_j. In the biorthogonal case, a dual multiresolution analysis with dual subspaces, $\widetilde{V}_j$ and $\widetilde{W}_j$, generated from a dual scaling function $\widetilde{\phi}_{j,k}$ and a dual wavelet function $\widetilde{\psi}_{j,k}$, respectively, is defined. The biorthogonality conditions:

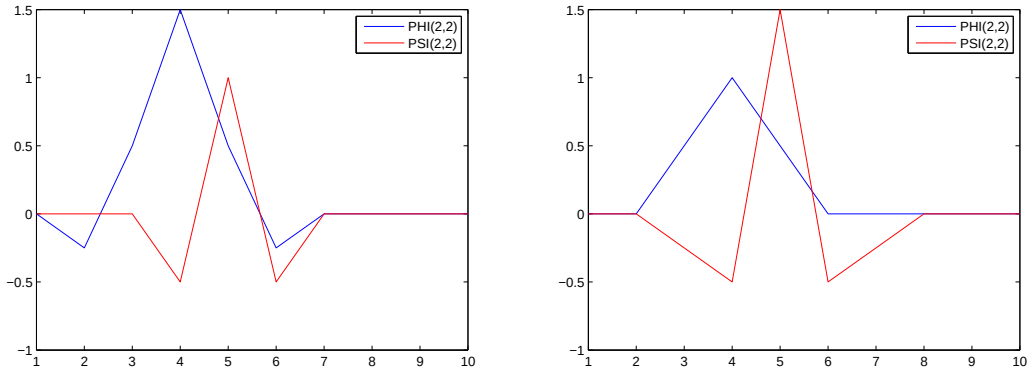
$$\widetilde{V}_j \perp W_j,$$
$$V_j \perp \widetilde{W}_j$$

imply:

$$\left\langle \widetilde{\phi}_{j,k}, \phi_{j,l'} \right\rangle = \delta_{l-l'},$$
$$\left\langle \widetilde{\psi}_{j,k}, \psi_{j',l'} \right\rangle = \delta_{j-j'}\delta_{l-l'}.$$

Because the biorthogonal wavelets form a Riesz basis, i.e.:

$$A \left\| f \right\|^2 \leq \sum_l \lambda_{j,l}^2 + \sum_l \gamma_{j,l}^2 \leq B \left\| f \right\|^2, \tag{1.5}$$



(a) (b)

Figure 1.8: Scaling $\phi(x)$ and wavelet $\psi(x)$ functions for the Cohen-Daubechies-Feauveau (2,2) biorthogonal wavelets: (a) analysis, (b) synthesis.

with A and B positive constants[1], the biorthogonal wavelet decomposition is still stable, the decomposition coefficients $\lambda_{j,l}$ and $\gamma_{j,l}$ being calculated as inner products with the dual basis functions:

$$\lambda_{j,l} = \left\langle f, \widetilde{\phi}_{j,l} \right\rangle, \gamma_{j,l} = \left\langle f, \widetilde{\psi}_{j,l} \right\rangle.$$

We can still use the filter-bank algorithm if we use the analysis filter pair (h_1, g_1) for the decomposition and the synthesis filter pair (h_0, g_0) for the reconstruction, as shown in Fig. 1.7. An example of biorthogonal wavelets is the family of biorthogonal wavelets constructed by Cohen, Daubechies and Feauveau [47]. The analysis/synthesis scaling and wavelet functions with two vanishing moments are shown in Fig. 1.8, and a three-level decomposition of Lenna (512×512) image using this filter is shown in Fig. 1.9.

A biorthogonal wavelet decomposition extends the orthogonal one, it is more flexible, and generally easier to design. The advantages of a biorthogonal system with respect to an orthogonal one are:

- ⋆ in the case of FIR filter-banks, the orthogonal filters must be of the same length, and the length must be even; this restriction is relaxed for biorthogonal systems.
- ⋆ biorthogonal wavelets may be symmetric, and thus, filter frequency response may have a linear phase; on the other hand, there are no two-band orthogonal transforms with more than two non-zero coefficients having FIR linear phase.
- ⋆ in a biorthogonal decomposition, the analysis and synthesis filters may be switched and the resulting decomposition can still give good results. Therefore, the appropriate arrangement may be chosen for the application at hand. For example, in image compression, it has been observed that the use of the smoother filter in the reconstruction of the coded image leads to better visual appearance.

Biorthogonal decompositions have also some disadvantages:

[1] In the orthogonal case $A = B = 1$, i.e. Parsevals's identity.

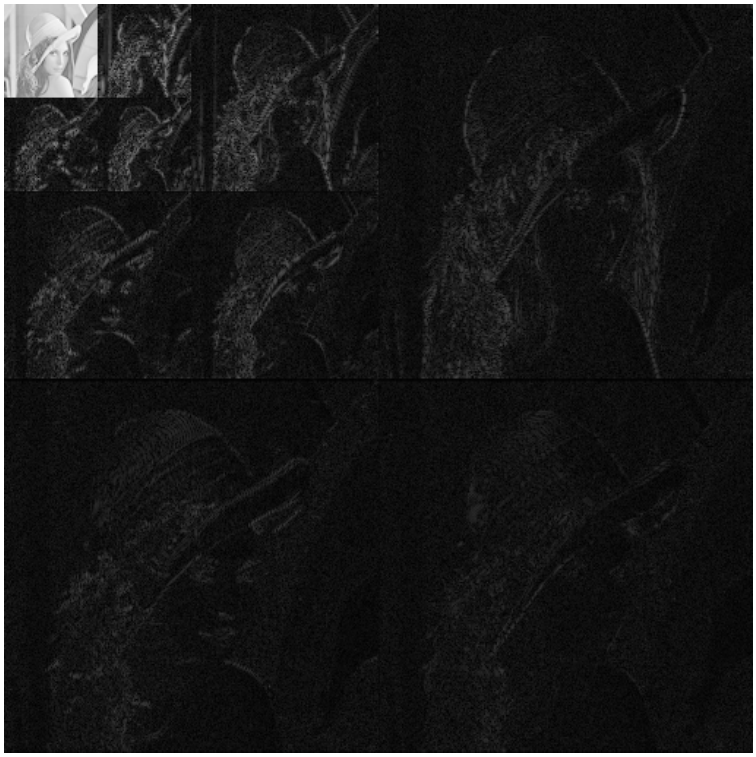


Figure 1.9: Three level Cohen- Daubechies-Feauveau (2,2) biorthogonal wavelet decomposition of the Lenna (512×512) image.

- ⋆ Parseval's theorem no longer holds for biorthogonal wavelets. This means that the energy of the coefficients is not the same as the energy of the images being spanned. Many design efforts have been devoted for making near-orthogonal systems.
- ⋆ White noise remains white after an orthogonal transform, but becomes correlated after a non-orthogonal transform. This may be considered when biorthogonal systems are employed in estimation or detection applications.

1.2.3 Redundant wavelets

We have mentioned in section 1.2 that the wavelet transform is not translation invariant. The idea behind the *redundant wavelet transform* (or overcomplete wavelet transform) [131] is to solve this problem by removing the subsampling step. This is important for the applications that can suffer from translation variance (see Fig. 1.5), like features detection, denoising or motion-vector estimation in MCTF-based in-band video coding, as we will see in section 2.3.1.6.

In the redundant wavelet transform one gets rid of the decimation step, causing all the subbands to have the same size as the size of the input signal. At each resolution level, the filters have to be upsampled [1] in order to keep a consistent multiresolution analysis. Because all the subbands have the same size, the computational complexity is no longer $O(n)$ but $O(kn)$, where $k \leq \log n$ is the number of decomposition levels.

[1] A filter is upsampled by putting zeros between the succesive filter coefficients.

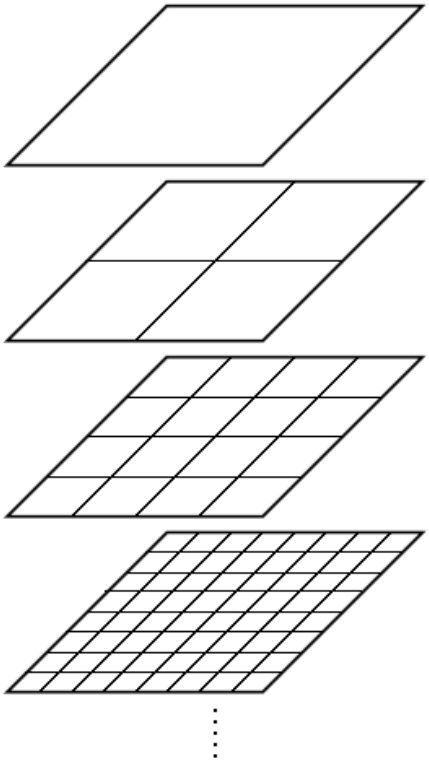


Figure 1.10: 2D full wavelet-packet decomposition.

1.2.4 Wavelet packets

In a separable wavelet decomposition, at each transform level the low-frequency data is split into a low- and a high-frequency part, as shown in Fig. 1.2. In a wavelet-packet transform, the high-frequency part may also be split into a *low* and a *high* frequency part [50]. For two-dimensional signals, this produces a decomposition as shown in Fig. 1.10 and leads to a redundant tree of possible basis functions. A wavelet-packet transform can be calculated with a complexity of $O(n \log n)$, compared to $O(n)$ as in the case of a dyadic wavelet transform.

The aim of wavelet-packet analysis is to choose the *best basis*[1], i.e. to find the set of functions that best decorrelates the input data and which form a subset of the basis functions in the decompostion tree [208]. The selection of the best basis is equivalent to the answer to the question: do we best split this part into a low and a high frequency part or not? We can make these decisions by starting at the bottom level of the tree and comparing the *cost* of two (1D) respectively four (2D) neighboring *child* subbands in this level (i.e. j) with the *cost* of the parent subband in the lower level (i.e. $j-1$). The *cost* is usually a measure for the number of bits we would need to store the data in the corresponding basis, i.e. a measure for the achieved decorrelation. We retain the cheapest

[1]Generally speaking, a wavelet-packet decomposition is any subset of the full decomposition tree. Sometimes, one does not necessarily have to search for the best basis: a fixed wavelet-packet decomposition might work well.

decomposition for this current level and proceed in the same way with the lower level, until we arrive at the first level (i.e. V_0). The basis functions that correspond to the retained level decompositions form the best basis.

For the cost computation of a wavelet-packet coefficient matrix $v = [v_{i,j}]$, there are several possibilities. We mention a few of them:

- ⋆ *1-Norm* based cost function: the cost of storing a matrix is proportional the the 1-norm of the matrix:
$$cost = \sum_{i,j} |v_{i,j}|$$
- ⋆ *First-order Shannon entropy*-based cost function: the first-order entropy is given by the number of bits per symbol needed to encode a string of symbols, by considering individual symbols:
$$cost = -\sum_{i,j} v_{ij} \log |v_{ij}|$$
- ⋆ *Logarithm of energy*-based cost function:
$$cost = \sum_{i,j} \log(v_{ij})^2$$
which is equivalent to:
$$cost = \sum_{i,j} \log |v_{ij}|$$
this being a variant of first-order Shannon entropy.

A pruning algorithm used for the computation of the best basis for orthogonal wavelet transforms will be presented in section 4.2.1.1. Moreover, in section 4.2.1.2 we will present an efficient best-basis algorithm for finding the best wavelet-packet decomposition of a biorthogonal wavelet transform. In the following we will introduce the lifting scheme, an efficient mechanism for implementing wavelet tranforms.

1.3 Lifting scheme

The lifting scheme, formally introduced by Sweldens [173, 174, 175], enables an easy and efficient construction of wavelet transforms. A very important feature of the lifting scheme is that every filter bank based on lifting automatically satisfies perfect reconstruction properties. The lifting scheme starts with a set of well known filters, whereafter lifting steps are used in an attempt to improve (lift) the properties of a corresponding wavelet decomposition.

1.3.1 Lifting steps: predict and update

As seen in section 1.2, the wavelet transform of a one-dimensional signal is a multiresolution representation of that signal where the wavelets are the basis functions which, at each resolution level, decorrelate the signal. Thus, at each level, the (low-pass part of the) signal is split into a high-pass and a low-pass part. These high-pass and low-pass parts are obtained by applying corresponding wavelet filters. In general, these filters are

coupled if certain conditions are to be fulfilled, like, for example, perfect reconstruction: there have to exist filters to perform the inverse transform without any data loss.

The lifting scheme is an efficient implementation of these filtering operations at each level when computing a discrete wavelet transform. So suppose that the low-resolution part of a signal at level $j-1$ is given and that it consists of a data set which we denote by λ_{j-1}. This set is transformed into two other sets at level j: the low-resolution part λ_j and the high-resolution part γ_j. This is obtained first by just splitting the data set λ_{j-1} into two separate data subsets. Traditionally this is done by separating the set of even samples and the set of odd samples. Such a splitting is sometimes referred to as the *lazy wavelet transform*. Doing just this of course does not improve our representation of the signal. Therefore, the next step is to recombine these two sets in several subsequent lifting steps which decorrelate the two signals.

The lifting steps usually come in pairs of a primal and a dual lifting step. A *dual lifting step* can be seen as a *prediction*: the data γ_j are predicted from the data in the subset λ_j. When the signals are still highly correlated, such a prediction will usually be very good, and thus we do not have to keep this information in both signals. That is why we can store only the part of γ_j that differs from its prediction (the prediction error). Thus γ_j is replaced by $\gamma_j - P(\lambda_j)$, where P represents the prediction operator. This represents the real decorrelating step.

However, the new representation has lost certain basic properties which one usually wants to keep, for example the mean value of the signal. Moreover, the simply subsampled λ_j data may suffer from Gibbs (ringing) artefacts. To restore these properties, one needs a *primal lifting step*, whereby the set λ_j is updated with data computed from the (new) subset γ_j . Thus λ_j is replaced by $\lambda_j + U(\gamma_j)$, where U is an updating operator.

In general, several such lifting steps can be applied in sequence to go from level $j-1$ to level j. To summarize, let us consider a simple lifting scheme with only one pair of lifting steps to go from level $j-1$ to level j.

- ⋆ **Splitting** (*lazy wavelet transform*) Partition the data set λ_{j-1} into two distinct data sets λ_j and γ_j.
- ⋆ **Predict** (*dual lifting*) Predict the data in the set γ_j by the data set λ_j and replace γ_j with the prediction error:
$$\gamma_j = \gamma_j - P(\lambda_j).$$
- ⋆ **Update** (*primal lifting*) Update the data in the set λ_j by the data in set γ_j:
$$\lambda_j = \lambda_j + U(\gamma_j).$$

These steps can be repeated by iterating on λ_j, thus creating a multi-level transform or multiresolution decomposition.

One of the great advantages of the lifting-scheme implementation (see Fig. 1.11) of a wavelet transform is that it decomposes the wavelet filters into extremely simple elementary steps, and each of these steps is very easily invertible. As a result, the inverse wavelet transform can always be obtained immediately from the forward transform. The inversion rules are obvious: revert the order of the operations, invert the signs in the lifting steps, and replace the splitting step by a merging step. Thus, inverting the three step procedure above results in:

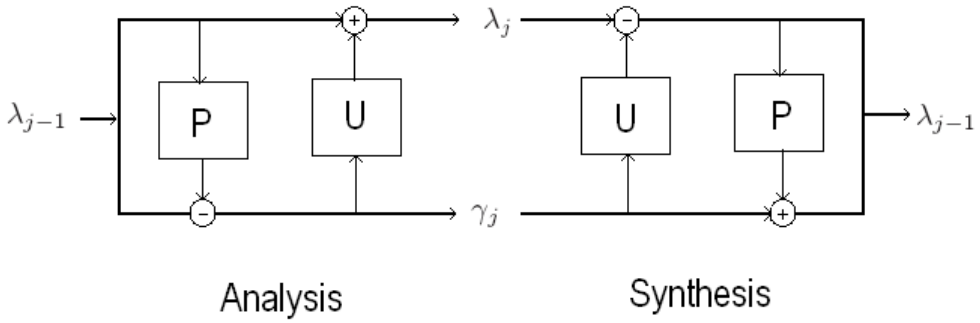


Figure 1.11: One-level lifting scheme for 1-D signals.

⋆ **Undo Update**

$$\lambda_j = \lambda_j - U(\gamma_j).$$

⋆ **Undo Predict**

$$\gamma_j = \gamma_j + P(\lambda_j).$$

⋆ **Merge**

$$\lambda_{j-1} = \lambda_j \cup \gamma_j.$$

1.3.2 Lifting advantages

Some of the advantages of the lifting wavelet implementation with respect to the classical wavelet transform are:

⋆ simplicity: it is easier to understand and implement.

⋆ the inverse transform is obvious to find and has exactly the same complexity as the forward transform.

⋆ the in-place lifting computation avoids auxiliary memory requirements since lifting outputs from one channel may be saved directly in the other channel. Such implementation considerations are explained in [178].

⋆ FIR decomposition: in [60] is shown that every biorthogonal wavelet transform with FIR filters can be decomposed into a finite number of lifting steps, followed by possible multiplication constants[1].

⋆ can be used on arbitrary geometries and irregular samplings.

Because the wavelets inherently provide a hierarchical representation of the analyzed content and also have proved very attractive for spatial and quality scalability in still image coding, an intense effort has been deployed in the last years to extend these decompositions in the temporal direction. Moreover, the implementation simplicity through lifting and the improvements brought to this scheme made possible the rapid evolution of the MCTF-based coding methods, as it will be shown in section 2.3.1. As during this

[1]Filter normalization factors, for which two computation methods will presented in section 3.2.2.

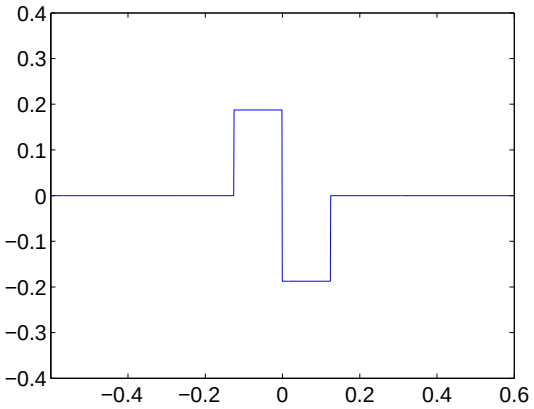


Figure 1.12: Haar wavelet.

thesis dissertation we will use the lifting implementations of Haar, Le Gall's 5/3 [78] and Daubechies 9/7 [19, 47] filter banks, we present in the following their corresponding lifting steps.

1.3.3 Lifting implementations of some wavelet filter banks

As mentioned above, in this thesis we will be using mainly three filters: Haar, Le Gall's 5/3 [78] and Daubechies 9/7 [19, 47]. Their corresponding lifting steps for one transform level of a discrete one-dimensional signal $x = \{x_k\}$ are presented in the following. The splittting step is the same for all the filters to present:

- ⋆ *Split* the signal x (i.e. λ_{j-1}) into even samples (i.e. λ_j) and odd samples (i.e. γ_j):

$$s_j[n] = x_{j-1}[2n]$$
$$d_j[n] = x_{j-1}[2n+1]$$

Haar analysis lifting steps:

- ⋆ *Predict*:

$$d_j[n] = \frac{\sqrt{2}}{2}(d_j[n] - s_j[n])$$

- ⋆ *Update*:

$$s_j[n] = \sqrt{2}s_j[n] + d_j[n]$$

Le Gall's 5/3 analysis lifting steps:

- ⋆ *Predict*:

$$d_j[n] = d_j[n] - \frac{1}{2}(s_j[n] + s_j[n+1])$$

- ⋆ *Update*:

$$s_j[n] = s_j[n] + \frac{1}{4}(d_j[n-1] + d_j[n])$$

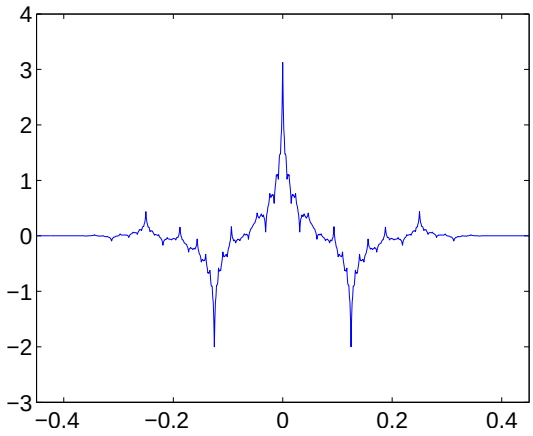


Figure 1.13: Le Gall's 5/3 wavelet.

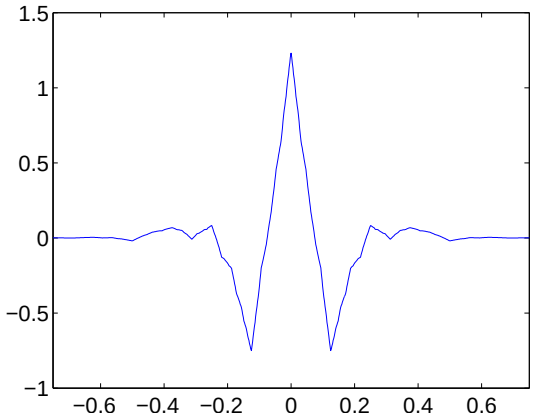


Figure 1.14: Daubechies 9/7 wavelet.

Scaling:

$$s_j[n] = \sqrt{2}s_j[n]$$
$$d_j[n] = \frac{\sqrt{2}}{2}d_j[n]$$

Daubechies 9/7 analysis lifting steps: One level wavelet decomposition using the Daubechies 9/7 contains two lifting stages each of each consisting a predict and update operation

⋆ *Predict*$_1$:

$$d_j[n] = d_j[n] + \alpha(s_j[n] + s_j[n+1])$$

⋆ *Update*$_1$:

$$s_j[n] = s_j[n] + \beta(d_j[n-1] + d_j[n])$$

⋆ *Predict*$_2$:

$$d_j[n] = d_j[n] + \gamma(s_j[n] + s_j[n+1])$$

⋆ *Update*$_2$:

$$s_j[n] = s_j[n] + \delta(d_j[n-1] + d_j[n])$$

Scaling:

$$s_j[n] = \zeta s_j[n]$$
$$d_j[n] = \frac{1}{\zeta} d_j[n]$$

$$\alpha = -1.586134342$$
$$\beta = -0.05298011854$$
$$\gamma = 0.8829110762$$
$$\delta = 0.4435068522$$
$$\zeta = 1.149604398$$

1.4 Conclusion

In this chapter we have reviewed the basic concepts of multiresolution analysis and wavelets. Moreover, we have introduced the lifting mechanism, which provides a framework for implementing the classical wavelet transforms. It has several advantages over the classical filter bank schemes and provides additional features, like implementation simplicity and in-place computation, and it will be the instrument used for wavelets implementation during this thesis work. We propose in the following to review scalable video-coding concepts by presenting some recent developments in both predictive and MCTF-based scalable-coding strategies.

Chapter 2

Scalable video coding

The developments in networking technologies and video coding over the last decade have enabled the broadcasting of video signals over various networks. Internet, mobile wireless, broadcast, video-on-demand and other potential applications and clients have different demands on video quality and different conditions for video receiving and decoding. A conventional video-coding system for storage purposes encodes a video sequence at a desired, fixed bitrate which is adequate for a given application. Serving different clients generally requires transcoding the given video sequence. Furthermore, some particular applications, like the transmission over the Internet, can even change demands on video bitrate during a single video-sequence transmission. Scalable video coding provides a straightforward solution for a universal video-coding system that can serve a broad range of applications. Over the last decades, intensive research activities on diverse algorithms for scalable video coding were undertaken and have finally reached their mature phase. In this chapter an overview of the requirements and the current status of scalable video coding are given.

2.1 Video coding scalability degrees

There are several types of scalability which should be supported by a scalable video coding system. However, a set of basic scalability types can be defined:

- ⋆ spatial or resolution scalability
- ⋆ temporal or framerate scalability
- ⋆ SNR or quality scalability

Scalable coding should support combinations of basic scalabilities, i.e. combined scalability, which we denote as full scalability (see Fig. 2.1). Moreover, scalable video coders may include:

- ⋆ complexity scalability
- ⋆ region of interest scalability
- ⋆ object based scalability

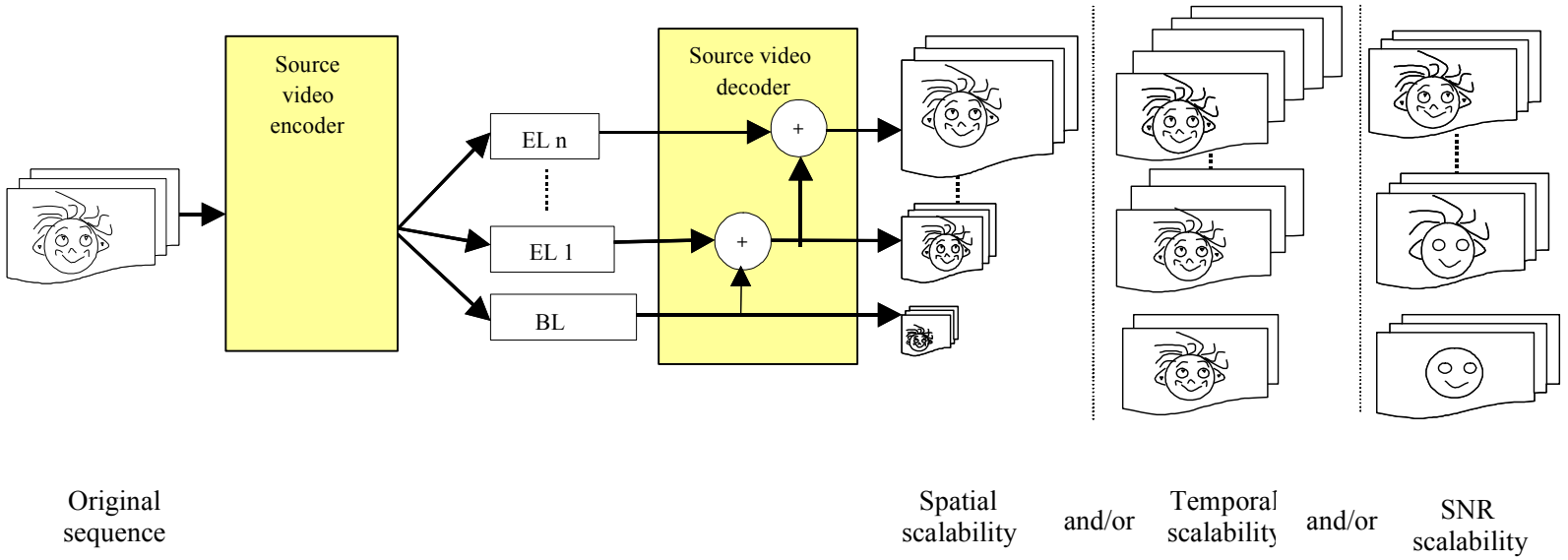
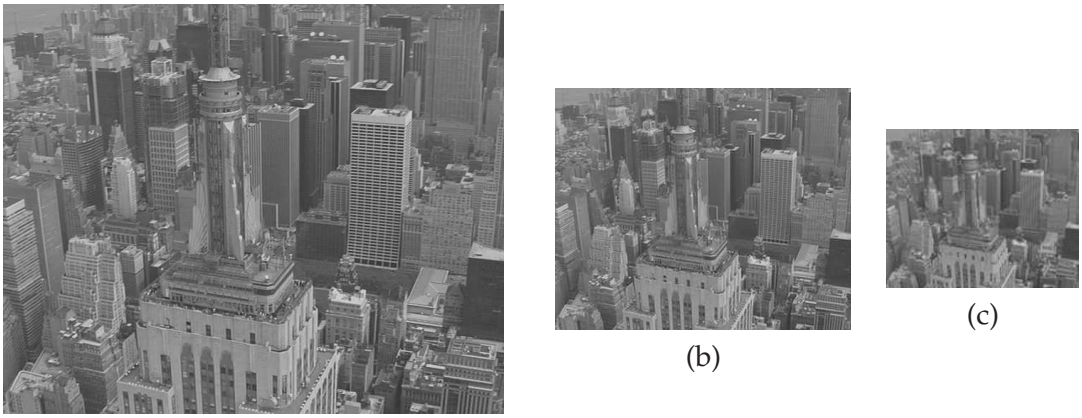


Figure 2.1: Global structure of a layered scalable video-coding scheme.

and other features such as: support for both progressive and interlaced material, support for various colour depths (including component scalability) and robustness to different types of transmission errors. We propose in the following to define and exemplify each of these three key factors in scalable video coding.

2.1.1 Spatial scalability

Spatial scalability represents the ability of a media file or picture image to reduce or vary the image geometry (i.e., height and width) without significantly changing the quality of the image.



(a) (b) (c)

Figure 2.2: City frame in: (a) 4CIF format (704x576 pixels), (b) CIF format (352x288 pixels) and (c) QCIF format (174x144 pixels).

As shown in Fig. 2.2, a video sequence can be represented in several resolution formats. With a spatially scalable codec, this can be obtained by simply encoding a video sequence at the highest resolution level and extracting the bitstream corresponding to a certain level of spatial scalability in function of user constaints (i.e., display format).

Spatial scalability has many applications. It may be used in certain HDTV[1] systems to maintain compatibility with standard definition TV. For transmitting video over dual-priority networks, we can transmit a low-resolution version of the video over the high-priority channel and an enhancement layer over the low-priority channel. Also, one solution for video transmission over bandwidth-constrained channels is to transmit a low-resolution version of the video. For browsing a remote video database, it would be more economical to send low-resolution versions of the video clips to the user, and then, depending on his or her interest, progressively enhance the resolution.

2.1.2 Temporal scalability

Temporal scalability is the ability of a streaming media program or moving-picture file to reduce or vary the number of images or data elements representing that media file for a particular time period (temporal segment) without significantly changing the quality or resolution of the media over time. In other words, for a video sequence, it consists in decreasing / increasing the temporal frequency from the encoded bitstream (i.e., the

[1]High Definition Television

ability of encoding a sequence such that it can be extracted at different frame rates). In Fig. 2.3, an initial temporal frequency for the City sequence (8Hz) is decreased at 4Hz by simply decoding one frame out of two.

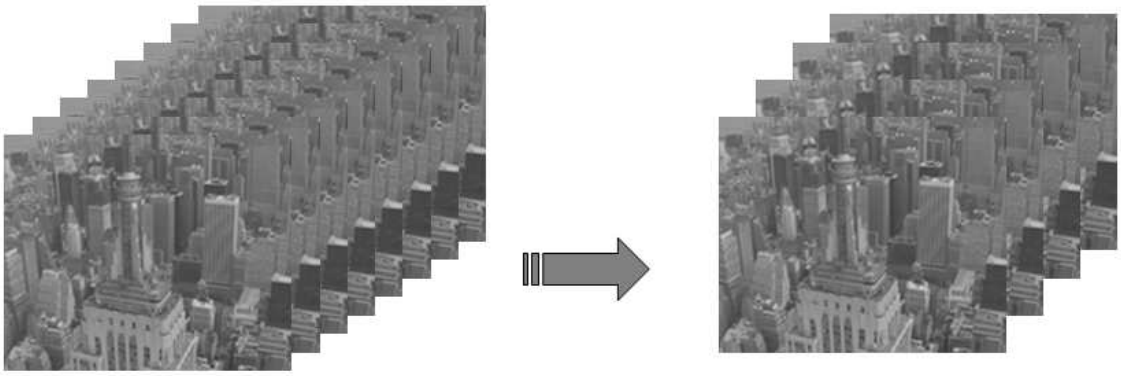


Figure 2.3: Temporal scalability example: framerate reduction from 8Hz (left) to 4Hz (right)

Because the human visual system is less sensitive to temporal details and more sensitive to spatial details when a video recording depicts a stationary scene, if a video encoder supports temporal scalability, we can decide to lower the frame rate in exchange for an increase of the bitrate and hence, we obtain an improvement in the spatial quality of the remaining frames. This way an optimal scalability tradeoff is achieved with satisfactory results for the user. Together with the spatial frexibility, the frame rate can be varied in order to fullfill the bandwidth-constrained transmision channels or for browsing mostly static video surveillance sequences.

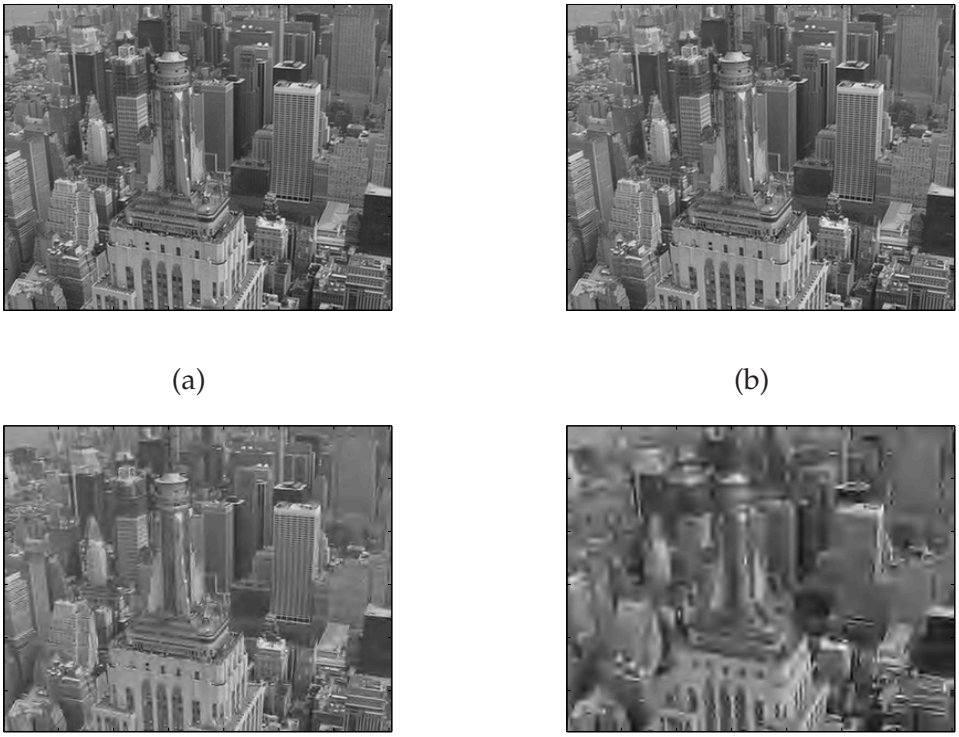
2.1.3 Quality scalability

Quality or SNR scalability consists of the ability to reconstruct a signal with a different amount of information. In video coding, this can be translated to the ability of a system to encode a sequence once and to extract the encoded bitstream at different bitrate constraints. This can be done by simply varying gradually the pixel representation precision or bitplane coding.

In Fig. 2.4 an example of variable SNR is drawn for the City (CIF@30Hz) video sequence, where the initial bitstream was truncated at 750, 350 and 180 kbs respectively .

2.2 Scalable predictive coding

We have seen in Section 2.1 the principles and requirements of scalable video coding. In the following, we propose to continue with the scalable hybrid-coding field, by recalling the guiding principles of hybrid video-coding, which is the basis for MPEG and ITV[1] codecs. Starting from the general structure of a hybrid codec, passing through a short overview of scalability in previous standards, we shall finally present the current status



(a) (b)

(c) (d)

Figure 2.4: City frame (352×288) at: (a) original , (b) 750 kbs, (c) 350 kbs and (d) 180 kbs.

of scalable predictive coding.

2.2.1 General structure of a hybrid codec

Current standards, such as H.263 [7], H.264 [1], MPEG-2 [4] and MPEG-4 [2] (both part 2 and part 10) are based on a predictive video-coding scheme (see Fig. 2.5).

The predictive coding is made in a closed loop: a decoder is integrated at the encoder side such that the reconstructed frames are used for prediction of the current frames, leading thus to a closed-loop structure (the part enclosed in the red rectangle in Fig. 2.5). The usual encoding process has three main stages, namely the temporal processing and spatial decorrelation stages, both reducing the temporal and spatial redundancy in a video shot respectively, and finally an entropy codec. In the following, we propose to review the main functionalies of each stage.

2.2.1.1 Temporal processing

The temporal decorrelation of a video sequence is generally a two-step process described by motion estimation and compensation. To achieve compression, the temporal redundancy between adjacent frames can be exploited. That is, a frame is selected as a reference, and subsequent frames are predicted from the reference using a technique known

[1]Internet TeleVision

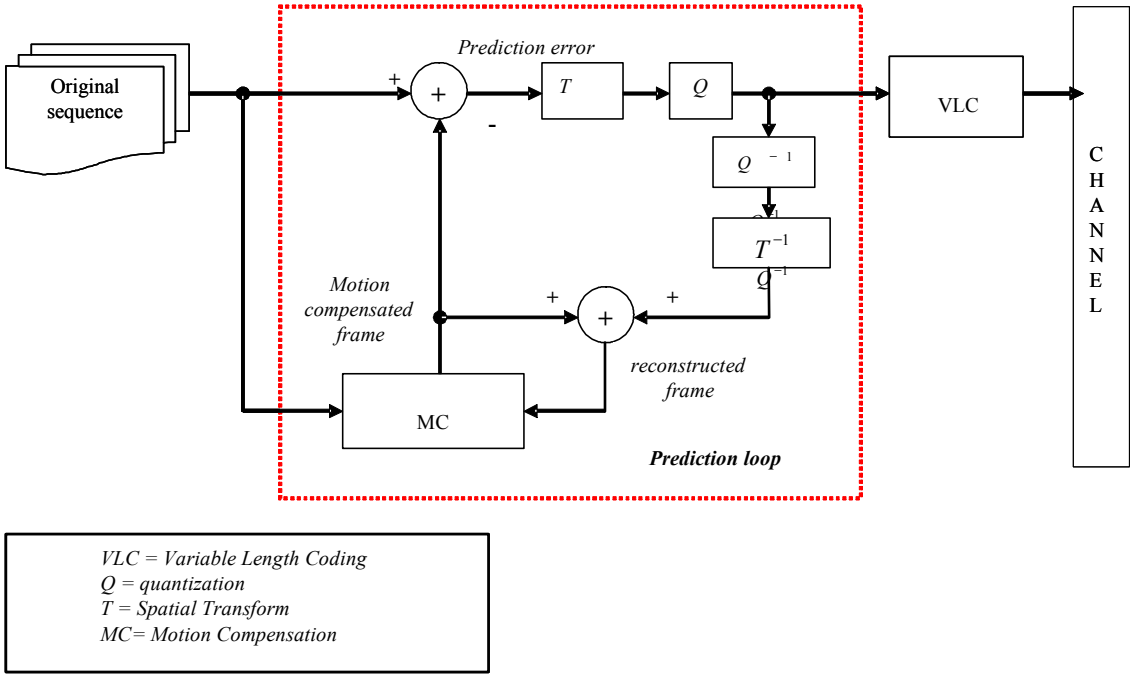


Figure 2.5: Predictive (hybrid) video coding scheme.

as *motion estimation*. This is the process of finding optimal or near-optimal motion vectors (MVs).

In block-based motion estimation (BME), the frames are partitioned in blocks of pixels (blocks of 8×8 or macroblocks of 16×16 pixels for example). Each block is predicted from a block of equal size in the reference frame. The blocks are not transformed in any way apart from being shifted to the position of the predicted block. This shift is represented by the motion vector. The amount of prediction error for a block is often measured using the mean squared error (MSE), the sum of absolute differences (SAD) or the mean absolute error (MAE) between the predicted and actual pixel values over all pixels of the motion-compensated region. To find optimal motion vectors, one basically has to calculate the prediction error for each motion vector within a certain search range and pick the one that has the best compromise between the amount of error and the number of bits needed for the motion-vector data. The motion-estimation technique of simply exhaustively testing all possible motion representations to perform such an optimization is called full search. Because the motion-estimation process is the most expensive from computational point of view, faster methods, which are near-optimal with respect to rate-distortion criterion are developed, by using a coarse search grid for a first approximation and refining the grid in the surrounding of this approximation in further steps. Several techniques have been proposed in an effort to reduce the motion estimation complexity, such as the Three-Step Search (TSS) [106], New Three Step Search (NTSS) [116], Diamond Search (DS) [220], Motion Vector Field Adaptive Search Technique (MVFAST) [122] or Predictive Motion Vector Field Adaptive Search Technique (PMVFAST) [187], all being block-based motion-estimation methods.

For overlapped block motion estimation (OBME) [115], the prediction errors of a block and its overlapping neighbouring blocks have to be weighted and summed according to a window function before being squared. As in the process of successively

finding/refining motion vectors some neighbouring MVs are not known yet, the corresponding prediction errors can be ignored (not added) as a sub-optimal solution. The major disadvantage of OBME is the increased computational complexity and the fact that prediction errors and, thus, also the optimal motion vectors depend on neighbouring blocks/motion vectors. Therefore, there is no algorithm with polynomial computational complexity that guarantees optimal motion vectors. However, there are near-optimal iterative and non-iterative methods with acceptable computational complexity. Moreover, the motion-estimation process is the most computation-intensive part of an encoder and can cause visual artefacts when subject to errors.

Once the motion-estimation process is finished and the motion-vector fields are know, the reference frames are *compensated* in the direction of the motion flow. This process, known as *motion compensation*, consists of the prediction of one frame from previous or next frames, such that only the motion vectors and the prediction error are coded. In a predictive codec, the current frames are predicted from the previous coded and reconstructed frames, in order to minimize the drift effect caused by the quantization process. When a previous frame is used as a reference, the prediction is referred to as forward prediction, and, in hybrid coding, the predicted frame is called an inter-P (predictive) frame. If the reference frame is a future frame, then the prediction is referred to as backward prediction. Backward prediction is typically used with forward prediction, and this is referred to as bidirectional prediction (i.e., inter-B (bidirectional) frame). There are also frames which are not motion-compensated. In predictive coding, they are denoted as I (intra) frames and are used for the prediction of the inter-P and B frames. Usually the prediction chain (the frames between two intra I frames) has a fixed length, as can be seen in Fig. 2.6. For example, AVC/H.264 [1] offers the possibility of choosing from multiple reference frames for motion estimation, meaning that the codec can decide whether simply to refer to the previous frame or even to a frame before that. Because of that (usually, a P-Frame can refer to a frame before the latest I-frame), a new frametype had to be introduced: IDR-frames, which are a special type of I-frame to which no following frame is allowed to refer to. Thus, scene cutting is possible only at the IDR-frames.

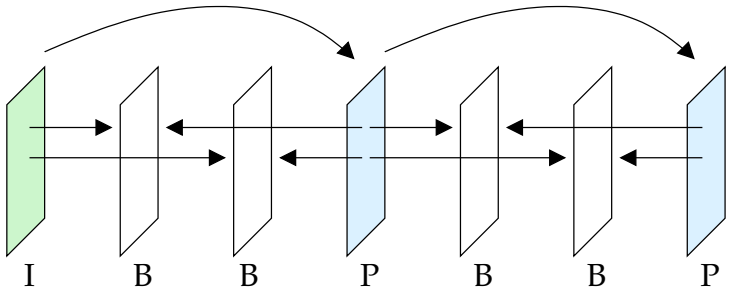


Figure 2.6: Video-shot prediction chain in hybrid coding

2.2.1.2 Spatial processing

The prediction residual images are spatially decorrelated in order to exploit their spatial redundancy. In hybrid coding, the employed transform is generally an 8×8 block-based Discrete Cosine Trasform (DCT) or a 4×4 and 8×8 block-based Hadamard transform. Starting with the second version of H.263 (or H.263+ [7]), before applying a spatial transform, a spatial-domain prediction is used for improving the efficiency of intra video cod-

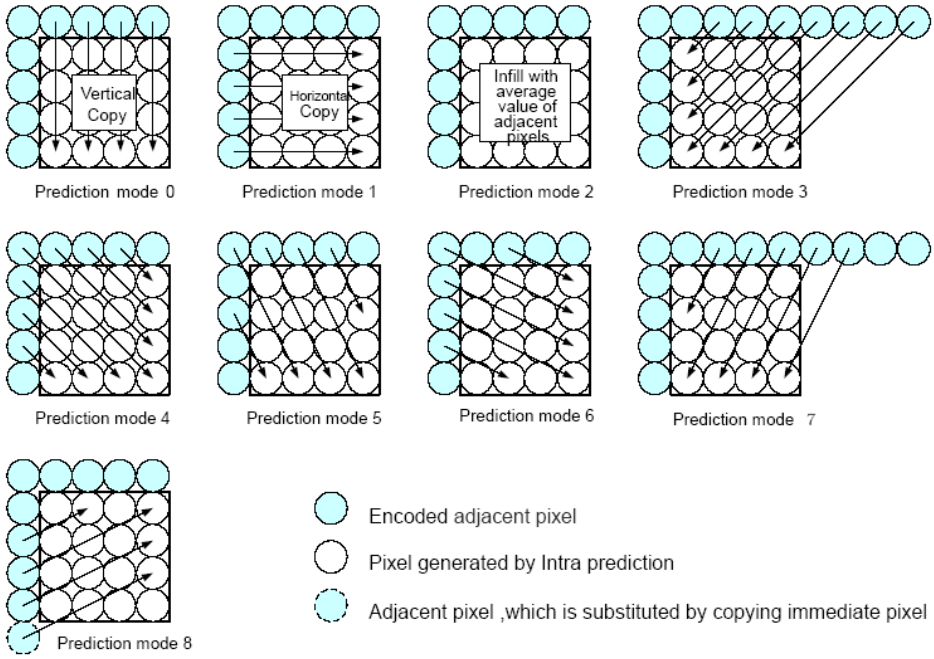


Figure 2.7: Intra-prediction modes for a 4×4 block in H.264.

ing. Basically the scheme predicts and generates the pixel values from adjacent pixel values that have already been encoded. H.264 [1] provides nine modes for luminance signal prediction and four modes for color, designed in a directional manner, as shown in Fig. 2.7. The key point of the effective improvement is to select the proper prediction mode for each block, and this is done based on a rate-distortion criterion.

2.2.1.3 Entropy coding

After the spatial-coding module, the resulting coefficients are passed to a quantizer. By quantization, they suffer some loss of information. The quantization consists of dividing the spatial coefficients matrix by another, called quantization matrix, which contains coefficients selected by the coder, depending on a quality control parameter, Q. The goal is to attenuate the high frequencies, i.e. those to which the human eye is less sensitive. The quantized coefficients are then zig-zag parsed following their magnitude and finally coded using an entropy coder, such as Run-Length Coding (RLC) [130], Huffman coding [102], Context-based Adaptive Variable-Length Coding (CAVLC) [119] or Context-based Adaptive Binary Arithmetic Coding (CABAC) [128]. The motion-vector fields are losslessly coded, using the Variable Length Coding (VLC) [90] method.

The hybrid video-coding strategies can achieve high compression rates, but they are not able to provide a direct scalable representation. However, both MPEG-2 and MPEG-4 (part 2) have a layered predictive structure, able to supply a coarse scalability form, where each layer represents a version of the video sequence to a time-space resolution and a given flow. In the absence of this layered structure, it is not possible to modify the flow, the space resolution, or the temporal frequency of a compressed video sequence without transcoding. This operation requires the complete decoding and a re-encoding

of the whole video sequence and is generally very expensive from the computational complexity point of view. Many strategies [13, 140] have been developed however in order to decrease this computational burden.

2.2.2 Scalability evolution through standardization

In early video compression standards such as ITU-T H.261 [6] and ISO/IEC MPEG-1 [3], no scalability mechanisms were provided. One reason for this was the dedicated design for specific applications such as conversational services or storage which did not require scalability. In fact, scalability can nevertheless be achieved by providing different bitstreams targeting at different decoded resolutions: the method of simulcast ties together two or several streams for the purpose of parallel transmission. Parallel storage could also be implemented. ISO/IEC MPEG-2 [4], which is identical to ITU-T H.262, was the first general-purpose video compression standard which also included a number of tools providing scalability. One of the reasons was the desire for forward compatibility with MPEG-1, where eventually base information could be encoded and decoded by the old standard, while higher-quality enhancement information is processed by the new standard [138]. MPEG-2 was the first standard to include implementations of layered coding, where the standalone availability of enhancement information (without the base layer) is useless, because differential encoding is performed with reference to the base layer. All dimensions of scalability as mentioned above are supported (spatial, temporal, SNR); however, the number of scalable bitstream layers is generally restricted to a maximum of three in any of the existing MPEG-2 profiles. In addition, data partitioning allows the separation of the bitstream into different layers, according to the importance of the underlying elements for the quality of the reconstructed signal.

The video codec of the ISO/IEC MPEG-4 standard [2] provides even more flexible scalability tools, including spatial and temporal scalability within a more generic framework, but also SNR scalability with fine granularity and scalability at the level of (eventually semantic) video objects. In Simple Profile (SP) mode, MPEG-4 video is equivalent to the ITU-T H.263 [7] baseline codec, which provides no scalability. Extensions of H.263 define spatial, temporal and SNR scalabilities as well. Advanced Video Coding (AVC), as defined as part 10 of the MPEG-4 standard [1], aka ITU-T H.264 AVC, can, in principle, be run in different temporal scalability modes, due to its flexibility in the definition of prediction frame references.

The basic idea in MPEG-4 (part-2) Fine Granularity Scalability (FGS) is to encode a video sequence into a non-scalable base layer and a scalable enhancement layer. The MPEG-4 Advanced Simple Profile (ASP) provides a subset of non-scalable video-coding tools to achieve high coding efficiency for the base layer. The bitrate of the base layer is the lower bound of a bitrate range that FGS supports. The base layer is typically encoded at a very low bitrate. The FGS profile is used to obtain the enhancement layer for achieving optimized video quality with a single stream for a wide range of bitrates. More precisely, each frame residue (i.e., the difference between the original frame and the corresponding frame reconstructed from the base layer) is encoded for the enhancement layer in a scalable manner: DCT coefficients of the residue are compressed bitplane-wise from the most significant bit to the least significant bit. For a temporal enhancement frame which does not have a corresponding frame in the base layer, the bitplane coding is applied to the entire DCT coefficients of the frame. This is called FGS temporal scalability (FGST). FGST can be encoded using either forward or bidirectional prediction from the

base layer. MPEG-4 FGS provides very fine grain scalability to allow near RD-optimal bitrate reduction. More details about FGS can be found in [118, 5].

Nevertheless, it must be noted that any of the video-coding standards existing so far restricts scalability at the bitstream level to a predefined number of layers which must be known at the time of encoding. Even if the MPEG-4 FGS provides a scalable bitstream, a problem of drifting may emerge due to differences in the reference frames used in the encoder and decoder. In general, the bitrate of the base-layer is low enough to fit in the minimum network bandwidth. Therefore, the base layer is always available at the decoder. However, since the high-quality references in FGS comprise part of the DCT coefficients encoded in the enhancement layers, more bandwidth is needed to transmit them to the decoder. When channel bandwidth somehow drops, the decoder may partially or completely loose the high-quality references. In this case, the decoder has to use the low-quality references instead, which would inevitably cause the drifting error in the enhancement layer. In the following, we examine the specifications of the scalable video coding (SVC) extention of the current MPEG-4 AVC standard which intends to overcome the drifting issue present in MPEG-4 FGS.

2.2.3 Scalable Extension of AVC: H.264/MPEG-4 SVC

To serve different needs of users with different displays connected through different network links by using a single bitstream, a single coded version of the video should provide spatial, temporal and quality scalability. MPEG and ITU standardization organizations launched a joint call for proposals [12] in 2003 aiming at the creation of a new standard for scalable video coding: the SVC standard (*Scalable Video Coding*) [164]. This will be standardized as an amendment to MPEG-4 Part 10 AVC/ITU-T H.264.

As a distinctive feature, SVC allows generation of an H.264 /MPEG-4 AVC compliant, i.e., backwards-compatible, base layer and one, or several, enhancement layer(s). Each enhancement layer can be turned into an AVC-compliant standalone (and not anymore scalable) bitstream, using built-in SVC tools. The base-layer bitstream corresponds to a minimum quality, frame rate, and resolution (e.g., QCIF video), and the enhancement-layer bitstreams represent the same video at gradually increased quality and/or increased resolution (e.g., CIF) and/or increased frame rate. The texture coefficients are initially encoded with a coarse quantization step, the resulting quantization error being re-quantized with a finer step, and so on in a progressive way, thus allowing a medium-grain quality scalability. Moreover, there is a mechanism of prediction between the various enhancement layers, allowing the reuse of textures and motion-vector fields obtained in preceding layers.

The basic SVC design can be classified as a layered video codec. In general, the coder structure, as well as the coding efficiency, depends on the scalability space that is required by an application. For illustration, Fig. 2.8 shows a typical coder structure with two spatial layers. In each spatial, or coarse-grain, SNR layer, the basic concepts of motion-compensated prediction and intra prediction are employed as in H.264/MPEG4-AVC. The redundancy between different layers is exploited by additional inter-layer prediction concepts that include prediction mechanisms for motion parameters as well as texture data (intra and residual data). A base representation of the input pictures of each layer is obtained by transform coding similar to that of H.264/MPEG4-AVC, the corresponding NAL [1] units contain motion information and texture data. The NAL units of the lowest

[1] Network Abstraction Layer

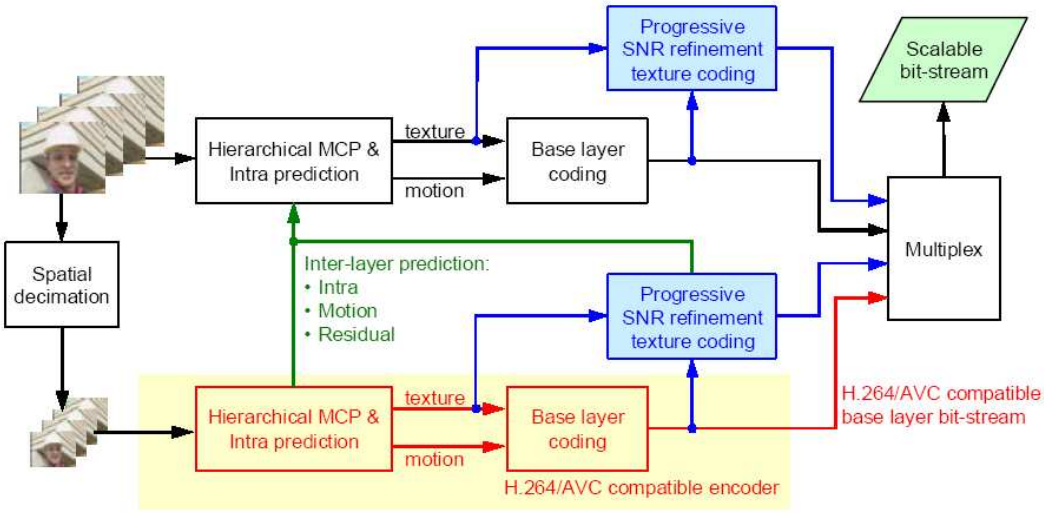


Figure 2.8: H.264/MPEG-4 SVC coding scheme.

layer are compatible with single-layer H.264/MPEG4-AVC [1].

The reconstruction quality of these basic representations can be improved by an additional coding of so-called progressive-refinement slices. In contrast to all other slice-data NAL units, the corresponding NAL units can be arbitrarily truncated in order to support fine grain quality scalability or flexible bit-rate adaptation.

An important feature of the SVC design is that scalability is provided at the bitstream level. Bitstreams for a reduced spatial and/or temporal resolution can be simply obtained by discarding NAL units (or network packets) from a global SVC bitstream that are not required for decoding the target resolution. NAL units of progressive refinement slices can additionally be truncated in order to further reduce the bitrate and the associated reconstruction quality. In order to assist easy bitstream manipulations, the one-byte NAL unit header of H.264/MPEG4-AVC was extended by 2 bytes for SVC NAL units. These additional bytes signal whether the NAL unit is required for decoding a specific spatio-temporal resolution and quality (or bitrate), as well as whether the NAL unit can be truncated.

Coding efficiency of SVC depends on the application requirements but the goal is to achieve a rate-distortion performance that is comparable to non-scalable H.264 / MPEG-4 AVC. The design of the scalable H.264/MPEG4-AVC extension and promising application areas are pointed out in [164].

2.3 Scalable lifting-based wavelet coding

In parallel with the hybrid-coding schemes, a new coding paradigm has been developed: the scalable lifting-based wavelet (subband) coding. The general structure for a scalable interframe wavelet-based video-coding system is presented in Fig. 2.9. It is based on two key technologies: motion compensated temporal filtering (MCTF) and spatial wavelet transform. The wavelet-based compression schemes have become increasingly important and gained widespread acceptance, an example being the JPEG2000 still image compression standard [8, 178]. Because of their inherent multiresolution signal representa-

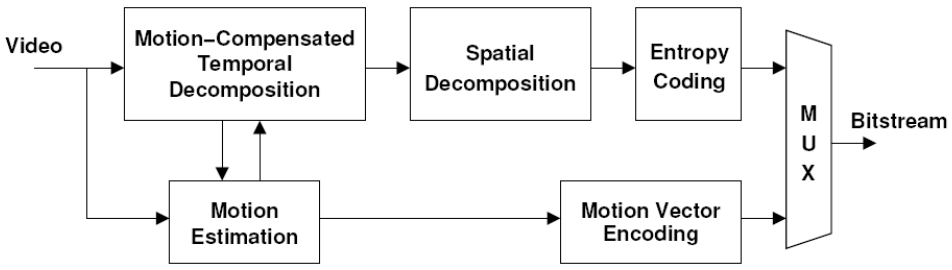


Figure 2.9: Encoder structure of scalable interframe wavelet-based video-coding system

tion, wavelet-based coding schemes have the potential to support temporal, spatial and SNR scalability. This is the reason for which we have chosen the scalable lifting-based wavelet-coding paradigm as the conceptual development framework for this thesis.

In MCTF-based codecs, instead of performing temporal decorrelation along the direct temporal-axis of the input video, the temporal filtering is performed along the motion trajectories in order to remove the temporal redundancy. In this way, the energy of the temporal highpass subbands decreases substantially. Fewer motion artefacts will also increase coding efficiency. So motion compensated temporal filtering plays an essential role in motion compensated $t+2D$ subband/wavelet coding. It will influence the coding efficiency and temporal scalability features.

In the following we will introduce the MCTF concept as well as several MCTF schemes, and we will finally present the interesting family of wavelet-based coders.

2.3.1 Motion-Compensated Temporal Filtering

The idea of temporal extensions of subband decompositions appeared in the late 80's, with the works of Karlsson and Vetterli [101] and Kronander [109]. In these works, the classical temporal closed-loop prediction scheme was replaced by a temporal subband decomposition, which, at that time, did not take into account any motion compensation. However, it was shown [135] that the prediction in the motion direction leads to important energy reduction in the temporal detail subbands, thus a much better compression performance and visual quality. Consequently, there has been significant interest in motion-compensated temporal filtering in which the temporal transform follows the motion trajectories. MCTF plays an essential role in motion compensated $t+2D$ subband/wavelet coding, influencing both the coding efficiency and the temporal scalability.

The current 3D wavelet video-coding schemes involving MCTF can be divided into two main categories. The first one performs MCTF on the input video sequence directly in the full-resolution spatial domain before spatial transform, is often referred to as *spatial domain* or classical MCTF, and is usually denoted by $t+2D$ subband coding. The second one performs MCTF in wavelet subband domain generated by the spatial transform, being often referred to as *in-band* or wavelet domain MCTF and is generally denoted by $2D+t$. Fig. 2.16 illustrates a general framework for the above-mentioned schemes. We propose to review in the following several MCTF-based coding schemes, starting with the classical ones where the temporal decomposition is firstly performed in the spatial domain ($t+2D$) and finishing with the wavelet-domain MCTF ($2D+t(+2D)$). Before in-

troducing the motion-compensated prediction/update methods, we propose to overview several motion-estimation strategies, which are adapted to the MCTF setting.

2.3.1.1 Some motion-estimation strategies

The key step in removing the temporal redundancy is motion estimation, where a motion vector is predicted between the current frame and a reference frame, based on some error minimization criterion (e.g. mean square error (MSE), sum of absolute difference (SAD), mean absolute error (MAE)). Following motion estimation, a motion-compensation stage is applied to obtain the residual image, i.e. the pixel differences between the current frame and the predicted frame. As the result of motion estimation is directly reflected by the energy of the residual frames, it is important to have a good motion estimator. The traditional motion estimation algorithm uses a Full Search (FS), where every possible displacement within a search region is searched, but this is computationally expensive. Since, in most cases, motion estimation constitutes roughly 70% of the computational load of a video encoder, there is a need for fast, simple and efficient motion-estimation algorithms. Many fast algorithms were proposed in order to diminish the computational complexity. Some fast search methods based on rectangular patterns and some geometric or shape-based patterns have been proposed so far. There are three main types of motion-estimation algorithms, namely the pel-based, block-based and object-based methods. A description and a performance comparison between these algorithms for video compression can be found in [126].

In the framework of wavelet-based video coders, several motion-estimation algorithms [68, 36, 15, 146, 139] have been developed. The block matching algorithms (BMA) are widely used to estimate the motion vectors because of their relatively simple implementation. However, since the boundaries of the moving objects do not usually coincide with the boundaries of the blocks used for the BMA, objects having different types of motion can exist in a block. In this case, the block can not be adequately compensated by employing a single motion vector. Hence, variable block size (VBS) motion-estimation techniques [154, 172] have been proposed to improve the performance of motion-compensated transform coding (MCTC). In VBS, the block size for estimating the motion is adapted according to the type of motion in the block. The VBS technique is known to be very effective for areas containing complex motions. In the conventional VBS motion estimation, like the Hierarchical Variable Size Block Matching (HVSBM) algorithm proposed by Choi and Woods in [44], various decision rules are used to form the VBS motion for quadtree structures. However, the problem with the conventional VBS technique lies in encoding the motion vectors efficiently with entropy coding [103, 172, 45] since the motion vectors within a block are observed to be quite different from each other.

The work in this thesis is partially implemented in the framework of MC-EZBC [210] and respectively Vidwav MSRA [211] video coders, which use as motion-estimation strategy the Hierarchical Variable Size Block Matching algorithm [44] and a multi-layered adaptive block-size motion alignment [213] respectively.

2.3.1.2 Connected and unconnected pixels

In block-based motion-compensated prediction, the same area in the reference frame can be used to predict several areas in the current frame, while some parts of the reference

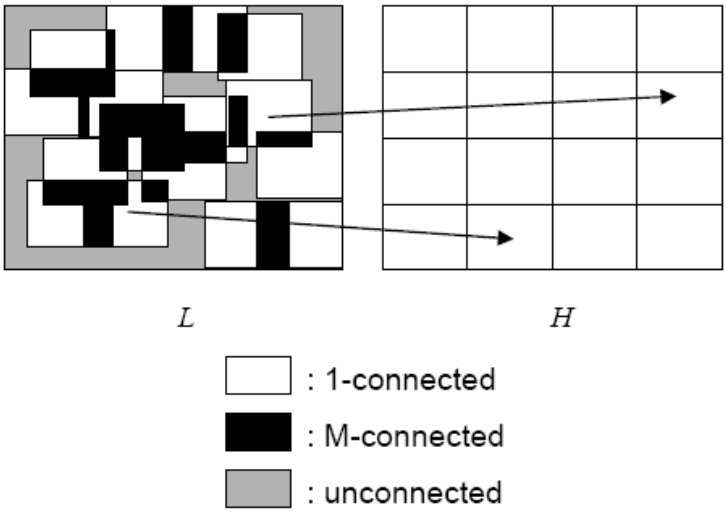


Figure 2.10: Motion compensated prediction: unconnected and multiple-connected areas

frame may not be used at all for prediction. This gives rise to *multiple connected* and *unconnected* pixels. With moving objects, some pixels in the first frame are covered in the second frame, but some pixels in the second frame are uncovered. This is called the occlusion effect [108]. The best matched pixels for the second frame are certainly not present in the to-be-covered pixels of the first frame; in other words, only a subset of the first frame should be used as the reference for the second frame. On the other hand, uncovered pixels in the second frame are not present in the first frame. It should be noted that the phrase *are not present* does not mean there do not exist pixels with the same luminance and chrominance components in the first frame, but it indicates that specific pixels, indicating real motion of objects, do not exist in the first frame. So connections between pixels in two frames are not one-to-one, and, thus, at some lattice sites, the motion field is not defined. Clearly, backward motion estimation should be turned off for uncovered pixels due to the lack of temporal coherence, but forward motion estimation should be turned on. It is an important task in MCTF to properly distinguish and handle covered and uncovered pixels including newly appearing objects in the scene.

In existing MCTF methods [135, 44] (see Fig. 2.10), the placement of unconnected pixels is related to the scan order. Furthermore, no test of motion-vector accuracy is used, so that the motion field is defined for each block of the second frame, even though some of these blocks may not have a true match in the first frame. These poor, or simply wrong, connections lead to faulty classification of connected and unconnected pixels. Hence, both the motion estimates and the decision of unconnected pixels are questionable. Sometimes the positions selected by such methods may be totally different from the real occurrence of the occlusion effect.

A consequence of poor pixel classification is the appearance of visual artefacts in the temporal low frame-rate video, as illustrated by Fig. 2.11, which shows an object in the first frame moving to the right. Here the covered background region in the first frame becomes uncovered in the second frame. Using backward motion estimation, the motion vectors of the uncovered region and the object may both point to the object in the first frame. Then according to the scan-order rule introduced earlier, the uncovered region in

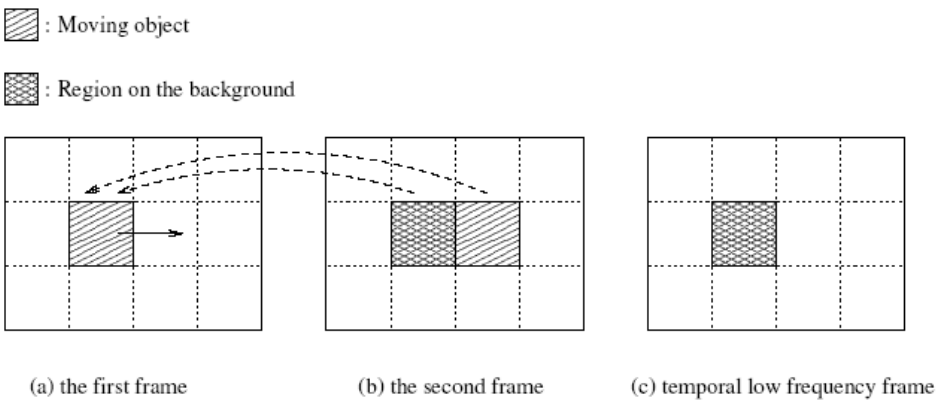


Figure 2.11: Wrong pixel classification on temporal approximation frames

the second frame will be connected with the object in the first frame, while the moving object is left unconnected. So the uncovered region will wrongly take part in temporal filtering, which will likely cause the temporal low-frequency frame to have visual artefacts, as we will also see in Section 3.1.1.

In a temporal scalable coder using MCTF, the most natural choice for the low frame-rate data is the MCTF output. Therefore, it is important for it to be artefact free and visually pleasant. Note that this is in contrast with the usual choice for the low frame-rate data in the case of hybrid coders, which is the sub-sampled frames themselves.

Chen *et al.* propose in [38] a simple way to detect uncovered pixels by doing backward and forward motion estimation for each block in the second frame. If forward motion estimation has a smaller displaced frame difference (DFD), then this block will be classified as uncovered. But since there are scene-illumination changes and noise, some blocks in the second frame will choose forward motion estimation, even though there are matched blocks in the first frame. Since those matched blocks in the first frame will not be used as reference, pixels in those blocks will be processed as unconnected pixels. For blocks in the second frame using forward motion estimation, their pixels will also be processed as unconnected. This will generate many more unconnected pixels in bidirectional as compared with unidirectional MCTF.

In order to avoid such problems, other motion models, such as meshes can be employed [166]. However, when the percentage of multiple connected or unconnected pixels is too high, a scene-cut adapted MCTF [189] can be used for processing the uncorrelated shots of the video sequence. Such scheme will be introduced in Section 3.1.

2.3.1.3 Classical MCTF: Haar and 5/3 MC lifting transforms

The simplest temporal wavelet transform is the Haar filter bank (Eq. 2.1), performing sums and differences on pairs of frames $(x(2t), x(2t+1))$ to obtain the approximation (L_t) and the detail (H_t) subbands. A Haar temporal decomposition is illustrated in Fig. 2.12 on a GOP (Group of Pictures) of eight frames, which allows a dyadic decomposition over a maximum three levels.

The basic operations for obtaining the high-pass and low-pass subbands in lifting form are as follows:

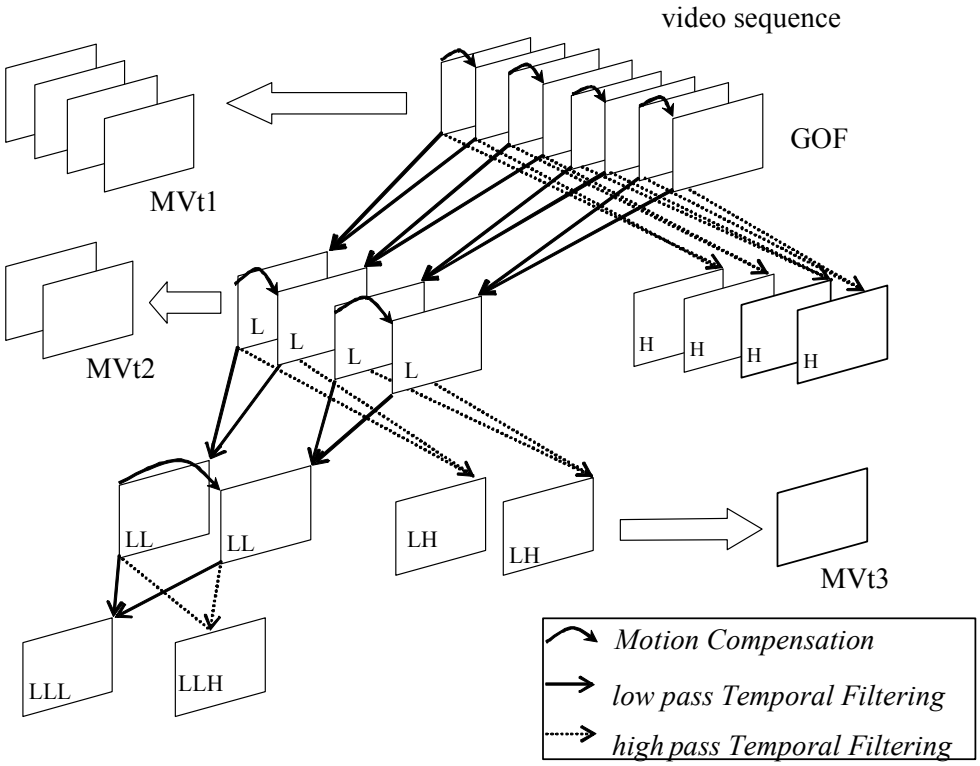


Figure 2.12: Motion compensated temporal decomposition of a 8-frames GOP using Haar wavelet.

$$\begin{cases} H_t(n) = x_{2t+1}(n) - x_{2t}(n) \\ L_t(n) = x_{2t}(n) + \dfrac{1}{2} H_t(n) \end{cases} \tag{2.1}$$

An overview of various MCTF structures for scalable video coding can be found in [137], and, in [150], some interesting lifting formulations of these temporal decompositions are presented.

Due to the two-tap low-pass and high-pass filters and the downsampling being made by a factor of 2, no boundary problems appear when decomposing a GOP of size 2^L into a number of up to L resolution levels. Moreover, if motion estimation and compensation is performed between pairs of successive frames, without overlapping, the number of operations and the number of motion vector fields are the same as for coding the same number of frames in a predictive scheme (and thus equal to $2^L - 1$). However, as the pairs of pixels have to be processed in successive frames in order to obtain the coefficients of the approximation and detail frames, motion invertibility becomes a very important problem.

In the temporal decomposition, motion estimation is first performed between input frames and the motion vector fields (denoted by v in Fig. 2.13) are used for motion-compensated operations in both the predict and update steps. An important remark is that the predict operator can use all the even-indexed input frames (denoted by x_{2t}) to perform the motion-compensated prediction of the odd-indexed frames (denoted by x_{2t+1}), while the update operator can use all the detail frames (H_t) thus computed in

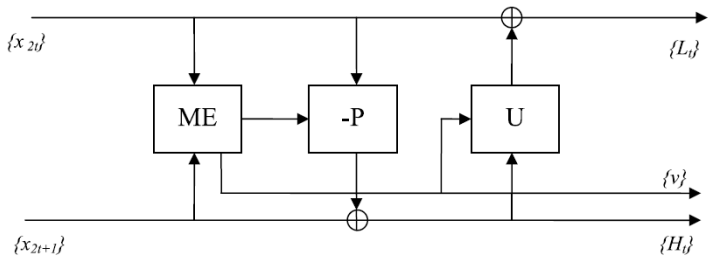


Figure 2.13: General lifting-based MCTF scheme

order to obtain the approximation-subband frames (L_t). Both predict and update operators involve the motion vectors used to match corresponding positions. Therefore, in the $t+2D$ framework, they actually become spatio-temporal operators:

$$\begin{cases} H_t = x_{2t+1} - P(\left\{x_{2(t-k)}, v_{2t+1}^{2(t-k)}\right\}_{k\in T_k^p}) \\ L_t = x_{2t} + U(\left\{H_{t-k}, v_{2t}^{2(t-k)+1}\right\}_{k\in T_k^u}) \end{cases} \tag{2.2}$$

where v_i^j is the motion vector field used to predict the current frame i from the reference frame j , T_k^p (respectively T_k^u) being the support of the temporal predict (respectively update) operator.

Considering the motion-estimation and according to Eq.(2.2), the Haar multiresolution analysis equations (2.1) become:

$$\begin{cases} H_t(n) = x_{2t+1}(n) - x_{2t}(n - v_{2t+1}^{2t}) \\ L_t(p) = x_{2t}(p) + \dfrac{1}{2}H_t(p + v_{2t}^{2t+1}) \end{cases} \tag{2.3}$$

It has been shown in [84] that longer filters are more efficient from the temporal prediction point of view. Among these, the biorthogonal 5/3 filter bank [218] has been the most studied. In this case, both forward and backward motion vectors need to be used for a bidirectional prediction. Thus, the analysis equations of the 5/3 filter-bank have the form:

$$\begin{cases} H_t(n) = x_{2t+1}(n) - \dfrac{1}{2}(x_{2t}(n - v_{2t+1}^{2t}) + x_{2t+2}(n - v_{2t+1}^{2t+2})) \\ L_t(p) = x_{2t}(p) + \dfrac{1}{4}(H_{t-1}(p + v_{2t}^{2t+1}) + H_t(p + v_{2t+2}^{2t+1})) \end{cases} \tag{2.4}$$

We can see that this scheme needs motion estimation between every consecutive pair of motion vectors as opposed to every other pair for 2-tap Haar filters (see Fig. 2.14 (a)). As we have mentioned previously, no boundary problems appear when decomposing a GOP of size 2^L into a number of up to L resolution levels with the Haar filter bank. In the case of 5/3 temporal filtering, we could process one GOP at a time with symmetric extension of the motion trajectories at the boundaries (see Fig. 2.14 (b)). Even though this is essential for perfect reconstruction, it produces a PSNR drop at the GOP boundaries especially at the starting frames where the temporal high-subband is at the boundary.

In order to avoid such artefacts, Zhan *et al.* proposes in [218] (and one year later, Golwelkar in [84]) the *sliding window* implementation for the bidirectional temporal filters. The sliding window approach uses actual data at both GOP ends, instead of a symmetric

extension (see Fig. 2.14 (c)). This means that one has to look ahead, causing a certain amount of delay at the receiver. If we are using a $2M(n)+1$ tap filter at stage n, we will need the future $M(n)$ frames at each temporal level. Hence the longer the filter, the longer the delay. For k levels of temporal resolution, this delay can be evaluated as:

$$D(k) = \sum_{n=0}^{k-1} 2^n M(n).$$

If the $5/3$ filter bank is used at each stage, the coefficient $M(n)$ equals 2, and we will need 30 frames on both sides for a 4 stage MCTF in order to get perfect reconstruction.

As it has been shown in Fig. 2.14, for the bidirectional $5/3$ decomposition the number of motion-vector fields is double that compared to the Haar decomposition, and, therefore, the coding of this information may represent an important part of the bitstream at low bitrates. Efficient algorithms are thus needed to further exploit redundancies between motion-vector fields at the same temporal decomposition level or at different levels [23, 196].

The energy distributed update (EDU) proposed in [214, 74] is an update scheme which tries to avoid a second set of motion vectors, or complex and inaccurate inversion of the motion information, as used in the traditional update step. The basic idea consists of the correlation of the predict/update steps, that is, to perform the update only where predict was made, by distributing high-pass signals to the low-pass frame. Meanwhile, it provides further coding-efficiency gain, as implemented by the Vidwav-MSRA 3D scalable video coding scheme [211].

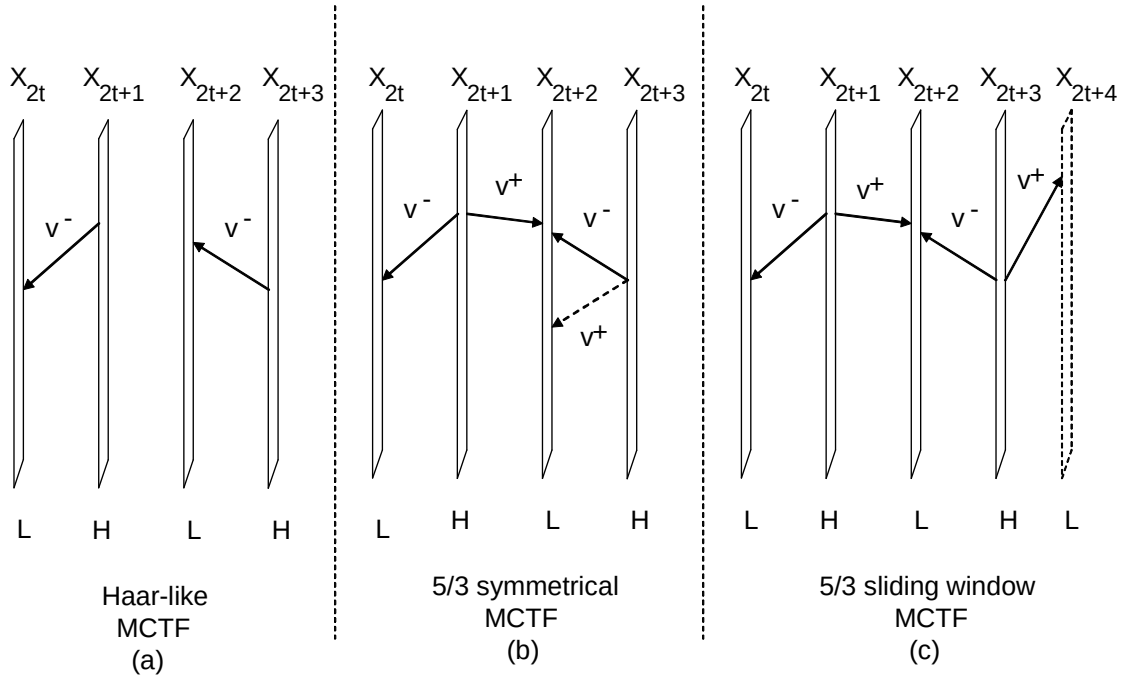


Figure 2.14: One level MCTF using: (a) Haar filtering, (b) $5/3$ symmetrical implementation and (c) $5/3$ sliding window implementation (v^-/v^+ denote the forward/backward motion vectors)

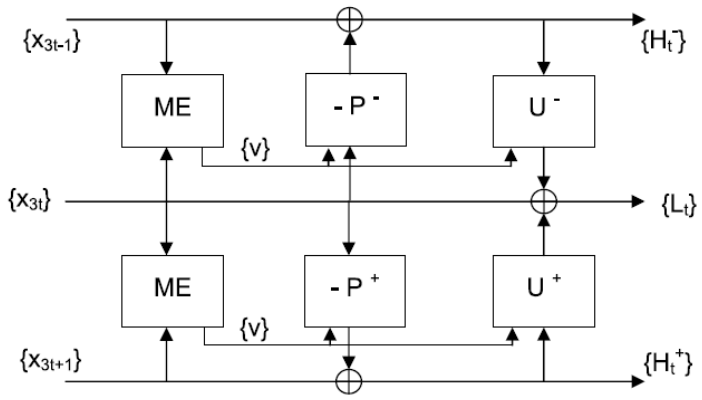


Figure 2.15: Three-band lifting scheme

2.3.1.4 MCTF extentions: Unconstrained-MCTF, 3-Bands

The concept of *unconstrained* MCTF (UMCTF) introduced by van der Schaar *et al.* in [195, 198] allows very useful extensions of MCTF. By selecting the temporal filter coefficients appropriately, multiple reference frames and bidirectional prediction can be introduced in the motion-compensated wavelet framework, such as in H.264/AVC. No update step is used, however, which makes this scheme comparable with an open-loop multiresolution predictive structure. We can adaptively change the number of reference frames, the relative importance attached to each reference frame, the extent of bidirectional filtering, and so on.

UMTCF mechanism provides adaptive temporal filtering through:

- ⋆ variable number of temporal decomposition levels based on the video content or desired complexity level.
- ⋆ adaptive choice of filters enabling different temporal-filtering enhancements.
- ⋆ adaptive choice of filters, within and between temporal- and spatial-decomposition levels.
- ⋆ variable number of successive H frames within and between levels, for flexible (non-dyadic) temporal-scalability and temporal-filtering enhancemements.
- ⋆ different temporal-decomposition structures.

These filters can be adapted across the different frames and between temporal levels, as well as within a frame, on a block or region level. Through appropriate choice of filters and decomposition structures, many different improvements to MCTF become possible. For instance, predictive-coding options such as subpixel accuracies, bidirectional prediction, multiple reference frames etc., may easily be introduced into the MCTF framework. Simultaneously, variable decomposition structures, such as modifying the number of decomposition levels, the number of successive H frames, decomposing H frames etc., can also be introduced. Therefore, with this filter choice, the efficient compensation strategies of conventional predictive coding can be obtained by UMCTF, while preserving the advantages of conventional MCTF.

Other extensions of the temporal transform are aimed at providing non-dyadic scalability factors. This can be achieved by M-band filter banks. In particular, a three-band filter bank in lifting form was proposed by Tillier *et al.* in [180] and is illustrated in Fig. 2.15. For simplicity, Fig. 2.15 shows only the predict and update blocks; however, as in the dyadic case, they involve motion estimation/compensation. Following the figure notations, the analysis equations, which lead to one approximation and two detail subbands, are:

$$\begin{cases} H_t^+(n) = x_{3t+1}(n) - P^+(\{x_{3t}\}_{t\in\mathcal{N}}) \\ H_t^-(n) = x_{3t-1}(n) - P^-(\{x_{3t}\}_{t\in\mathcal{N}}) \\ L_t(p) = x_{3t}(p) + U^+(\{H_t^+\}_{t\in\mathcal{N}}) + U^-(\{H_t^-\}_{t\in\mathcal{N}}) \end{cases} \tag{2.5}$$

Note that in this scheme all the frames indexed by multiples of three are used by the two prediction operators. For example, by choosing frames x_{3t} and x_{3t+3} for the prediction of frame x_{3t+1}, and likewise choosing frames x_{3t-3} and x_{3t} for predicting frame x_{3t-1}, a structure similar to the classical $IBBP\ldots$ can be obtained.

However, the simplest choice, corresponding to a Haar-like transform, is to have identity predict and linear update operators. In this case, the analysis equations become:

$$\begin{cases} H_t^+(n) = x_{3t+1}(n) - x_{3t}(n - v_{3t+1}^{3t}) \\ H_t^-(n) = x_{3t-1}(n) - x_{3t}(n - v_{3t-1}^{3t}) \\ L_t(p) = x_{3t}(p) + \alpha(H_t^+(p + v_{3t+1}^{3t}) + H_t^-(p + v_{3t-1}^{3t})) \end{cases} \tag{2.6}$$

with $\alpha = \frac{1}{4}$ found using the low-pass filter-existence constraint, $L(-1) = 0$. More complex lifting-like schemes have been proposed in [183], as well as other possible M-band motion-compensated temporal structures, like the 5-band temporal lifting scheme that we have introduced in [192] and that we will develop in Section 3.2. For example, these structures allow a framerate adaptation from 30Hz to 10Hz or from 30Hz to 6Hz. Flexible framerate changes can also be achieved by cascading dyadic and M-band filter banks. Another direction for the extension of spatio-temporal transforms is to replace the 2D wavelet decomposition by other representations, such as wavelet packets [141]. Joint wavelet packets [191] describing a unique best-basis representation for several frames, rather than one basis per frame as is classically done, will be introduced in Section 4.2. General filter banks, such as the fully separable wavelet and wavelet-packet transform [188] will be presented in Section 4.3. Also, flexible spatial scalability factors are allowed by the method proposed in [142].

2.3.1.5 Transforms switching: $t + 2D$ and $2D + t$

The interframe wavelet video-coding schemes presented in Sections 2.3.1.3 and 2.3.1.4 employ MCTF before the spatial wavelet decomposition. As mentioned in Section 2.3.1, we refer to this class of interframe wavelet video-coding schemes as $t + 2D$ MCTF. Despite their good coding efficiency performance and low complexity, these types of MCTF structures have also several drawbacks:

- ⋆ *Limited motion-estimation efficiency*. The $t+2D$ MCTF schemes are inherently limited by the quality of the matches provided by the employed motion-estimation algorithm. For instance, discontinuities in the motion boundaries are represented as

high frequencies in the wavelet subbands, and the *intra/inter* mode switch for motion estimation is not very efficient in $t+2D$ MCTF schemes, as the spatial wavelet transform is applied globally and cannot encode the resulting discontinuities efficiently. Moreover, the motion-estimation accuracy, motion model, and adopted motion-estimation block size are fixed for all spatial resolutions, thereby leading to sub-optimum implementations compared with non-scalable coding that can adapt the motion-estimation accuracy based on the encoded resolution. Also, because the motion vectors are not spatially scalable in $t+2D$ MCTF, there is necessary to decode a large set of vectors even at lower resolutions.

⋆ *Limited efficiency of spatial scalability*. If the motion reference during $t+2D$ MCTF is, for example, at HD (high-definition, 1920×1080 pels) resolution and decoding is performed at a low resolution (e.g., QCIF 176×144 pels), this leads to subsampling phase drift for the low-resolution video.

⋆ *Limited spatio-temporal decomposition structures*. In $t+2D$ MCTF, the same temporal-decomposition scheme is applied for all frames. Hence, the same level of temporal scalability is provided independent of the spatial resolution.

A possible solution for the above mentioned drawbacks is to employ *in-band temporal filtering* schemes, where the order of motion estimation and compensation and that of the spatial wavelet transform (2D-DWT) are interchanged (i.e., $2D+t$ MCTF schemes). The spatial wavelet transform for each frame is entirely (or partially in a $2D+t+2D$ coding scheme) performed first and multiple separate motion-compensation loops are used for the various spatial wavelet bands in order to exploit the temporal correlation present in the video sequence. MCTF can now also be applied to spatial wavelet high-pass bands.

The $t+2D$ MCTF schemes (Fig. 2.16(a)) can be easily modified into $2D+t$ or $2D+t+2D$ MCTF (Fig 2.16(b)). More specifically, in $2D+t$ MCTF, the video frames are spatially decomposed into multiple subbands using wavelet filtering, and the temporal correlation within each subband is removed using MCTF (see [149, 148]). The residual texture after the MCTF is coded subband by subband using any desired texture-coding technique (DCT-based, wavelet-based, matching pursuit, etc.). Also, all the recent advances in MCTF can be employed for the benefit of $2D+t$ schemes, which have been developed in [217, 18, 16].

2.3.1.6 Overcomplete MCTF

Overcomplete MCTF is an extention of the in-band $(2D+t)$ coding scheme. Due to the decimation procedure in the spatial wavelet transform (see Section 1.2.3), the wavelet coefficients are not shift invariant with respect to the original signal resolution. Thus the translational motion in the spatial domain cannot be accurately estimated and compensated from the wavelet coefficients, thereby leading to a significant coding efficiency loss. To avoid this inefficiency, motion estimation and compensation should be performed in the overcomplete wavelet domain rather than in the critically sampled domain.

As shown in Section 1.2.3, the overcomplete (redundant) discrete wavelet data (ODWT) can be obtained through a process similar to the critically sampled discrete wavelet signals (DWT) by omitting the subsampling step. Consequently, the ODWT generates more samples than DWT, but enables accurate wavelet-domain motion compensation for the

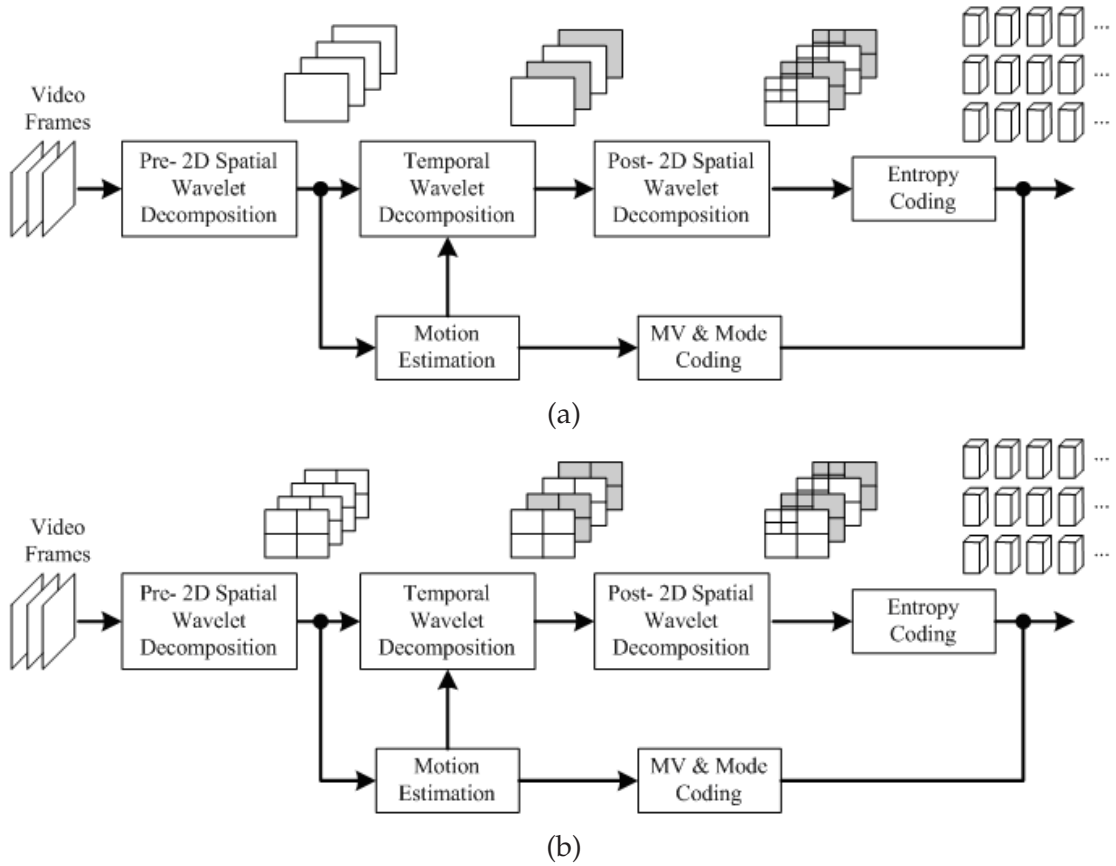


Figure 2.16: Framework for 3D wavelet video coding: (a) $t + 2D$ scheme (pre-spatial decomposition is avoided); (b) case for the $2D + t(+2D)$ scheme (pre-spatial decomposition exists, optional post-spatial decomposition).

high-frequency components, and the signal does not bear frequency-inversion alias components. Despite the fact that ODWT generates more samples, an ODWT-based encoder needs to encode only the critically sampled coefficients. This is because the overcomplete transform coefficients can be generated locally within the decoder. Moreover, when the motion shift is known before the analysis and synthesis stages, it is necessary to compute only those samples of the overcomplete representation that correspond to the actual motion shift. Several in-band motion-compensated video coders involve overcomplete wavelet transforms, like the ones proposed in [18, 16] or [165, 17, 205].

2.3.2 Three-dimensional (3D) wavelet coefficient coding

After 3D ($t + 2D$ or $2D + t$) wavelet analysis, a video sequence will be decomposed into a certain number of 3D subbands. For example, in Fig. 2.17, a three-level motion-compensated wavelet decomposition is performed in the temporal direction, followed by a three-level 2D spatial dyadic decomposition within each of the resulting temporal bands.

The next step in 3D wavelet video coding is to encode the transformed 3D wavelet coefficients in each subband efficiently. Since the subband structure in 3D wavelet decomposition for video sequence is very similar to the subband structure in 2D wavelet

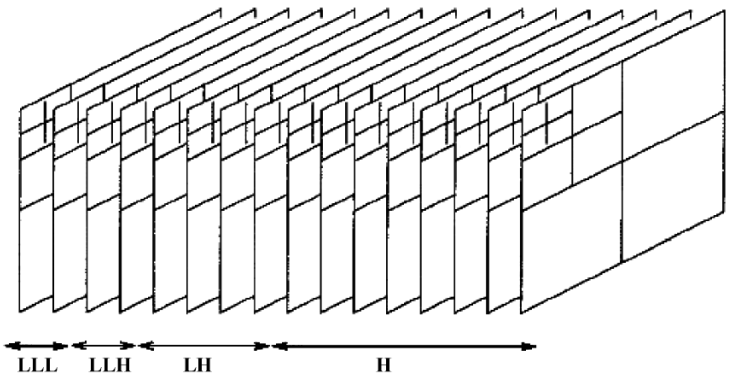


Figure 2.17: Separable 3D wavelet transform: Three-level dyadic temporal (motion-compensated) wavelet decomposition, followed by three-level 2D spatial dyadic decomposition.

decomposition for image, it is natural to extend many existing 2D wavelet-based image coding techniques, such as SPIHT [161], EBCOT [176], and EZBC [93], to the 3D case. As a matter of fact, almost all the existing 3D wavelet coefficients coding schemes use one form of these 3D extensions, such as 3D-SPIHT [104], ESCOT [117] or the 3D extension of EBCOT and 3D-EZBC [44, 210].

Generally speaking, after 3D (motion-compensated) wavelet decomposition, there is not only spatial similarity inside each frame across different scales, but also temporal similarity between two frames at the same temporal scale. Furthermore, the temporal coefficients typically show more correlation along the motion trajectory. An efficient 3D wavelet coefficient coding scheme should exploit these properties as much as possible. Several algorithms for texture coding in 3D wavelet schemes have been developed. We propose to review in the following three of these wavelet-based entropy coders, namely the 3D-SPIHT [104], ESCOT [117] and 3D-EZBC [210].

2.3.2.1 3D-SPIHT

3D-SPIHT is an extension of SPIHT (Set Partitioning in Hierarchical Trees) still-image coding to 3D video coding. The SPIHT algorithm takes advantage of the nature of energy clustering of subband/wavelet coefficients in frequency and space and exploits the similarity between subbands. It utilizes three basic concepts:

- ⋆ searching for sets in spatial-orientation trees in a wavelet transform,
- ⋆ partitioning the wavelet-transform coefficients in these trees into sets defined by the level of the highest significant bit in a bitplane representation of their magnitudes,
- ⋆ coding and transmitting bits associated with the highest bitplanes first.

The 3D-SPIHT scheme can be easily extended from 2D-SPIHT, with the following three similar characteristics:

- ⋆ partial magnitude ordering of the 3D wavelet coefficients with a 3D set-partitioning algorithm,

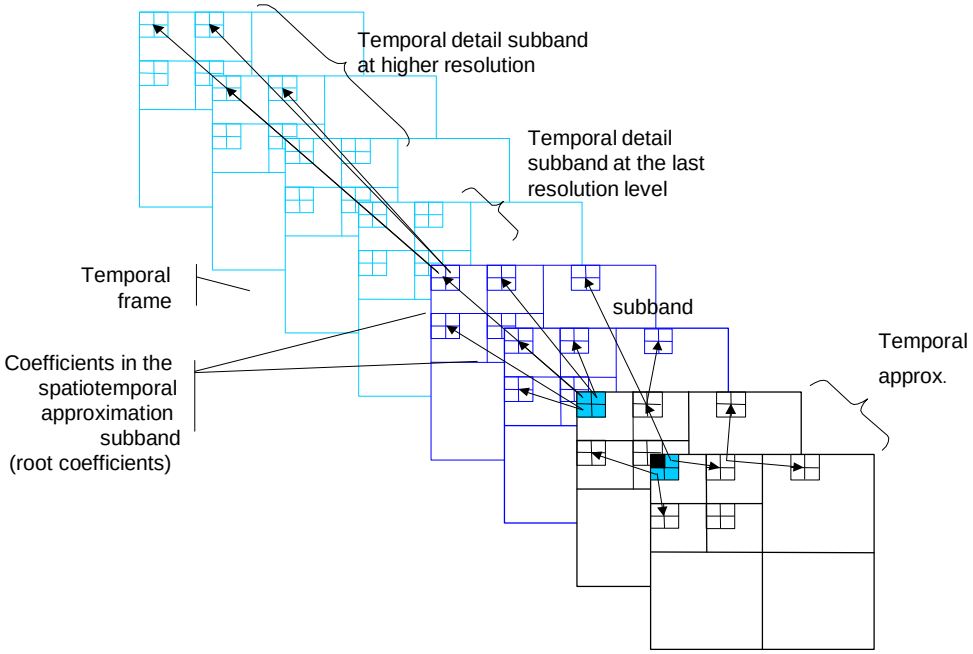


Figure 2.18: Parent-offspring relationship in a spatio-temporal decomposition.

- ordered bitplane transmission of refinement bits,
- exploitation of self-similarity across spatio-temporal orientation trees.

For the 3D wavelet coefficients, a new 3D spatio-temporal orientation tree and its parent-offspring relationships are defined. For pure dyadic wavelet decomposition with an alternate separable wavelet transform in each dimension, a straightforward extension from the 2D case is to form a node in 3D-SPIHT as a block with eight adjacent pixels, two in each dimension, hence forming a node of $2 \times 2 \times 2$ pixels. The root nodes (at the highest level of the pyramid) have one pixel with no descendants and the other seven pointing to eight offspring in a $2 \times 2 \times 2$ cube at corresponding locations at the same level. For non-root and non-leaf nodes, a pixel has eight offspring in a $2 \times 2 \times 2$ cube one level below in the pyramid. For non-dyadic decomposition similar to the 2D wavelet-packet decomposition case, the $2 \times 2 \times 2$ offspring nodes are split into pixels in these smaller subbands at the corresponding orientation in the nodes at the original level. For the common $t+2D$ type of wavelet decomposition the parent-offspring relationship is shown in Fig.2.18.

With such defined 3D spatio-temporal trees, the coefficients can be compressed into a bitstream by feeding the 3D data structure to the 3D SPIHT coding kernel. The 3D-SPIHT kernel will sort the data according to the magnitude along the spatio-temporal orientation trees (sorting pass) and refine the bitplane by adding necessary bits (refinement pass).

2.3.2.2 MC-EZBC

MC-EZBC [44] (Motion-Compensated Embedded Zero Block Coding) is an extension of the EZBC image coder [93], allowing the encoding of the 3D wavelet coefficients. The concept of EZBC is inspired by the success of two popular embedded image-coding techniques: zero tree-block coding, such as SPIHT [161], and context modeling of the sub-

band/wavelet coefficients, such as EBCOT [176]. As shown in Section 2.3.2.1, the zero tree-block coding takes advantage of the natural energy clustering of subband/wavelet coefficients in frequency and in space and exploits the similarity between subbands. Moreover, instead of all pixels, only a small number of elements in the lists [161] need to be processed in individual bitplane coding passes. Thus, processing speed for this class of coders is very fast. However, in the context-model based coders [176], individual samples of the wavelet coefficients are coded bitplane-by-bitplane using context-based arithmetic coding to effectively exploit the strong correlation of subband/wavelet coefficients within and across subbands. Nevertheless, unlike zero tree-block coders, these algorithms need to scan all subband/wavelet coefficients at least once to finish coding of a full bitplane, with an implied higher computation cost.

The EZBC algorithm combines the advantages of these two coding techniques, i.e., low computational complexity and effective exploitation of the correlation of subband coefficients, using both zero-blocks of subband/wavelet coefficients and context modeling. Similar to EZBC for image coding, 3D-EZBC is based on quadtree representations of the individual subbands and frames. The bottom quadtree level, or pixel level, consists of the magnitude of each subband coefficient. Each quadtree node of the next-higher level is then set to the maximum value of its four corresponding nodes at the current level. In the end, the top quadtree node corresponds to the maximum magnitude of all the coefficients from the same subband. As in EZBC, 3D-EZBC uses this quadtree-based zero-block coding approach for hierarchical set-partitioning of the subband coefficients to exploit the strong statistical dependency in the quadtree representation of the decomposed subbands. Furthermore, to code the significance of the quadtree nodes, context-based arithmetic coding is used. The context includes eight first-order neighboring nodes of the same quadtree level and the node of the parent subband at the next-lower quadtree level. Experiments have shown that including a node in the parent subband in the inter-band context model is very helpful in predicting the current node, especially at higher levels of a quadtree. Like SPIHT and other hierarchical bitplane coders, lists are used for tracking the set-partitioning information. However, the lists in 3D-EZBC are separately maintained for nodes from different subband and quadtree levels. Therefore, separate context models are allowed to be built-up for the nodes from different subbands and quadtree levels. In this way, statistical characteristics of quadtree nodes from different orientations, subsampling factors, and amplitude distributions are not mixed up. This ensures a resolution scalable bitstream while maintaining the desirable low complexity feature of this class of coders.

2.3.2.3 3D-EBCOT (ESCOT)

As mentioned in Section 2.3.2.1, 3D-SPIHT [104] provides natural SNR scalability due to the efficient bitplane representation. However, it is difficult to provide temporal or spatial scalabilities due to the inherent spatio-temporal tree structure. Even with extra effort, it can provide only partial temporal or spatial scalabilities by modifying the decoder or encoder [93]. However, the 3D extension of EBCOT [176], ESCOT [117], can provide full rate (SNR), temporal and spatial scalabilities by constraining the encoding of wavelet coefficients independently within each subband. Meanwhile, the R-D optimized bitstream-truncation process after encoding guarantees a bitstream with the best video quality given a bitrate constraint.

The ESCOT scheme is in principle very similar to EBCOT in the JPEG-2000 stan-

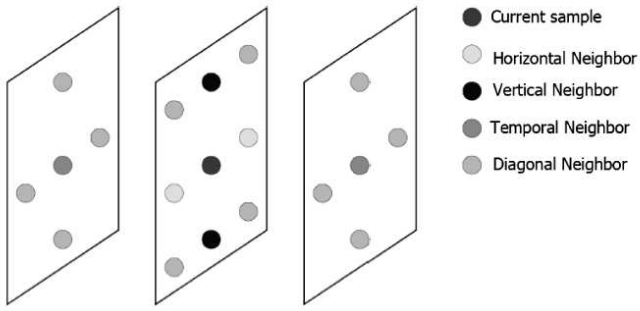


Figure 2.19: Immediate neighbors of a sample in 3D-ESCOT coding.

dard [8], which offers high compression efficiency and other functionalities (e.g., error resilience and random access) for image coding. By extending the 2D-EBCOT algorithm to 3D-ESCOT, a different coding structure is used to form a new set of 3D contexts for arithmetic coding, which makes the algorithm very suitable for scalable video compression. Specifically, each subband is coded independently in the extended coding structure. The advantage of doing so is that each subband can be decoded independently to achieve flexible spatial/temporal scalability. The user can mix an arbitrary number of spatio-temporal subbands in any order to obtain the desired spatial or temporal resolution. Unlike the EBCOT encoder in JPEG2000, the ESCOT encoder takes a subband as a whole entity. There are two reasons for this:

- ⋆ normally a video frame has lower resolution than a still image; not splitting a subband further into many small 3D blocks brings better coding efficiency of the context-based adaptive arithmetic coder.
- ⋆ taking a subband as a whole entity is also convenient for incorporating the possible motion model in the coding process, since within the same 3D subband, the motion vector may point from any coefficient on a temporal plane to any other coefficient on other temporal planes.

As in the 2D-EBCOT case, the contexts for 3D-ESCOT are also formed from immediate neighbors in the same subband. The difference is that the immediate neighbors are now in three directions instead of two: horizontal, vertical, and temporal (see Fig. 2.19). In addition, the temporal neighbors may be not only spatially collocated in different frames, but also neighbors pointed to by motion vectors across frames with a certain motion model [117].

The encoding of the 3D wavelet coefficients in the 3D-ESCOT scheme is done bitplane by bitplane. For each bitplane, the coding procedure consists of three distinct passes: Significance Propagation, Magnitude Refinement, and Normalization, which are applied in turn. Each pass processes a *fractional bitplane*. In each pass, the scanning order is along the horizontal direction first, the vertical direction second, and finally the temporal direction. In the *Significance Propagation* pass, the samples that are not yet significant but have a *preferred neighborhood* are processed. A sample has a *preferred neighborhood* if and only if the sample has at least a significant immediate diagonal neighbor for a HHH (high frequency in three directions) subband or a significant immediate horizontal, vertical or temporal neighbor for the other types of subbands. In the *Magnitude Refinement* pass, the samples that have been significant in the previous bitplanes are encoded. In the

Normalization pass, those samples that have not yet been coded in the previous two passes are coded.

In the previous stage, each subband is coded separately up to a specific precision and each forms an independent bitstream. The objective of optimal bitstream truncation is to construct a final bitstream that satisfies the bitrate constraint and minimizes the overall distortion. As in the EBCOT algorithm [176], the end of each pass at each *fractional bitplane* is a candidate truncation point with a pre-calculated R-D value pair for that subband. A straightforward way to achieve R-D optimized truncation is to find the convex hull of the R-D pairs at the end of each fractional bit plane and truncate only at the candidate truncation points that are on the convex hull. To achieve quality scalability, a multilayer bitstream may be formed, where each layer represents a quality level. Depending on the available bandwidth and the computational capability, the decoder can choose to decode up to the layer it can handle. The fractional bitplane coding ensures that the bitstream is finely embedded. Since each subband is independently coded, the bitstream of each subband is separable. The encoder can choose to construct a bitstream favoring spatial scalability or temporal scalability. Also, the decoder can easily extract only a few subbands and decode only these subbands. Therefore, the implementation of resolution scalability and temporal scalability is natural and easy.

2.4 Conclusion

In this chapter we have briefly introduced scalable video coding concepts. As it has been shown in Section 2.3, MCTF-based coders provide high flexibility in bitstream scalability across different temporal, spatial, and quality resolutions. In addition, they provide better error resilience than conventional (prediction-based) coders. In fact, MCTF-based coders are better able to separate relevant from irrelevant information. The temporal low-pass bands highlight information that is consistent over a large number of frames, establishing a powerful means for exploiting multiple frame redundancies not achievable by conventional frame-to-frame or multiframe prediction methods. Moreover, noise and quickly changing information that cannot be handled by motion compensation appear in the temporal, high-pass bands, which can supplement the low-pass bands for more accurate signal reproduction whenever desirable, provided that a sufficient data rate is available. Hence, the denoising process that is often applied as a preprocessing step before conventional video compression is an integral part of scalable MCTF-based coders. Due to a non-recursive structure, higher degrees of freedom are possible for both encoder and decoder optimization. In principle, a decoder could integrate additional signal synthesis elements whenever the received information is incomplete, such as framerate up-conversion, film-grain noise overlay or other elements of texture and motion synthesis, which could be integrated easily as part of the MCTF-synthesis process without losing any synchronization between encoder and decoder. From this point of view, even though many elements of MCTF in the lifting interpretation can be regarded as extensions of proven techniques from motion-compensated prediction-based coders, this framework exhibits and enables a number of radically new options in video encoding. However, when a wavelet transform is applied for encoding of the low-pass and high-pass frames resulting from the MCTF process, the commonalities with 2D wavelet coding methods are obvious.

If the sequence of spatial and temporal filtering is exchanged ($2D+t$ instead of $t+2D$

wavelet transform), MCTF can be interpreted as a framework for further interframe compression of (intra-frame restricted) 2D wavelet representations such as JPEG2000. From this point of view, a link between the previously separate worlds of 2D wavelet coding with their excellent scalability properties and compression-efficient motion-compensated video-coding schemes is established by MCTF. This shows the high potential for future developments in the area of motion-picture compression, even allowing seamless transition between intra-frame and inter-frame coding methods, depending on the application requirements for flexible random access, scalability, high compression, and error resilience. Furthermore, scalable protection of content, allowing access management for different resolution qualities of video signals, is a natural companion of scalable compression methods. This is why we have chosen the scalable MCTF-based wavelet coding paradigm as conceptual development framework for this thesis work.

All the contributions of the thesis work have the common aim of improving several aspects of $t + 2D$ wavelet-based video coding. Following the natural processing order of the chosen framework, we will introduce in Chapter 3 several MCTF schemes in a $t + 2D$ lifting-based coding approach which will improve certain aspects of temporal decorrelation following the motion-field direction.

Part II

Design of a scalable video codec

Chapter 3

Temporal processing of video sequences

We have seen in section 2.3 that three-dimensional (3D) subband/wavelet coding via a motion-compensated temporal filter (MCTF) is a very effective structure for highly scalable video coding. In this chapter we will present some of our contributions to the improvement of the MCTF coding approach.

As discussed in section 2.3.1, the most-used decompositions for temporal decorrelation are dyadic, i.e. Haar and 5/3 two-tap filter banks. Generally, uniform filtering is made on the assumption that the frames are highly correlated. However, this assumption no longer holds when the video shot encounters scene-cuts, as in the case of action movies, music video clips etc. As mentioned in section 2.3.1.3, the inefficiency of the motion estimator in this case leads to poor predict/update stages, which, combined with the sliding window implementation (see section 2.3.1.3) of the temporal filters, leads to prediction/update error propagation through the decomposition levels. In section 3.1, we propose an adaptive motion-compensated temporal coding scheme able to overcome this deficit by detecting and adequately processing the scene-cuts that may occur in a video sequence.

In section 2.3.1.3, and also in 2.3.1.4, we mentioned that the longer filters are preferred for temporal decompositions for their efficiency in removing the temporal redundancy. However, when the filters are too long, it is very likely that they will encompass several (different) scenes and, thus, lose their decorrelation efficiency. A question can be risen here: in which situations could we use long temporal filters without having the motion-complexity problem? A simple answer would be: for video surveillance, where there is at least a 50%-50% chance for smooth motion transitions (during night period, for example). Moreover, the longer the temporal filter, the fewer temporal decomposition levels needed in order to obtain some key (approximation) frames useful for video database search and storage. This is one of the reasons for proposing the 5-band temporal lifting scheme, presented in section 3.2.

Since the most-used method for motion estimation is block-based (see section 2.3.1.1), even with a bidirectional temporal prediction, block artefacts are still present. In order to avoid such artefacts, motion-compensation solutions such as weighted-average update operator [83, 182] or overlapped block motion-compensation [211] have been proposed in order to alleviate this problem. In section 3.3 we propose to improve the prediction of the high-frequency temporal subband frames by using an adaptive filter bank. The proposed

LMS based adaptive prediction can be applied to any temporal-prediction scheme.

MCTF-based coding efficiency is strongly related to the correlation of the data being processed. Based on this assumption, we have thought that the $t + 2D$ video-coding principles can be applied also to multispectral data sequences, which are 3D sequences, the third dimention being given by the frequency spectrum, not the time as in the case of common video sequences. In order to achieve data compression, coding techniques applied to multispectral data take advantage of the presence of two redundancy sources: spatial correlation among neighboring pixels in the same spectral band and spectral correlation among different bands at the same spatial location. In section 3.4, we propose to evaluate the performance on SPOT (1,2 and 4) sequences of still-image (JPEG2000) and video ($t + 2D$) compression techniques based on wavelet tools.

3.1 Scene-cut processing in motion-compensated temporal filtering

It has been shown in section 2.3.1 that the $t + 2D$ subband schemes exploit the temporal interframe redundancy by applying an open-loop temporal wavelet transform over the frames of a video sequence. A weakness of the existing $t + 2D$ video codecs is related to the way the temporal filtering behaves near scene changes. Usually, the input video signal is partitioned into GOPs and temporally filtered without checking the correlation between the GOP frames. Moreover, the sliding window implementation [218] of the temporal filtering is done using frames from adjacent GOPs in the processing of the current GOP. When the input signal involves complex motion transitions and especially scene-cuts, this can translate into inefficient prediction/update operations, leading to poor-quality results and also to reduced temporal-scalability capabilities.

Several attempts to avoid the artefacts related to these abrupt changes have already been proposed for *hybrid* coding, such as the scene-cut detection and content-based sampling of video sequences [168] or video segmentation using encoding-cost data [61], alleviating, but not completely solving, this problem.

This section presents a MCTF coding scheme specially adapted to the detection and processing of uncorrelated shots of the input video sequence. The method was presented in the proceedings of the ACIVS'05 conference [189].

Once the scene-cuts are detected, we propose to encode each set of frames between two consecutive scenes separately, by adapting the temporal filtering to cope with an arbitrary number of frames in a shot. An advantage of the proposed scheme is that once the scene-cuts are eliminated, MCTF efficiency is maximal, as for highly-correlated video signals. The problem is related to border effects and is therefore much easier to deal with in the case of Haar MCTF. However, as mentioned in section 2.3.1.3 and in [218, 136, 195], the use of longer bidirectional filters, like the 5/3 filter bank, can take better advantage of the temporal redundancy between frames. Existing methods for adaptive GOP structure in the MCTF framework [209, 38] basically detect changes and limit the number of temporal decomposition levels based on a measure of unconnected pixel percentage. However, compared to our approach, this technique does not make a strict correspondence between the scene-cut and the GOP boundary. Our proposed approach varies the GOP size only on the frames previous to the transition, and these frames are encoded in several GOPs of sizes of a power of two. In this way, the scene cut does not span any GOP. We present in the following our scene-cut processing method in the framework of

5/3 MCTF, but the proposal can be adapted to other temporal filters.

3.1.1 Scene-cut detection

The MCTF approach consists of a hierarchical open-loop subband motion-compensated decomposition (see section 2.3.1.3). Let us denote by x_t the original frames, t being the time index, and by h_t and l_t the high-frequency (detail) and low-frequency (approximation) subband frames, respectively. Let us recall the lifting-form implementation of the 5/3 filterbank, where the operators allowing computation of the decomposition subbands are bidirectional, and the equations have the form (see also Fig. 3.1):

$$\begin{cases} h_t = x_{2t+1} - \frac{1}{2}\big(\mathcal{F}(x_{2t}, \boldsymbol{v}_t^+) + \mathcal{F}(x_{2t+2}, \boldsymbol{v}_t^-)\big) \\ l_t = x_{2t} + \frac{1}{4}\big(\mathcal{F}^{-1}(h_{t-1}, \boldsymbol{v}_{t-1}^-) + \mathcal{F}^{-1}(h_t, \boldsymbol{v}_t^+)\big) \end{cases} \tag{3.1}$$

where $\mathcal{F}(x_t, \boldsymbol{v}_t)$ is the motion-prediction operator, compensating the frame x_t by projection in the direction of the motion-vector field $\boldsymbol{v}_t$, and $\boldsymbol{v}_t^+$, $\boldsymbol{v}_t^-$ are respectively the forward and backward motion vectors predicting x_{2t+1}. The notation $\mathcal{F}^{-1}(h_t, \boldsymbol{v}_t)$ corresponds to the compensation of the h_t frame in the opposite direction of the motion vector field $\boldsymbol{v}_t$. Indeed, in general, the motion prediction is not an invertible operator. Unconnected and multiple connected pixels are processed as detailed in [184].

When the input sequence involves complex motion transitions, this can translate to inefficient prediction/update operations, leading to poor-quality results and temporal-scalability capabilities, as illustrated in Fig. 3.2. One can remark, in particular, the energy of the detail frames to be encoded and also the poor visual quality of the approximation frame, very detrimental to temporal scalability.

Several criteria for scene-cut detection have been proposed in the literature, such as: the variation of the relative energy of the displaced frame difference (DFD) along the sequence [194], the energy and angle distribution of the motion-vector fields in consecutive frames [152], by keeping track of the percentage of the unconnected pixels given by

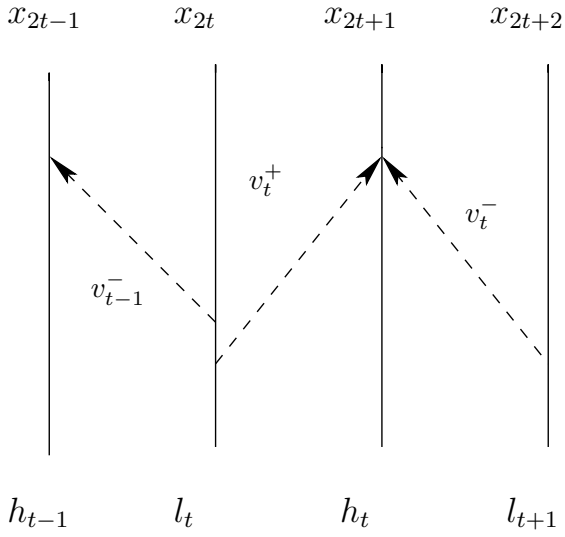


Figure 3.1: MCTF with bidirectional predict and update lifting steps.

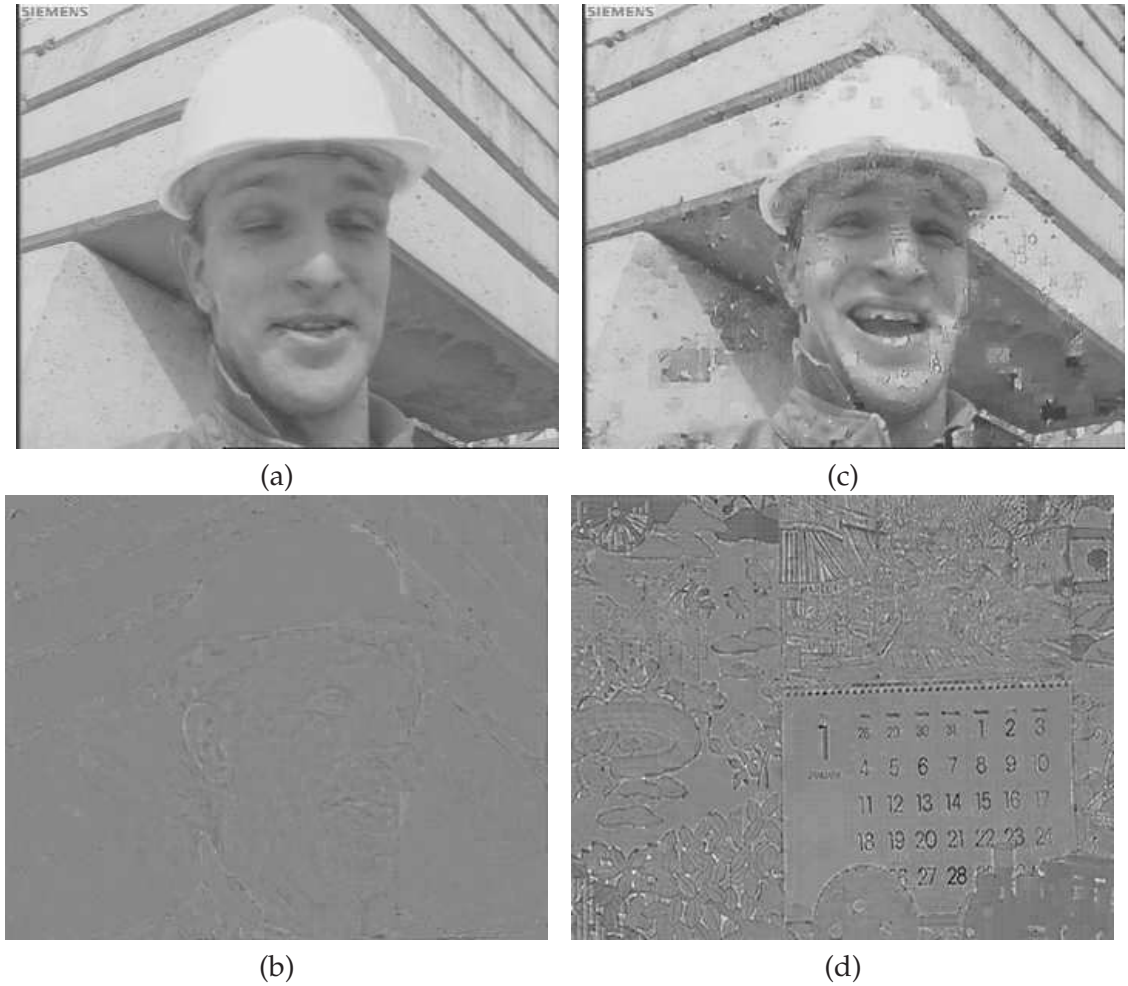


Figure 3.2: Approximation (a) and detail (b) frames in a GOP without scene-cut. Approximation (c) and detail (d) frames when the GOP contains a scene-cut (first part: Foreman (CIF, 30Hz) sequence, second part: Mobile (CIF, 30Hz) sequence).

motion estimation [108], or using unsupervised segmentation and object tracking [40].

3.1.1.1 Variation of relative energy

We have chosen as detection criterion the variation of the relative energy of the DFD along the sequence, mainly because of its reduced complexity, coupled with a good detection capability. If the DFD between two successive frames is computed as:

$$d_t = DFD(x_t, x_{t+1}) = x_{t+1} - \mathcal{F}(x_t, \boldsymbol{v}_t) \tag{3.2}$$

then the variation of the relative energy of the DFD is computed as:

$$\Delta_{2t} = \frac{d_{2t}^2}{d_{2t-1}^2} \tag{3.3}$$

When the input signal is highly correlated, the variation of the relative energy of the DFD along the sequence is almost constant (i.e., $\Delta \approx 1$). We say a scene-cut is detected when the variation of relative energy has a rapid change. For appropriately chosen parameters τ_1 and τ_2, we say that the scene-cut occurs after the frame x_{2t+1} when:

$$\begin{cases} |\Delta_{2t} - 1| < \tau_1 \\ |\Delta_{2t+1} - 1| > \tau_2 \end{cases} \tag{3.4}$$

Note that any other scene-change detection algorithm present in the literature could replace the DFD criterion. Once the scene-cut decision has been made, we pass to the next step, namely the processing of the frames preceding the cut.

3.1.2 Scene-cut processing

Before introducing our method for processing the frames between two scene changes, we propose to review Chen's alternative solution [209, 38] for MCTF-based coders using adaptive GOP structures.

3.1.2.1 Adaptive GOP structure

Existing methods for adaptive GOP structure in the MCTF framework [209, 38] basically detect changes and limit the number of temporal decomposition levels based on a measure of unconnected pixel percentage (see section 2.3.1.2). MCTF is performed in each temporal level. Such filtering makes sense only when the motion information is reliable at that level. The variable-size block-matching motion estimation used in MC-EZBC [93] is based on the assumption of a rigid-motion and affine-transformation motion model. When the video sequence presents editing effects such as fade-in, fade-out, and dissolve, the motion-estimation and compensation algorithms will fail, by introducing multiple local motions, even though there is still temporal correlation. In these cases, smaller GOP sizes or even intraframe coding are preferred.

Based on the percentage of unconnected pixels at one temporal level, the decision is taken whether to proceed with motion-compensated filtering at the next temporal level, i.e., next-lower frame rate. In this way, an adaptive GOP size structure is achieved with a varying number of temporal decomposition levels.

3.1.2.2 Proposed method

First, temporal filtering needs to be changed in order not to filter over a scene-cut. The second modification is related to the encoding of the last group of frames (GOP) before the scene-cut. To this end, both the predict and update steps have to be modified near the end of the first scene, as illustrated in Fig. 3.3.

For homogeneously processed sequences, the temporal subbands resulting from the MCTF are encoded by GOPs of 2^L frames, where L is the number of temporal decomposition levels that are performed, as mentioned in section 2.3.1.3. When a scene-cut occurs in a sequence, the GOP just before the change will have, in general, a different number of frames. If we denote its number of frames by A_n and write this number in a binary representation as:

$$A_n = (a_0 a_1 \dots a_{L-1})_2 = \sum_{l=0}^{L-1} a_l 2^l,$$

then we shall decompose the GOP into smaller GOPs, in decreasing order of their size: $a_l\, 2^l,\ l \in \{0, \dots, L-1\},\ a_l \in \{0, 1\}$, which will be filtered and encoded separately. This also corresponds to changing the number of temporal-decomposition levels and filtering operations for these sub-GOPs.

Indeed, we can do only l temporal decomposition levels for a sub-GOP of size 2^l, $l < L$. Moreover, the prediction across the scene-cut is not allowed, as well as the usage of the reverse motion-vector field over the same transition, during the update step. After the scene-cut, the normal filtering with sliding window (or on-the-fly) is started, the effect of the scene cut being only a slight modification of the filters to take into account the induced border effects.

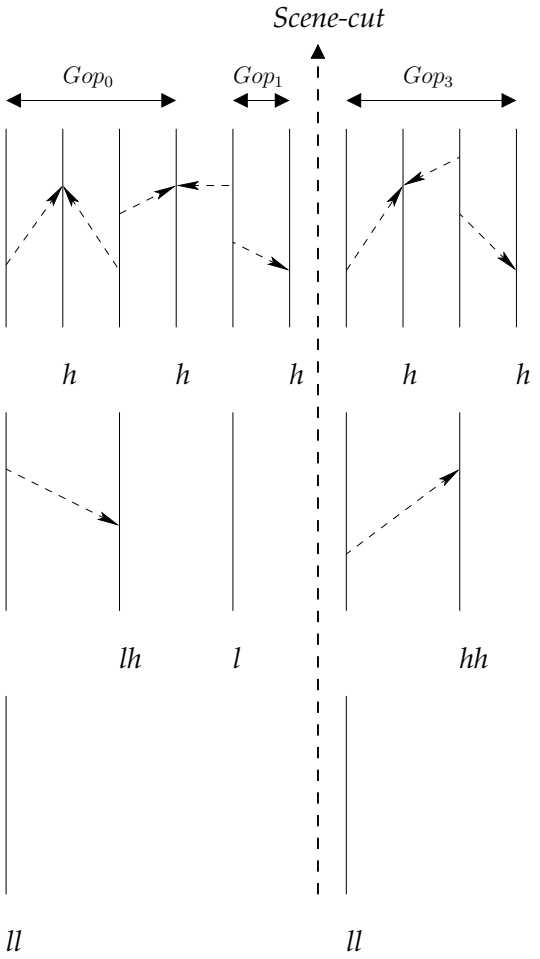


Figure 3.3: Scene-cut processing over two temporal levels of a 10-frame video shot.

Chen's adaptive GOP structure presented above and described in [38] in comparison to our approach does not make a strict correspondence between the scene-cut and the GOP boundary. Our proposed method varies the GOP size only on the frames previous to the transition, and these frames are encoded in several GOPs of power-of-two size. In this way, the scene cut does not span any GOP.

3.1.3 Experimental results

For the simulations, we have considered a high-definition video sequence (HD format: 1920×1280, 60 Hz) from the Erin Brockovich movie, containing 180 frames and 3 scene-cuts: after the 44^{th}, the 80^{th} and respectively, the 161^{th} frame. Moreover, in order to work on a representative set of test sequences, we have also built-up several test sequences obtained by concatenating parts of some standard CIF sequences at 30 Hz: Foreman and Mobile (i.e.: MF_18×16 - video file containing the first 18 frames from Mobile and the next 16 frames from Foreman, FM_16×16 - with the first 16 frames from Foreman, followed by the first 16 frames from Mobile).

The aim was to test all possible configurations for the number of frames in the GOP previous to the scene-cut. In order to detect the abrupt scene transitions, the values of τ_1 and τ_2 in Eq.(3.4) were empirically determined as being equal to 0.1 and 0.4, respectively. These parameters ensured that all the scene-cuts were detected and no false alarms appeared for the considered sequences. Sequences with fade or dissolve transitions can be processed with the described MCTF scheme, but the detection method should be replaced with an appropriate one, as described in [193].

The target number of decomposition levels for motion-compensated 5/3 temporal filtering is $L = 4$. The coding procedure is based on the MC-EZBC codec [210] and the motion-estimation algorithm used is Hierarchical Variable Size Block Matching (HVSBM) [44]. The motion vectors have been estimated with $1/8^{th}$ pixel accuracy and the temporal subbands were spatially decomposed over 4 levels with the biorthogonal 9/7 wavelets. The encoding of the entire YUV sequence was performed, but the results are further expressed only in terms of average YSNR.

YSNR (dB)	6000 kbs	8000 kbs	12000 kbs
SC-MCTF	36.4227	36.8639	37.6387
MCTF	34.9281	35.7519	36.5217

Table 3.1: PSNR results of 5/3 MCTF with and without scene-cut processing for Erin Bronckovich - (HD, 60Hz,180 frames).

MF_18x16 sequence (30Hz)				
YSNR (dB)	512 kbs	768 kbs	1024 kbs	1536 kbs
SC-MCTF	30.1185	32.3141	33.7612	35.6489
MCTF	23.9811	28.5192	30.4135	32.8334
FM_16x16 sequence (30Hz)				
YSNR (dB)	512 kbs	768 kbs	1024 kbs	1536 kbs
SC-MCTF	30.3151	32.7043	34.1021	35.9510
MCTF	26.4706	30.3061	31.8275	33.8650

Table 3.2: PSNR results of 5/3 MCTF with and without scene-cut processing for MF_18x16 and FM_16x16 sequences.

The importance of correctly processing the scene-cuts is illustrated in Fig. 3.4, Fig. 3.6, as well as in Tab. 3.1 and Tab. 3.2, where the rate-distortion performances for 5/3 MCTF with (denoted in these tables by SC-MCTF) and without (simply denoted by MCTF) scene-cut processing are compared. Moreover, the quality of the reconstructed frames prior and after the cut is enhanced, as it can be observed in Fig. 3.5.

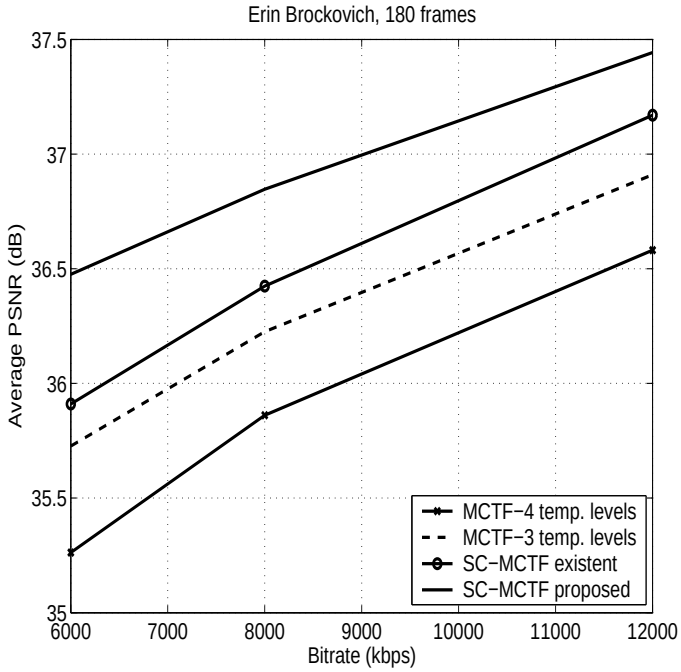


Figure 3.4: Rate-distortion curves for 180-frames Erin Brockovich (HD 1920×1280, 60Hz) sequence (*SC* stands for scene-cut processing and *SC-MCTF existent* refers to the method proposed in [209, 38]).

(a) (b)

Figure 3.5: Reconstructed frame from Erin Brockovich (HD 1920×1280, 60Hz) sequence: (a) homogeneous processing, (b) scene-cut processing using the proposed method.

It can be noticed easily that, in all the cases, our scheme performs better, achieving a gain between 0.5 and 2.0 dB over classical MCTF. Results in Fig. 3.4 indicate that reducing the GOP size (from 16 to 8 frames) can alleviate the problem of scene-cuts by decreasing their influence, but a correct processing of these zones allows us both to take advantage of the temporal correlation in homogeneous shots and to increase the coding efficiency. It can also be observed that our proposed technique outperforms the one described in [209, 38].

3.1.4 Conclusion

We have presented in this section an improved version of the 5/3 MCTF coding scheme, able to detect and process scene-cuts appearing in video sequences. The lifting structure of the filter bank has been modified such that the filtering does not encompass the scene-cut. Moreover, the coding units were reduced to accommodate this change. As can be observed from the experimental results, our method gives an average YSNR gain of about 1.5 dB on the tested video sequences and higher for frames close to the scene-cut.

Following the leading ideas regarding the coding efficiency of the longer temporal filters discussed in section 2.3.1.4 and proposed in [218], we present in the next section a new motion-compensated temporal lifting scheme, specially adapted for the encoding of low-motion sequences and having a direct application in video surveillance.

3.2 5-band motion compensated temporal lifting scheme

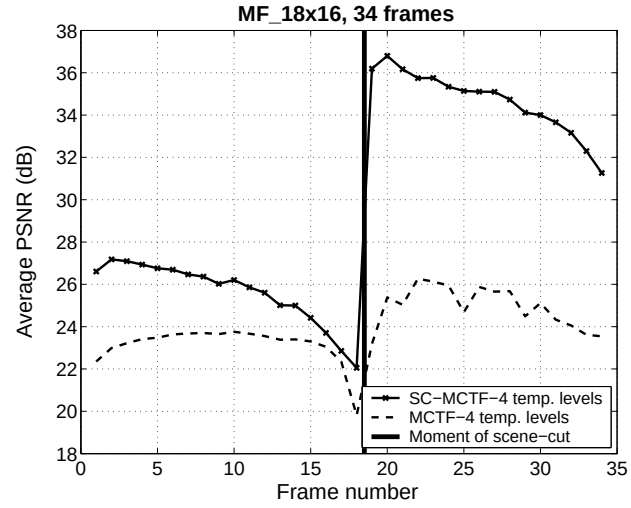
As mentioned in section 2.3, the wavelet-based video-coding schemes are well known for high coding efficiency and spatio-temporal scalability, high energy compaction of subband transforms for efficient video compression and resilience to transmission errors. These are key factors for multimedia applications over heterogeneous networks.

The lifting [173, 99] implementation of these subband-coding schemes insures low complexity which makes it the most widely adopted approach to 3D subband wavelet coding in the literature [44, 134, 94]. As previously mentioned, the dyadic MCTF schemes were subject to numerous optimizations and improvements, concerning, for example, the lifting predict/update operators [196, 145, 143, 144, 184] or precision of motion estimation [82, 94]. Moreover, as shown in section 2.3.1.4, M-band lifting schemes with perfect reconstruction [41, 89] or 3-band temporal decomposition [181, 180] were proposed, allowing non-dyadic scalability factors. Using a method similar to that introduced for the 3-band motion-compensated temporal structures, we can extend this decomposition to several channels.

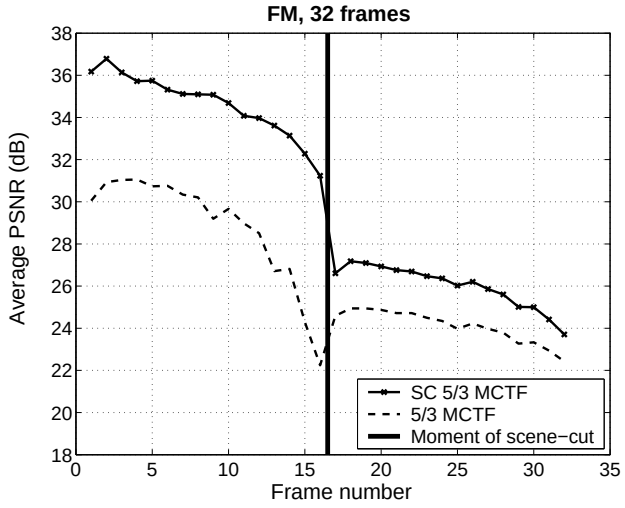
The interest for more channels can be two-fold. First, to allow complete freedom in the choice of the scalability factor (e.g. allowing temporal subsampling with factors of 5, 7, and combinations of such factors: for example, a 3-band scheme followed by a 5-band scheme leads to a reduction of a factor 15 in the framerate). Second, this enables the creation of approximation subbands using a reduced number of temporal decompositions.

We will introduce in this section a new lifting-based method of temporal decomposition which provides a scalability factor of 5 in a motion-compensated subband video-coding scheme. This work has been presented in the proceedings of the MMSP'06 [192] and NSIP'07 [190] conferences.

Depending on the sequence characteristics, motion model etc., this structure can provide higher coding performance. Also, depending on the desired framerates/temporal



(a)



(b)

Figure 3.6: PSNR for the MF_18x16 (a) and FM_16x16 (b) sequences, with and without scene-cut processing. Scene-change after the 18^{th} and respectively, 16^{th} frame.

scalabilities a particular structure could be more efficient. It provides better coding efficiency of the temporal subband approximation frames, so an improved temporal scalability is obtained. This feature benefits certain applications, for example video-surveillance sequences, where the motion activity is very low in most cases. Moreover, scalable video-coding schemes have been successfully used in multiple video-surveillance systems [221], such as metropolitan transportation networks, airports or other public places, such systems being in considerable growth over the past years.

3.2.1 5-band MCTF structure

Let us denote by x_t the original frames, t being the time index, and by h_t and l_t the high-frequency (detail) and low-frequency (approximation) subband frames, respectively.

As an example, a typical 5-band lifting (and therefore, invertible) scheme could result in one approximation subband and 4 detail subbands (see Fig. 3.7).

The corresponding equations describing the analysis part are:

$$\begin{cases} h_{1,t}^{-} = x_{5t-1} - P_1^{-}\left\{x_{5t-2}, x_{5t}\right\}_t \\ h_{1,t}^{+} = x_{5t+1} - P_1^{+}\left\{(x_{5t+2}, x_{5t}\right\}_t \\ h_{2,t}^{-} = x_{5t-2} - P_2^{-}\left\{x_{5t}\right\}_t \\ h_{2,t}^{+} = x_{5t+2} - P_2^{+}\left\{x_{5t}\right\}_t \\ \quad l_t = x_{5t} + U^{-}\left\{h_{1,t}^{-}, h_{2,t}^{-}\right\}_t + U^{+}\left\{h_{1,t}^{+}, h_{2,t}^{+}\right\}_t . \end{cases} \tag{3.5}$$

As one can remark, there are four prediction operators, P_1^{-} / P_1^{+} and P_2^{-} / P_2^{+} , as well as two updates, U^{-} and U^{+}, used for obtaining the temporal subbands. Due to the symmetry of the scheme, we propose to use symmetrical predict operators for the generation of detail subbands $h_{1,t}^{-}/h_{1,t}^{+}$ and $h_{2,t}^{-}/h_{2,t}^{+}$. We will pass in the following to the presentation of the 5-band scheme both without and with motion compensation.

3.2.1.1 General 5-band filter bank

A possible embodiment for the case when no ME/MC is considered is:

$$\begin{aligned} P_1^{-}\left\{x_{5t-2}, x_{5t}\right\}_t &= \alpha x_{5t-2} + (1-\alpha)x_{5t}, \\ P_1^{+}\left\{(x_{5t+2}, x_{5t}\right\}_t &= \alpha x_{5t+2} + (1-\alpha)x_{5t}, \\ P_2^{-}\left\{x_{5t}\right\}_t &= \beta x_{5t} - (1-\beta)x_{5t-5} \\ P_2^{+}\left\{x_{5t}\right\}_t &= \beta x_{5t} - (1-\beta)x_{5t+5}. \end{aligned}$$

This corresponds to:

$$\begin{cases} h_{1,t}^{-} = x_{5t-1} - \alpha x_{5t-2} - (1-\alpha)x_{5t} \\ h_{1,t}^{+} = x_{5t+1} - \alpha x_{5t+2} - (1-\alpha)x_{5t} \\ h_{2,t}^{-} = x_{5t-2} - \beta x_{5t} - (1-\beta)x_{5t-5} \\ h_{2,t}^{+} = x_{5t+2} - \beta x_{5t} - (1-\beta)x_{5t+5}, \end{cases} \tag{3.6}$$

where $\alpha, \beta \in (0, 1]$ are weighting factors. Denoting by $H^{-}(z)$, resp. $H^{+}(z)$, the z-transform of the previous defined filters, one can remark that, for any $\alpha, \beta \in (0, 1]$, we have $H^{-}(1) = H^{+}(1) = 0$, meaning that we have indeed four highpass filters:

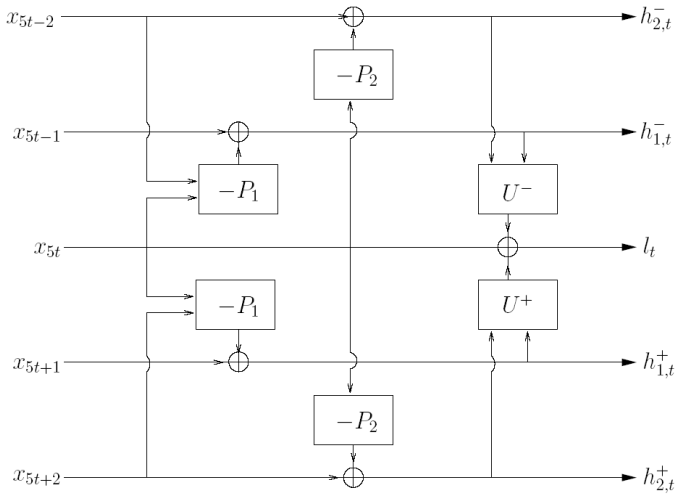


Figure 3.7: Five-band motion-compensated temporal lifting scheme.

$$\begin{cases} H_1^-(z) = z^{-1} - \alpha z^{-2} - (1-\alpha) \\ H_1^+(z) = z - \alpha z^2 - (1-\alpha) \\ H_2^-(z) = z^{-2} - \beta - (1-\beta) z^{-5} \\ H_2^+(z) = z^2 - \beta - (1-\beta) z^5 \end{cases} \tag{3.7}$$

The parameters α and β can be tuned, for example, to take into account the irregularities along motion trajectories.

For the temporal approximation subband, a very simple choice is:

$$l_t = x_{5t} + \gamma(h^-_{1,t} + h^+_{1,t}) + \delta(h^-_{2,t} + h^+_{2,t}), \tag{3.8}$$

where δ and γ are parameters in $(0,1)$ which can be tuned in order to ensure the low-pass characteristics of the filter, i.e. $L(-1) = 0$, where L is the z-transform of l_t. Using the expressions of the temporal detail subbands in Eq. (3.6), Eq. (3.8) becomes:

$$\begin{aligned} l_t &= [1 - 2\gamma(1-\alpha) - 2\delta\beta] x_{5t} + \gamma x_{5t-1} + \gamma x_{5t+1} \\ &+ (\delta - \gamma\alpha) x_{5t-2} + (\delta - \gamma\alpha) x_{5t+2} - \delta(1-\beta) x_{5t-5} - \delta(1-\beta) x_{5t+5} \end{aligned} \tag{3.9}$$

and its z-transform is given by:

$$\begin{aligned} L(z) &= [1 - 2\gamma(1-\alpha) - 2\delta\beta] + \gamma z^{-1} + \gamma z \\ &+ (\delta - \gamma\alpha) z^{-2} + (\delta - \gamma\alpha) z^2 - \delta(1-\beta) z^{-5} - \delta(1-\beta) z^5 \end{aligned} \tag{3.10}$$

From the low-pass constraint on the above expression (i.e., $L(-1) = 0$), we get the following relation between the update parameters:

$$1 - 4\gamma - 4\delta\beta + 4\delta = 0 \tag{3.11}$$

3.2.1.2 Simple implementation approach

In a simple implementation approach, we can set $\alpha = \frac{1}{2}$ and $\beta = 1$, these values respecting the high-pass filter existence condition, i.e. $H^-(1) = H^+(1)$. When no ME/MC is considered, this leads to the following predictors:

$$\begin{aligned} P_1^-(x_{5t-2}, x_{5t}) &= \frac{1}{2}(x_{5t-2} + x_{5t}), \\ P_1^+(x_{5t+2}, x_{5t}) &= \frac{1}{2}(x_{5t+2} + x_{5t}), \\ P_2^-(x_{5t-2}) = Id, &\quad P_2^+(x_{5t+2}) = Id. \end{aligned}$$

and corresponds to:

$$\begin{cases} h_{1,t}^- = x_{5t-1} - \frac{1}{2}(x_{5t-2} + x_{5t}) \\ h_{1,t}^+ = x_{5t+1} - \frac{1}{2}(x_{5t+2} + x_{5t}) \\ h_{2,t}^- = x_{5t-2} - x_{5t} \\ h_{2,t}^+ = x_{5t+2} - x_{5t}, \end{cases} \tag{3.12}$$

In this case, Eq. (3.9) becomes:

$$l_t = (1 - \gamma - 2\delta)x_{5t} + \gamma x_{5t-1} + \gamma x_{5t+1} + (\delta - \frac{\gamma}{2})x_{5t-2} + (\delta - \frac{\gamma}{2})x_{5t+2}, \tag{3.13}$$

and its z-transform is given by:

$$L(z) = (1 - \gamma - 2\delta) + \gamma z_{-1} + \gamma z + (\delta - 0.5\gamma)z_{-2} + (\delta - 0.5\gamma)z_2 \tag{3.14}$$

A pair of values satisfying the constraint in Eq.(3.11) and which minimizes the total reconstruction error is $\gamma = \delta = 1/4$, which yields the following five-tap filter:

$$l_t = \frac{1}{8}\big(x_{5t-2} + 2x_{5t-1} + 2x_{5t} + 2x_{5t+1} + x_{5t+2}\big).$$

Motion estimation and compensation is done as shown in Fig. 3.8; in this case, the analysis equations become:

$$\begin{cases} h_{1,t}^- = x_{5t-1} - \frac{1}{2}\big(\mathcal{F}(x_{5t-2}, v_{1,t^-}^-) + \mathcal{F}(x_{5t}, v_{1,t^-}^+)\big) \\ h_{1,t}^+ = x_{5t+1} - \frac{1}{2}\big(\mathcal{F}(x_{5t+2}, v_{1,t^+}^-) + \mathcal{F}(x_{5t}, v_{1,t^+}^+)\big) \\ h_{2,t}^- = x_{5t-2} - \mathcal{F}(x_{5t}, v_{2,t^-}) \\ h_{2,t}^+ = x_{5t+2} - \mathcal{F}(x_{5t}, v_{2,t^+}) \\ l_t = x_{5t} + \frac{1}{4}\big(\mathcal{F}^-(h_{1,t}^-, v_{1,t^-}^+) + \mathcal{F}^-(h_{2,t}^-, v_{2,t^-})\big) + \frac{1}{4}\big(\mathcal{F}^-(h_{1,t}^+, v_{1,t^+}^-) + \mathcal{F}^-(h_{2,t}^+, v_{2,t^+})\big) \end{cases} \tag{3.15}$$

where $\mathcal{F}(x_t, v_t)$ is the motion-prediction operator, compensating the frame x_t by projection in the direction of the motion-vector field v_t, $v_{2,t^-}/v_{2,t^+}$ are the forward / backward

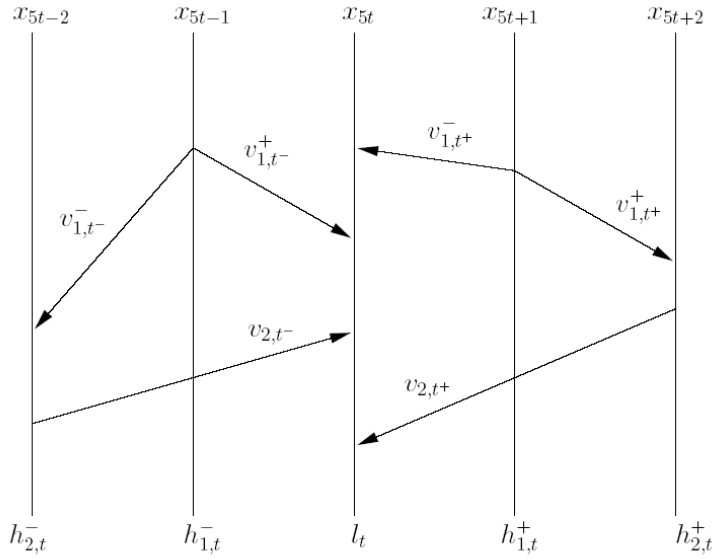


Figure 3.8: Temporal prediction using the simple implementation approach.

Haar-like motion vectors predicting x_{5t-2} / x_{5t+2}, and (v^-_{1,t^-},v^+_{1,t^-}) and (v^-_{1,t^+},v^+_{1,t^+}) are the 5/3-like pair of motion vectors predicting x_{5t-1} and respectively x_{5t+1}. The notation $\mathcal{F}^{-1}(h_t, v_t)$ corresponds to the compensation of the h_t frame in the opposite direction of the motion vector field $\boldsymbol{v}_t$.

3.2.1.3 Sliding-window implementation

Another set of satisfactory values for the existence of the high-pass filter in Eq. (3.6) (i.e. $H^-(1) = H^+(1) = 0$) and corresponding to the most selective filter, as illustrated in Fig. 3.9, is given by $\alpha = \beta = \frac{1}{2}$, values which will be considered for the sliding-window implementation, i.e.:

$$\begin{cases} h^-_{1,t} = x_{5t-1} - \dfrac{1}{2}(x_{5t-2} + x_{5t}) \\ h^+_{1,t} = x_{5t+1} - \dfrac{1}{2}(x_{5t+2} + x_{5t}) \\ h^-_{2,t} = x_{5t-2} - \dfrac{1}{2}(x_{5t} + x_{5t-5}) \\ h^+_{2,t} = x_{5t+2} - \dfrac{1}{2}(x_{5t} + x_{5t+5}), \end{cases} \tag{3.16}$$

For the temporal approximation subband, using the expressions of the temporal detail subbands in Eq. (3.16), Eq. (3.8) becomes:

$$l_t = (1-\gamma-\delta)x_{5t} + \gamma x_{5t-1} + \gamma x_{5t+1} + (\delta - \frac{\gamma}{2})x_{5t-2} + (\delta - \frac{\gamma}{2})x_{5t+2} - \frac{\delta}{2}x_{5t-5} - \frac{\delta}{2}x_{5t+5} \tag{3.17}$$

and its z-transform is given by:

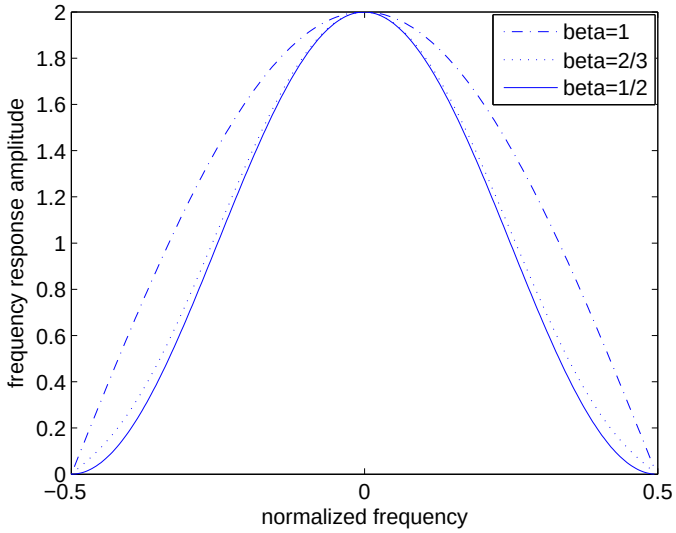


Figure 3.9: Frequency response of high-pass filters for different values of β.

$$L(z) = 1 - \gamma - \delta + \gamma z^{-1} + \gamma z^{1} + (\delta - \frac{\gamma}{2})z^{-2} + (\delta - \frac{\gamma}{2})z^{2} - \frac{\delta}{2}z^{-5} - \frac{\delta}{2}z^{5}. \tag{3.18}$$

For the new high-pass filter coefficients, the low-pass constraint derived in Eq. (3.11) becomes:

$$2\delta - 4\gamma + 1 = 0 \tag{3.19}$$

Motion estimation and compensation is done as shown in Fig. 3.10; in this case, the analysis equations become:

$$\begin{cases} h^{-}_{1,t} = x_{5t-1} - \frac{1}{2}(\mathcal{F}(x_{5t-2}, v^{-}_{1,t^{-}}) + \mathcal{F}(x_{5t}, v^{+}_{1,t^{-}})) \\ h^{+}_{1,t} = x_{5t+1} - \frac{1}{2}(\mathcal{F}(x_{5t+2}, v^{-}_{1,t^{+}}) + \mathcal{F}(x_{5t}, v^{+}_{1,t^{+}})) \\ h^{-}_{2,t} = x_{5t-2} - \frac{1}{2}(\mathcal{F}(x_{5t}, v^{+}_{2,t^{-}}) + \mathcal{F}(x_{5t-5}, v^{-}_{2,t^{-}})) \\ h^{+}_{2,t} = x_{5t+2} - \frac{1}{2}(\mathcal{F}(x_{5t}, v^{-}_{2,t^{+}}) + \mathcal{F}(x_{5t+5}, v^{+}_{2,t^{+}})) \end{cases} \tag{3.20}$$

One can remark that when there is a scene-cut, one of the predictions will be meaningless and is not used. This amounts to setting α or/and β to 0. Moreover, it requires the detection of scene-cuts and, possibly, the segmentation of the video sequence into homogeneous groups of pictures (GOP) (or spatial parts of a GOP), as detailed in [189] and in section 3.1. This means that an adaptive prediction can be realized in this manner. Note that this kind of adaptive behaviour is not yet implemented in the current version of our temporal lifting method, as it entails a more complex coding strategy.

The update operators U^{-}/U^{+} should also be applied along the motion trajectory. So, even though the structural property of the lifting scheme is that all the information available from the predict step (high-frequency frames) can be used for the update, the

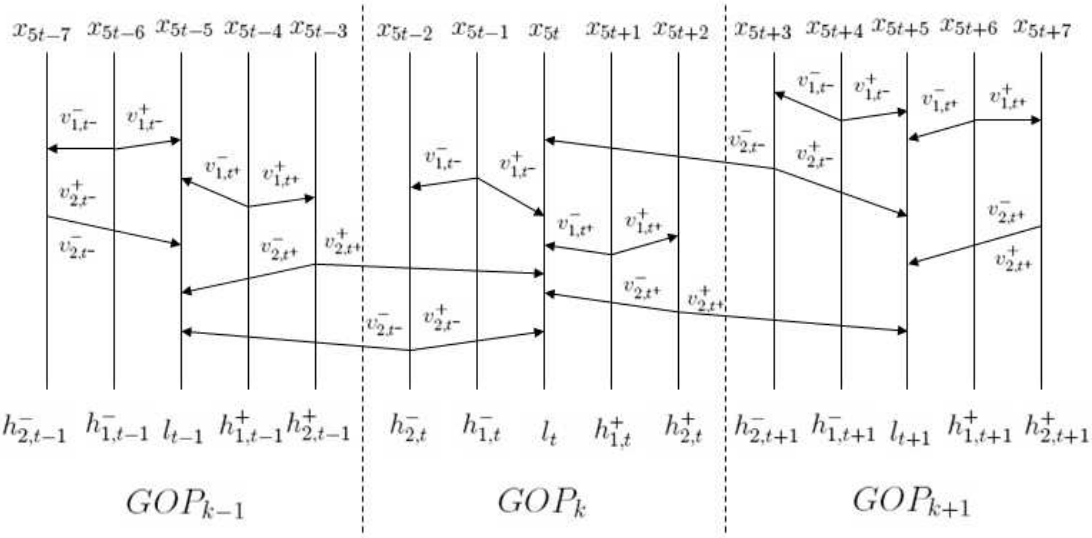


Figure 3.10: Temporal prediction using the sliding window implementation approach.

most useful information will be in the neighborhood of the pixels on the same trajectory [74]. This feature can be exploited in different ways to improve the temporal filtering in critical areas like, for example, the occlusion zones. Thus, the motion-compensated temporal approximation subband is obtained as:

$$l_t = x_{5t} + \gamma\left[\mathcal{F}^-(h^-_{1,t}, v^+_{1,t^-}) + \mathcal{F}^-(h^+_{1,t}, v^-_{1,t^+}) + \delta\left[\mathcal{F}^-(h^-_{2,t}, v^+_{2,t^-}) + \mathcal{F}^-(h^+_{2,t}, v^-_{2,t^+})\right]\right. \quad (3.21)$$

As in the case of the simple implementation, the problem rising from unconnected or multiple-connected pixels is treated as described in [83, 182].

3.2.2 Normalization factors

Let us now consider the filter normalization:

$$\widehat{l}_t = k_l l_t$$
$$\widehat{h}^-_{1,t} = k_{h1} h^-_{1,t}, \quad \widehat{h}^+_{1,t} = k_{h1} h^+_{1,t}$$
$$\widehat{h}^-_{2,t} = k_{h2} h^-_{2,t}, \quad \widehat{h}^+_{2,t} = k_{h2} h^+_{2,t}$$

Two conditions could be relevant to the goal we consider [183]. On one hand, we would like to preserve the unit norm for the impulse responses of the filters involved in the 5-band structure. On the other hand, an orthonormal structure preserves the energy of an input sequence. In particular, by considering the quantization error in each detail and approximation frame as i.i.d. random variables, the sum of reconstruction errors of the five consecutive frames should be equal to the sum of quantization errors of the approximation and detail frames:

$$\sigma^2_{x_{5t-2}} + \sigma^2_{x_{5t-1}} + \sigma^2_{x_{5t}} + \sigma^2_{x_{5t+1}} + \sigma^2_{x_{5t+2}} = \sigma^2_{\widetilde{l}_t} + \sigma^2_{\widehat{h}^+_{1,t}} + \sigma^2_{\widehat{h}^-_{1,t}} + \sigma^2_{\widehat{h}^+_{2,t}} + \sigma^2_{\widehat{h}^-_{2,t}}, \quad (3.22)$$

where σ^2_a denotes the variance of the frame a.

3.2.2.1 Simple implementation approach

According to the unit norm preservation criterion, from the system:

$$\begin{cases} k_{h1}^2 x_{5t-1}^2 + \frac{k_{h1}^2}{4} x_{5t}^2 + \frac{k_{h1}^2}{4} x_{5t-2}^2 = 1 \\ k_{h1}^2 x_{5t+1}^2 + \frac{k_{h1}^2}{4} x_{5t}^2 + \frac{k_{h1}^2}{4} x_{5t+2}^2 = 1 \\ k_{h2}^2 x_{5t-2}^2 + k_{h2}^2 x_{5t}^2 = 1 \\ k_{h2}^2 x_{5t+2}^2 + k_{h2}^2 x_{5t}^2 = 1 \\ \frac{k_l^2}{16}\left(x_{5t}^2 + x_{5t-1}^2 + x_{5t+1}^2\right) + \frac{k_l^2}{64}\left(x_{5t-2}^2 + x_{5t+2}^2\right) = 1 \end{cases} \tag{3.23}$$

the normalization constants for the first 5-band filter bank implementation are:

$$k_{h1} = \sqrt{\frac{2}{3}},\ k_{h2} = \frac{\sqrt{2}}{2},\ k_l = \frac{8}{\sqrt{14}}.$$

On the other hand, according to the energy-presenvation criterion, we can write the system:

$$\begin{cases} x_{5t} = \frac{\widehat{l}_t}{k_l} - \frac{\widehat{h}_{1,t}^+ + \widehat{h}_{1,t}^-}{4k_{h1}} - \frac{\widehat{h}_{2,t}^+ + \widehat{h}_{2,t}^-}{4k_{h2}} \\ x_{5t-1} = \frac{\widehat{l}_t}{k_l} - \frac{\widehat{h}_{1,t}^+ - 3\widehat{h}_{1,t}^-}{4k_{h1}} - \frac{\widehat{h}_{2,t}^+ - \widehat{h}_{2,t}^-}{4k_{h2}} \\ x_{5t+1} = \frac{\widehat{l}_t}{k_l} + \frac{3\widehat{h}_{1,t}^+ - \widehat{h}_{1,t}^-}{4k_{h1}} + \frac{\widehat{h}_{2,t}^+ - \widehat{h}_{2,t}^-}{4k_{h2}} \\ x_{5t-2} = \frac{\widehat{l}_t}{k_l} - \frac{\widehat{h}_{1,t}^+ + \widehat{h}_{1,t}^-}{4k_{h1}} - \frac{\widehat{h}_{2,t}^+ - 3\widehat{h}_{2,t}^-}{4k_{h2}} \\ x_{5t+2} = \frac{\widehat{l}_t}{k_l} - \frac{\widehat{h}_{1,t}^+ + \widehat{h}_{1,t}^-}{4k_{h1}} + \frac{3\widehat{h}_{2,t}^+ - \widehat{h}_{2,t}^-}{4k_{h2}} \end{cases} \tag{3.24}$$

and, applying Eq. (3.22), we get:

$$\begin{cases} \sigma_{x_{5t}}^2 = \frac{\sigma_{\widehat{l}_t}^2}{k_l^2} - \frac{\sigma_{\widehat{h}_{1,t}^+}^2 + \sigma_{\widehat{h}_{1,t}^-}^2}{16k_{h1}^2} - \frac{\sigma_{\widehat{h}_{2,t}^+}^2 + \sigma_{\widehat{h}_{2,t}^-}^2}{16k_{h2}^2} \\ \sigma_{x_{5t-1}}^2 = \frac{\sigma_{\widehat{l}_t}^2}{k_l^2} - \frac{\sigma_{\widehat{h}_{1,t}^+}^2 - 9\sigma_{\widehat{h}_{1,t}^-}^2}{16k_{h1}^2} - \frac{\sigma_{\widehat{h}_{2,t}^+}^2 - \sigma_{\widehat{h}_{2,t}^-}^2}{16k_{h2}^2} \\ \sigma_{x_{5t+1}}^2 = \frac{\sigma_{\widehat{l}_t}^2}{k_l^2} + \frac{9\sigma_{\widehat{h}_{1,t}^+}^2 - \sigma_{\widehat{h}_{1,t}^-}^2}{16k_{h1}^2} + \frac{\sigma_{\widehat{h}_{2,t}^+}^2 - \sigma_{\widehat{h}_{2,t}^-}^2}{16k_{h2}^2} \\ \sigma_{x_{5t-2}}^2 = \frac{\sigma_{\widehat{l}_t}^2}{k_l^2} - \frac{\sigma_{\widehat{h}_{1,t}^+}^2 + \sigma_{\widehat{h}_{1,t}^-}^2}{16k_{h1}^2} - \frac{\sigma_{\widehat{h}_{2,t}^+}^2 - 9\sigma_{\widehat{h}_{2,t}^-}^2}{16k_{h2}^2} \\ \sigma_{x_{5t+2}}^2 = \frac{\sigma_{\widehat{l}_t}^2}{k_l^2} - \frac{\sigma_{\widehat{h}_{1,t}^+}^2 + \sigma_{\widehat{h}_{1,t}^-}^2}{16k_{h1}^2} + \frac{9\sigma_{\widehat{h}_{2,t}^+}^2 - \sigma_{\widehat{h}_{2,t}^-}^2}{16k_{h2}^2} \end{cases} \tag{3.25}$$

which leads to the following normalization constants:

$$k_{h1} = \sqrt{266}/12,\ k_{h2} = 1,\ k_l = 1/\sqrt{5}.$$

3.2.2.2 Sliding-window implementation

For the sliding-window implementation, in order to compute the normalization factors, we should also find the update parameters, γ and δ. Three conditions are considered for their setup: one is given by the low-pass condition in Eq. (3.19); the second one is given by the minimization of the total reconstruction error $\mathcal{E}_T$, where:

$$\mathcal{E}_T = \sigma^2_{x_{5t-2}} + \sigma^2_{x_{5t-1}} + \sigma^2_{x_{5t}} + \sigma^2_{x_{5t+1}} + \sigma^2_{x_{5t+2}}. \tag{3.26}$$

The third condition is given by the preservation of the unit norm of the impulses responses of the filters involved in the 5-band structure.

From the system:

$$\begin{cases} x_{5t} = l_t - \gamma(h^-_{1,t} + h^+_{1,t}) - (2\gamma - \frac{1}{2})(h^-_{2,t} + h^+_{2,t}) \\ x_{5t+2} = \frac{1}{2}(l_t + l_{t+1}) - \frac{\gamma}{2}(h^-_{1,t} + h^+_{1,t} + h^-_{1,t+1} + h^+_{1,t+1}) - \\ \qquad (\gamma - \frac{1}{4})(h^-_{2,t} + h^-_{2,t+1} + h^+_{2,t+1}) - (\gamma - \frac{5}{4})h^+_{2,t} \\ x_{5t-2} = \frac{1}{2}(l_t + l_{t-1}) - \frac{\gamma}{2}(h^-_{1,t} + h^+_{1,t} + h^-_{1,t-1} + h^+_{1,t-1}) - \\ \qquad (\gamma - \frac{1}{4})(h^+_{2,t} + h^-_{2,t-1} + h^+_{2,t-1}) - (\gamma - \frac{5}{4})h^-_{2,t} \\ x_{5t+1} = \frac{3l_t + l_{t+1}}{4} - \frac{3\gamma h^-_{1,t}}{4} - (\frac{3\gamma}{4} - 1)h^+_{1,t} - \\ \qquad \frac{\gamma}{4}(h^-_{1,t+1} + h^+_{1,t+1}) - (\frac{3\gamma}{2} - \frac{3}{8})h^-_{2,t} - \\ \qquad (\frac{3\gamma}{2} - \frac{7}{8})h^+_{2,t} - (\frac{\gamma}{2} - \frac{1}{8})(h^-_{2,t+1} + h^+_{2,t+1}) \\ x_{5t-1} = \frac{3l_t + l_{t-1}}{4} - \frac{3\gamma h^+_{1,t}}{4} - (\frac{3\gamma}{4} - 1)h^-_{1,t} - \\ \qquad \frac{\gamma}{4}(h^-_{1,t-1} + h^+_{1,t-1}) - (\frac{3\gamma}{2} - \frac{3}{8})h^+_{2,t} - \\ \qquad (\frac{3\gamma}{2} - \frac{7}{8})h^-_{2,t} - (\frac{\gamma}{2} - \frac{1}{8})(h^-_{2,t-1} + h^+_{2,t-1}) \end{cases} \tag{3.27}$$

and Eq. (3.26) and under the hypothesis of decorrelated quantization errors, we get:

$$\mathcal{E}_T = \frac{13\sigma^2_{k_l}}{4k^2_l} + \left(\frac{13\gamma^2}{2} - 3\gamma + 2\right)\frac{\sigma^2_{k_{h1}}}{k^2_{h1}} + \left(26\gamma^2 - 20\gamma + \frac{46}{8}\right)\frac{\sigma^2_{k_{h2}}}{k^2_{h2}} \tag{3.28}$$

where $\sigma^2_{k_l}, \sigma^2_{k_{h1}}, \sigma^2_{k_{h2}}$ represent the variation of the subband quantization error. The third condition, i.e. preservation of the unit norm of the impulse responses of the filters involved in the 5-band structure reads:

$$\begin{cases} k_{h1}^2 x_{5t-1}^2 + \dfrac{k_{h1}^2}{4} x_{5t}^2 + \dfrac{k_{h1}^2}{4} x_{5t-2}^2 = 1 \\ k_{h1}^2 x_{5t+1}^2 + \dfrac{k_{h1}^2}{4} x_{5t}^2 + \dfrac{k_{h1}^2}{4} x_{5t+2}^2 = 1 \\ k_{h2}^2 x_{5t-2}^2 + \dfrac{k_{h2}^2}{4} x_{5t}^2 + \dfrac{k_{h2}^2}{4} x_{5t-5}^2 = 1 \\ k_{h2}^2 x_{5t+2}^2 + \dfrac{k_{h2}^2}{4} x_{5t}^2 + \dfrac{k_{h2}^2}{4} x_{5t+5}^2 = 1 \\ k_l^2 \left(-3\gamma + \dfrac{3}{2}\right)^2 x_{5t}^2 + k_l^2 \gamma^2 \left(x_{5t-1}^2 + x_{5t+1}^2\right) + k_l^2 \left(\dfrac{3\gamma}{2} - \dfrac{1}{2}\right)^2 \left(x_{5t-2}^2 + x_{5t+2}^2\right) + \\ k_l^2 \left(\dfrac{1}{4} - \gamma\right)^2 \left(x_{5t-5}^2 + x_{5+5}^2\right) = 1 \end{cases} \tag{3.29}$$

According to this criterion, the normalization constants for the temporal detail subbands are:

$$k_{h1} = k_{h2} = 0.8165.$$

Considering equal variation for the temporal detail subbands and minimizing the $\mathcal{E}_T$ expression in Eq. (3.28) with respect to γ, we obtain:

$$\gamma = \frac{23}{65}.$$

Replacing the value of γ in Eq. (3.19) and respectively Eq. (3.29), we get $\delta = \frac{27}{130}$ and, respectively, $k_l = 1.4647$.

According to the energy-preservation approach described in Eq.(3.22), the system:

$$\begin{cases} x_{5t} = \dfrac{\widehat{l}_t}{k_l} - \dfrac{23\widehat{h}_{1,t}^+ + 23\widehat{h}_{1,t}^-}{65k_{h1}} - \dfrac{27\widehat{h}_{2,t}^+ + 27\widehat{h}_{2,t}^-}{130k_{h2}} \\ x_{5t-1} = \dfrac{\widehat{l}_{t-1} + 3\widehat{l}_t}{4k_l} - \dfrac{23\widehat{h}_{1,t-1}^+ + 23\widehat{h}_{1,t-1}^- + 69\widehat{h}_{1,t}^+ - 191\widehat{h}_{1,t}^-}{260k_{h1}} - \dots \\ \qquad \dots - \dfrac{27\widehat{h}_{2,t-1}^+ + 27\widehat{h}_{2,t-1}^- + 81\widehat{h}_{2,t}^+ - 179\widehat{h}_{2,t}^-}{520k_{h2}} \\ x_{5t+1} = \dfrac{3\widehat{l}_t + \widehat{l}_{t+1}}{4k_l} - \dfrac{-191\widehat{h}_{1,t}^+ + 69\widehat{h}_{1,t}^- + 23\widehat{h}_{1,t+1}^+ + 23\widehat{h}_{1,t+1}^-}{260k_{h1}} - \dots \\ \qquad \dots - \dfrac{-179\widehat{h}_{2,t}^+ + 81\widehat{h}_{2,t}^- + 27\widehat{h}_{2,t+1}^+ + 27\widehat{h}_{2,t+1}^-}{520k_{h2}} \\ x_{5t-2} = \dfrac{\widehat{l}_{t-1} + \widehat{l}_t}{2k_l} - \dfrac{23\widehat{h}_{1,t-1}^+ + 23\widehat{h}_{1,t-1}^- + 23\widehat{h}_{1,t}^+ + 23\widehat{h}_{1,t}^-}{130k_{h1}} - \dots \\ \qquad \dots - \dfrac{27\widehat{h}_{2,t-1}^+ + 27\widehat{h}_{2,t-1}^- + 27\widehat{h}_{2,t}^+ - 233\widehat{h}_{2,t}^-}{260k_{h2}} \\ x_{5t+2} = \dfrac{\widehat{l}_t + \widehat{l}_{t+1}}{2k_l} - \dfrac{23\widehat{h}_{1,t}^+ + 23\widehat{h}_{1,t}^- + 23\widehat{h}_{1,t+1}^+ + 23\widehat{h}_{1,t+1}^-}{130k_{h1}} - \dots \\ \qquad \dots - \dfrac{-233\widehat{h}_{2,t}^+ + 27\widehat{h}_{2,t}^- + 27\widehat{h}_{2,t+1}^+ + 27\widehat{h}_{2,t+1}^-}{260k_{h2}} \end{cases} \tag{3.30}$$

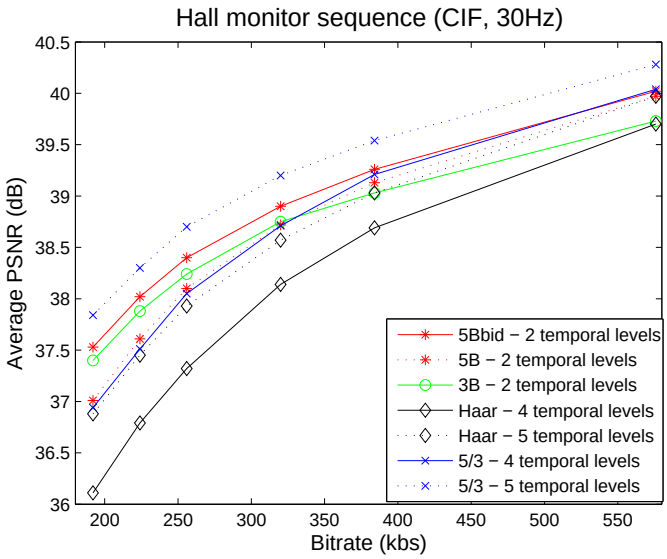


Figure 3.11: Rate-distortion comparison for the sequence Hall monitor(CIF, 30Hz).

yields:

$$k_{h1} = 0.9692,\ k_{h2} = 1.0375,\ k_l = 1.8028.$$

3.2.3 Experimental results

For the experiments, we have considered three low-motion sequences: Hall monitor (CIF format at 30 Hz), Bridge (close) monitor (CIF format at 30 Hz) and Apple (QCIF format at 7.5 Hz).

The tests have been made in the framework of the MSRA [212] video codec. For our simulations we have used only the t+2D video approach. Motion estimation is block-based and the motion-vector fields have been estimated with $1/4$ pixel accuracy. All video sequences used in our tests have been decomposed with a 4-level $9/7$ spatial filter. The spatio-temporal approximation and detail subband wavelet coefficients have been encoded using the 3D-ESCOT [216] algorithm. The bitrate test-points for the sequences used in our simulations correspond to those defined for the MPEG standardization [215].

In the simulations, we have chosen two levels of temporal subband decomposition with the proposed filtering scheme due to the length of the filters. The GOP size is thus composed of 25 frames. For three levels (125-frame GOP), the correlation between the frames is too weak to ensure a good motion compensation of the detail frames.

In Fig. 3.11 and Fig. 3.12, we have compared the results obtained for the Hall monitor (CIF, 30 Hz) and Bridge (close) monitor (CIF, 30 Hz) sequences using the following filters for temporal subband decomposition :

- ⋆ 5-band filter with sliding-window implementation (”5Bbid”).
- ⋆ 5-band filter with simple implementation (”5B”).
- ⋆ 3-band filter (”3B”).
- ⋆ dyadic $5/3$ filter-bank(”$5/3$”).

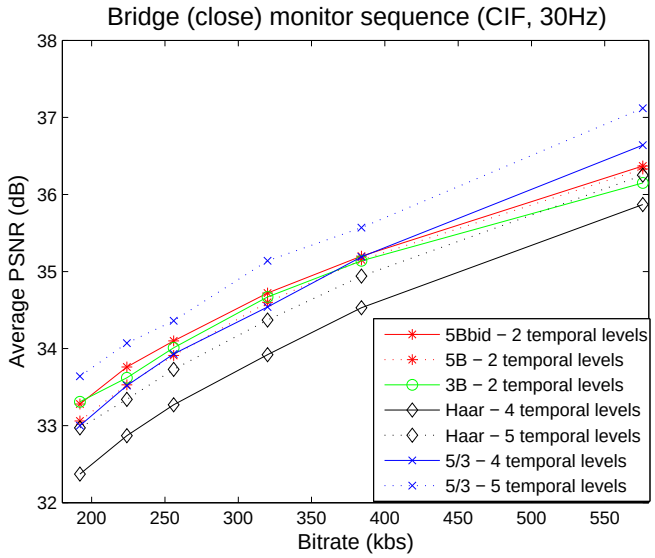


Figure 3.12: Rate-distortion comparison for Bridge (close) monitor (CIF, 30Hz) sequence.

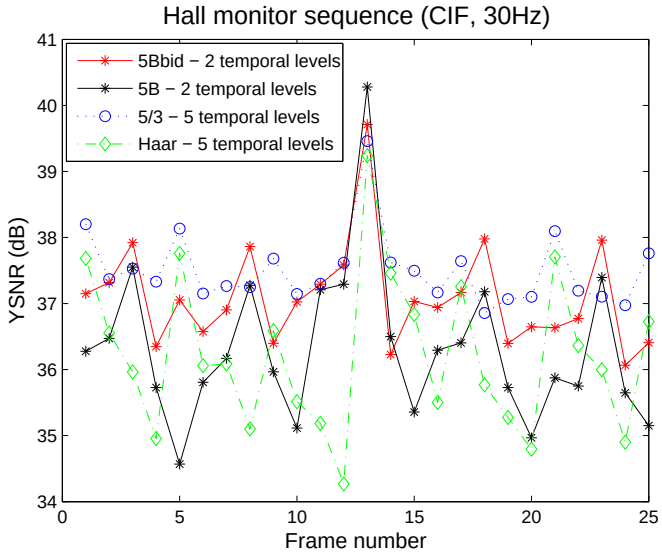
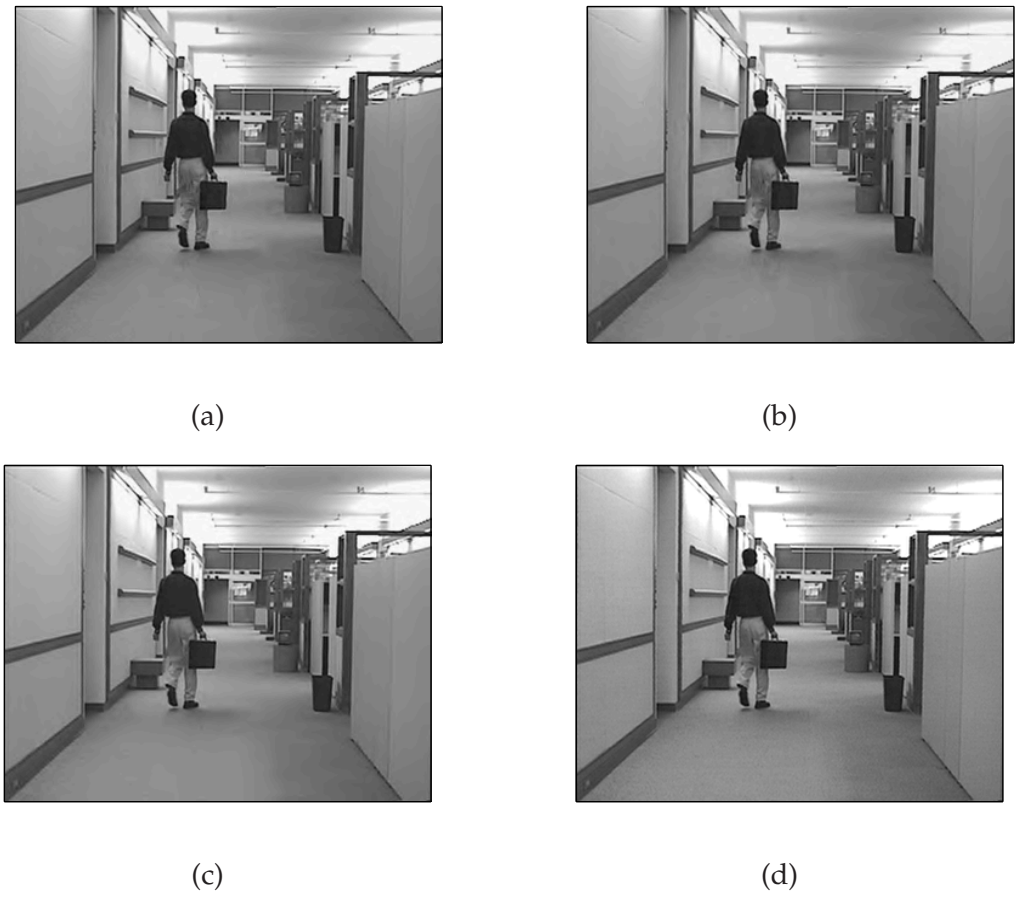


Figure 3.13: YSNR variation of the frames 26-50 of Hall monitor (CIF, 30Hz) sequence at 192 kbs.

⋆ dyadic Haar filter-bank (”Haar”).

The tests have been run for different numbers of decomposition levels in order to ensure an almost equivalent number of approximation frames in the encoded bitstream. As can be seen in Fig. 3.11, the quality in YSNR obtained with two temporal levels of decomposition with the 5-band filter is comparable with that obtained by four and, respectively, five levels of decomposition of the dyadic filters Haar and 5x3 (i.e., GOPs formed by 16 and 32 frames, respectively) and three levels of decomposition (27-frame GOP) of the 3-band filter. The same conclusion can be drawn from Fig. 3.12 and Fig. 3.15, where the results for the Bridge (close) monitor (CIF, 30 Hz) and Apple (QCIF, 7.5 Hz) sequences are presented.



(a) (b)

(c) (d)

Figure 3.14: Approximation frames from Hall monitor (CIF, 30Hz) sequence: (a) 5 decomposition levels with Haar filter (YSNR = 37.88 dB); (b) 5 decomposition levels with 5x3 filter (YSNR = 38.14 dB); (c) two decomposition levels with 5-band filter (YSNR = 38.75 dB) ; (d) original frame.

We have chosen two levels of temporal subband decomposition with the proposed filtering scheme due to the length of the filter; for three levels (125-frame GOP), the correlation between the frames is too weak to assure a good motion-compensation of the detail frames.

In Fig. 3.13 we analyse the YSNR variation of 25 frames from Hall monitor sequence

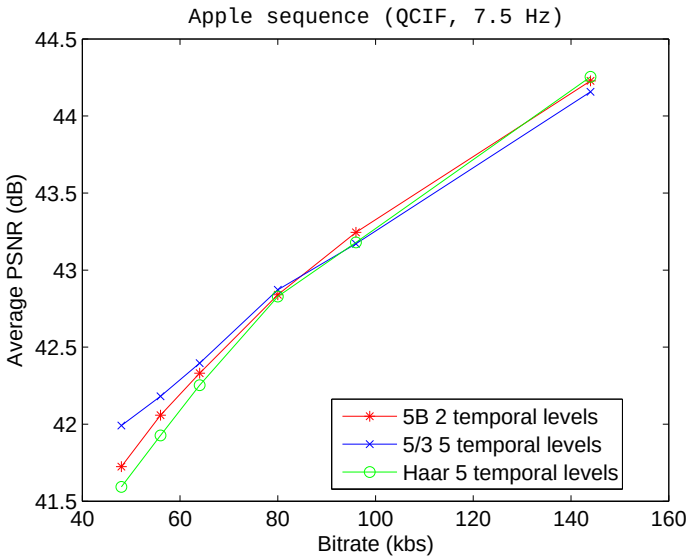


Figure 3.15: Rate-distortion comparison for Apple (QCIF, 7.5 Hz) sequence.

(subsequence made of the frames between 26-50 positions in the original sequence) at 192 kbs, for five temporal decomposition levels of Haar and 5/3 filters and 2 levels of 5-band filter with 4 (”5Bbid”) and, respectively, 2 (”5B”) bidirectional predictors. In order to have a reference for comparing the quality of the approximation subbands, we have to process the test sequence from different starting frames for the different filters. The results in Fig. 3.13 are obtained with an approximation subband synchronized with the 50^{th} frame in the original sequence Hall monitor sequence. One can see that the YSNR value obtained with the 5-band filter bank for the approximation subband is approximately 1 dB greater than those obtained with the dyadic decompositions. Moreover, the sliding-window implementation of the 5-band filter bank shows an improved quality of the temporal detail subbands in comparison with the simple implementation.

3.2.4 Conclusion

We have presented in this section a 5-band temporal lifting structure, allowing flexible scalability factors of order five in a MCTF video codec. As it can be seen from the experimental results, the proposed scheme has similar performance with the dyadic Haar and 5/3 filters and the 3-band temporal decomposition. Also, it gives better coding efficiency for the temporal approximation subbands leading then to an improved temporal scalability. It can be successfully used in certain applications, such as the encoding of video surveillance sequences, where the motion activity is weak in most cases.

All the temporal-processing contributions of this thesis presented till now are based on a linear approach to the lifting scheme. However, when a sequence presents complex motion transitions, this linearity assumption no longer holds. In the following, we will present an adaptive temporal prediction scheme which attempts to alleviate this problem.

3.3 LMS-based adaptive prediction for scalable video coding

Recall that MCTF exploits the temporal interframe redundancy by applying an open-loop temporal wavelet transform along the motion trajectories of the frames in a video sequence (see section 2.3.1). As explained in section 2.3.1.1, generally a block-matching algorithm is used for motion estimation. Even though a bidirectional prediction and mode selection can be used, and powerful algorithms such as Hierarchical Variable Size Block Matching (HVSBM) [44] or differential coding of motion vectors [196], blocking artefacts are still present. In addition, ringing artefacts appear at low bitrates and ghosting artefacts can be present as well in the approximation subbands.

In order to avoid such artefacts, motion-compensation solutions such as weighted-average update operator [184] or overlapped block motion compensation [211] have been proposed, alleviating, but not completely solving, this problem. In the following we propose to improve the prediction of the high-frequency temporal subband frames by using an adaptive filter bank. This proposal has been presented in the proceedings of the ICAASP'06 [186] and EUSIPCO'06 [185] conferences and represents a joint-work between Bilkent University and Telecom Paris.

There are various subband adaptive-filter structures which perform adaptive filtering in the subbands [206, 92, 20]. We will use the least mean squares (LMS) type FIR based adaptive filters proposed in [79]. In [79], the adaptation scheme is developed for 2D image compression. In the following, we propose to extend the adaptation scheme to the motion-compensated $t + 2D$ video coding case.

The proposed LMS-based adaptive prediction is used in the temporal prediction step in the lifting framework. Note, however, that it can as well be applied to any temporal prediction scheme. The detail subband-frame pixels are predicted using a set of pixels from the neighbouring previous and future frames. This way, the spatio-temporal filters are adapted to better take into account the changing input conditions, in particular moving objects having high contrast with the background or illumination variations. In such cases, fixed coefficient filter structures result in poor image quality with low PSNR values. The proposed scheme substantially improves the image quality while increasing the PSNR, as the number of pixels used for the adaptation is increased. Moreover, a special edge (contrast)-sensitive adaptation methodology is developed for the two-pixel case which introduces a low-cost alternative to adaptation with a larger number of pixels as in [80].

3.3.1 Adaptive prediction

As in the previous sections, we denote by x_t the original frames, t being the time index, by h_t and l_t the high-frequency (detail) and low-frequency (approximation) subband frames, respectively, and by $\mathbf{n}$ the spatial index inside a frame. For the purpose of illustration, we have used in our experiments a biorthogonal 5/3 filter bank for the temporal decomposition (see section 2.3.1.3). The temporal motion-compensated filtering in this case is illustrated in Fig. 3.16, where $\mathbf{v}_t^+(\mathbf{n})$ denotes the forward motion vector (MV) predicting the position $\mathbf{n}$ in the $(2t+1)^{th}$ frame from the $2t^{th}$ frame, and $\mathbf{v}_t^-(\mathbf{n})$ denotes the backward MV predicting the same position in the $(2t+1)^{th}$ frame from the $(2t+2)^{th}$ frame.

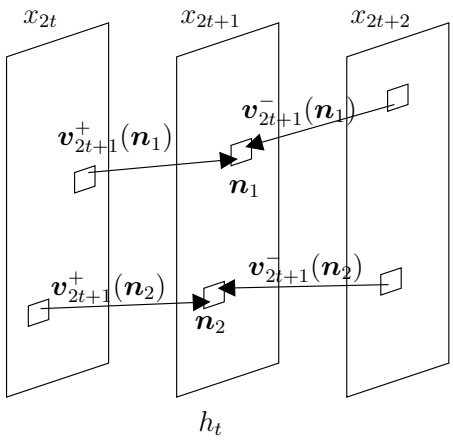


Figure 3.16: MCTF with bidirectional lifting steps.

3.3.1.1 Adaptive filter-bank structure

We first introduce an FIR estimator for $x_{2t+1}(\mathbf{n})$ by using for prediction a set of pixels from the neighboring $x_{2t}(\mathbf{n})$ and $x_{2t+2}(\mathbf{n})$ frames (note that no motion compensation is involved at this point in the prediction):

$$\hat{x}_{2t+1}(\mathbf{n}) = \sum_{\mathbf{k}\in\mathcal{S}} w_{2t,\mathbf{n},\mathbf{k}} x_{2t}(\mathbf{n}-\mathbf{k}) + \sum_{\mathbf{k}'\in\mathcal{S}'} w_{2t+2,\mathbf{n},\mathbf{k}} x_{2t+2}(\mathbf{n}-\mathbf{k}') \tag{3.31}$$

where the w filter coefficients are adaptively tuned using an LMS-type algorithm [53]. In the above equation, summations are carried out over appropriate spatial neighborhoods, $\mathcal{S}$ and $\mathcal{S}'$, in the $2t^{th}$ and $(2t+2)^{th}$ image frames, respectively. The adaptive estimator for $h_t(\mathbf{n})$ is illustrated in Fig. 3.17.

The FIR normalized LMS adaptation is performed in a conventional manner as follows:

$$\hat{\mathbf{w}}(\mathbf{n}+\mathbf{1}) = \hat{\mathbf{w}}(\mathbf{n}) + \mu \frac{\tilde{\mathbf{x}}_t(\mathbf{n}) e(\mathbf{n})}{||\tilde{\mathbf{x}}_t(\mathbf{n})||^2} \tag{3.32}$$

where $\hat{\mathbf{w}}(\mathbf{n})$ is the filter coefficient vector at image location $\mathbf{n}$, and the vector $\tilde{\mathbf{x}}_t(\mathbf{n})$ contains the pixels within the chosen neighborhoods at the $2t^{th}$ and $(2t+2)^{th}$ frames. The vector $\mathbf{1}$ in Eq.(3.32) represents a unit increment in the image index. The scalar μ determines the step size of the adaptive algorithm. It is well known that the convergence speed is low when μ is small, but the steady-state error is smaller. For large values of μ, the opposite happens, and the convergence speed increases with a higher steady-state error. There are various methods to change the value of μ during adaptation in the LMS algorithm [113, 11]. Usually the value of μ can be set to a number between 1 and 2 initially, as depicted in Eq. (3.33):

$$\mu = \begin{cases} 0.4, & \Delta_{\tilde{x}} < 10 \\ 0.6, & 10 \le \Delta_{\tilde{\mathbf{x}}} < 30 \\ 0.8, & 30 \le \Delta_{\tilde{\mathbf{x}}} < 80 \\ 1.0, & 80 \le \Delta_{\tilde{\mathbf{x}}} < 200 \\ 1.2, & 200 \le \Delta_{\tilde{\mathbf{x}}} < 256 \end{cases} \tag{3.33}$$

where:

$$\Delta_{\tilde{x}} = max(\tilde{\mathbf{x}}(\mathbf{n})) - min(\tilde{\mathbf{x}}(\mathbf{n})) \tag{3.34}$$

and it can be gradually decreased to a smaller value between 0 and 1. In our case, computational cost can be reduced by omitting the normalization with $||\tilde{\mathbf{x}}_\mathbf{t}(\mathbf{n})||^2$ and selecting a μ close to zero.

The detail frame $\mathbf{h_t}$ is given by

$$h_t(\mathbf{n}) = x_{2t+1}(\mathbf{n}) - \hat{x}_{2t+1}(\mathbf{n}) \tag{3.35}$$

and

$$e(\mathbf{n}) = h_t(\mathbf{n}) = x_{2t+1}(\mathbf{n}) - \tilde{\mathbf{x}}_t^T(\mathbf{n})\hat{\mathbf{w}}(\mathbf{n}) \tag{3.36}$$

When only two pixels (one from the $2t^{th}$ and another one from the $(2t+2)^{th}$ frame) are used for estimation, an edge-sensitive and/or motion-sensitive strategy is followed. In this case, input pixel values are compared to each other as well as to $x_{2t+1}(\mathbf{n})$. If one of the input pixel values is significantly different from $x_{2t+1}(\mathbf{n})$, then it is not used during the prediction process. This approach provides robustness against motion-compensation errors as well.

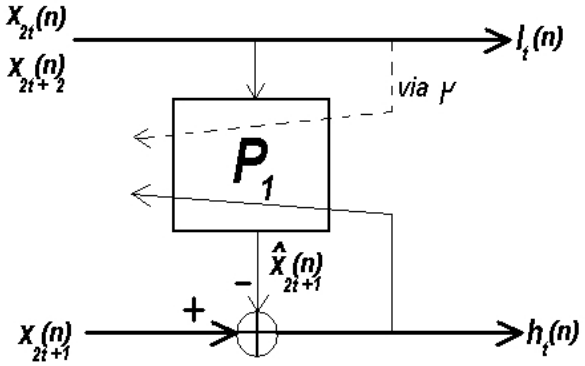


Figure 3.17: Adaptive estimator.

3.3.1.2 Motion-compensated adaptive predictor

In order to take into account the temporal filtering, as illustrated in Fig. 3.18, we rewrite the prediction Eq. (3.31) using the pixels matched to $\mathbf{n}$ by the motion-estimation process:

$$\hat{x}_{2t+1}(\mathbf{n}) = \sum_{\mathbf{k}} w_{2t,\mathbf{n},\mathbf{k}} x_{2t}(\mathbf{n} - \mathbf{k} - \mathbf{v}^+_{2t+1}(\mathbf{n})) + \sum_{\mathbf{k}} w_{2t+2,\mathbf{n},\mathbf{k}} x_{2t+2}(\mathbf{n} - \mathbf{k} - \mathbf{v}^-_{2t+1}(\mathbf{n})) \tag{3.37}$$

A great flexibility for the adaptation scheme is achieved by varying the number of pixels in the selected neighborhood, as illustrated in Fig. 3.19. Lighter pixels in the left and right images are the corresponding motion-compensated pixels of the pixel $\mathbf{n}$ at the $(2t+1)^{th}$ subband frame. The adaptation window is extended to the darker pixels to enhance the robustness of the proposed algorithm.

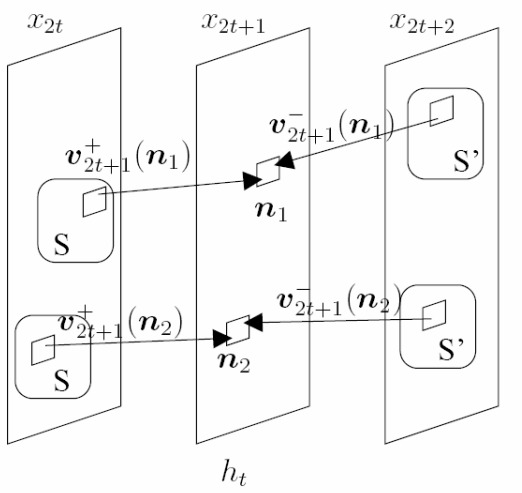


Figure 3.18: Adaptive MCTF with bidirectional lifting steps.

3.3.2 Experimental results

For the simulations, we have considered four representative test video sequences: Foreman (CIF, 30 Hz), Mobile (CIF, 30 Hz), Harbour (4CIF, 60 Hz) and Crew (4CIF, 60 Hz), which have been selected for their different motion, contrast and texture characteristics. The tests have been made in the framework of the MSRA [9] video codec. For our simulations, we have used only the $t + 2D$ video coding capability of this codec. The experiments have been run for 5 temporal decomposition levels, considering both MC and MCless (i.e., without motion compensation) temporal filtering for the CIF sequences and only MCless mode for the 4CIF ones. For the MC case, the motion estimation is block-based, and the motion-vector fields have been estimated with $1/4$ pixel accuracy. For comparisons, the video sequences used in our tests have been decomposed with a 5-level $5/3$ temporal decomposition. The temporal approximation subbands have been spatially decomposed over 5 levels with biorthogonal $9/7$ wavelets for both adaptive LMS and $5/3$ filtering schemes.

Rate-distortion curves for Harbour and Crew sequences are presented in Fig. 3.20 and Fig. 3.21, respectively. In the Harbour sequence, several foreground objects move while occluding the contrasting background. Similarly, in the Crew sequence, sudden flashes of light reflect from the crew, resulting in a high contrast between successive frames. For these two situations, the adaptation scheme yields significantly higher PSNR values compared to the no adaptive case.

The average YSNR values in Tab. 3.3 and Tab. 3.4 are computed on slightly smaller frames to reduce the inaccuracies due to frame boundaries. There is a relatively small increase in YSNR values for the two-pixel adaptation method when compared to the no-adaptation case. However, the quality of the image frames is substantially improved, especially for those segments of the video sequences where there is high contrast between moving objects and the background (see Fig. 3.22 (a), (b) and (c)). Indeed, there is a high contrast between the black moving train and the white calendar. The ghosts around the tunnel and edges of the train are removed even with a 2-pixel adaptation.

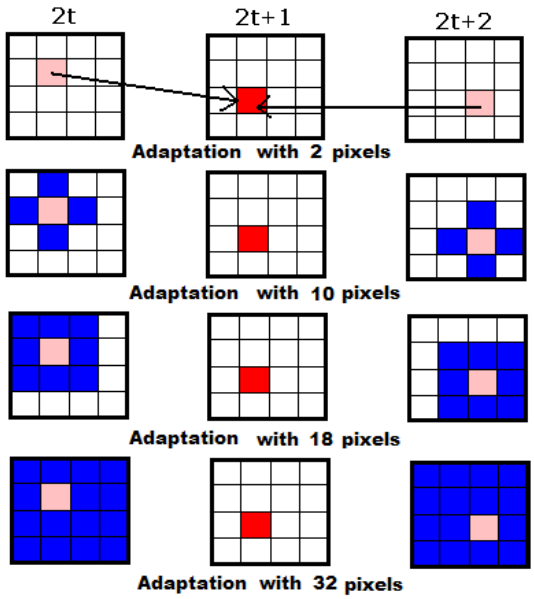


Figure 3.19: Adaptation scheme with 2, 10, 18 and 32 pixels.

YSNR	Mobile sequence				
	CIF			QCIF	
	30 Hz	15 Hz		15 Hz	7.5 Hz
BitRate(kbps)	384	256	128	64	48
NoAdapt.(dB)	29.52	28.63	25.84	24.14	23.63
Adapt.2px(dB)	30.38	29.11	26.22	24.42	23.77
Adapt.10px(dB)	31.17	30.17	26.93	24.98	24.03
Adapt.18px(dB)	32.02	31.23	27.44	25.39	24.40
Adapt.32px(dB)	33.12	32.20	28.01	25.88	24.86

Table 3.3: Rate-distortion results for Mobile (CIF, 30Hz) sequence.

YSNR	Foreman sequence				
	CIF			QCIF	
	30 Hz	15 Hz		15 Hz	7.5 Hz
BitRate(kbps)	256	192	96	48	32
NoAdapt.(dB)	35.06	34.51	31.83	30.21	29.60
Adapt.2px(dB)	35.71	34.96	32.11	30.36	29.67
Adapt.10px(dB)	36.54	35.67	32.49	30.69	29.82
Adapt.18px(dB)	37.31	36.21	33.04	30.93	30.05
Adapt.32px(dB)	38.16	36.92	33.82	31.48	30.36

Table 3.4: Rate-distortion results for Foreman (CIF, 30Hz) sequence.

The improvement increases for larger adaptation neighborhood. The same situation can be remarked in Fig. 3.23, where the quality of the frames is improved by the nonlinear filter-based prediction.

All the YSNR values reported in the tables and figures mentioned above are obtained

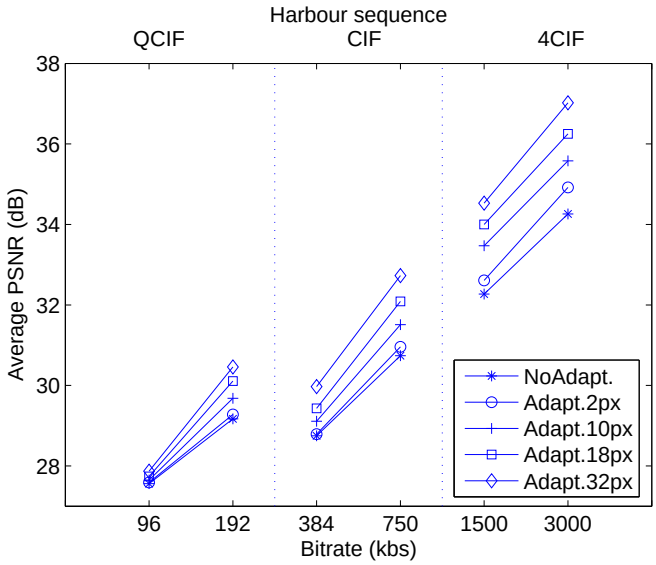


Figure 3.20: Rate-distortion comparison for Harbour (4CIF, 60 Hz) sequence.

by the adaptation method with the filter coefficients obtained from open-loop MCTF. As one can remark, the results obtained with only 2-pixels adaptation are in general 0.5 dB higher than those obtained when no adaptation is performed, as shown in Tab. 3.3 and Tab. 3.4.

3.3.3 Conclusion

We have presented in this section an LMS-based adaptive-prediction method and used it in the temporal prediction step for scalable video coding. The pixels of temporal detail subband frames are optimally predicted by using a set of pixels from the neighbouring subband frames. We have illustrated our proposed approach on a bidirectional prediction scheme, but the set of pixels for adaptation can be chosen from any number of frames involved in a longer-term prediction. Experimental results show that even for two-pixel adaptation case, the visual quality of the reconstructed frames is improved. A trade-off between compression efficiency and additional complexity coming from a larger adaptation window can be done, according to the target application. Significant PSNR improvements have been obtained for sequences with high contrast between various segments within the sequence and varying illumination conditions.

As mentioned in section 2.3 and seen so far from the previous sections, the $t + 2D$ wavelet video-coding schemes [177], [134, 44] provide a high coding performance. We propose in the following to use the MCTF-based video-coding principles for the compression of multispectral-satellite sequences. This proposal is pertinent, as $t + 2D$-based coding efficiency is strongly related to the correlation of the data being processed and starting from the assumption of a spectral correlation, we propose to decorrelate it using lifting-based methods inspired from temporal processing.

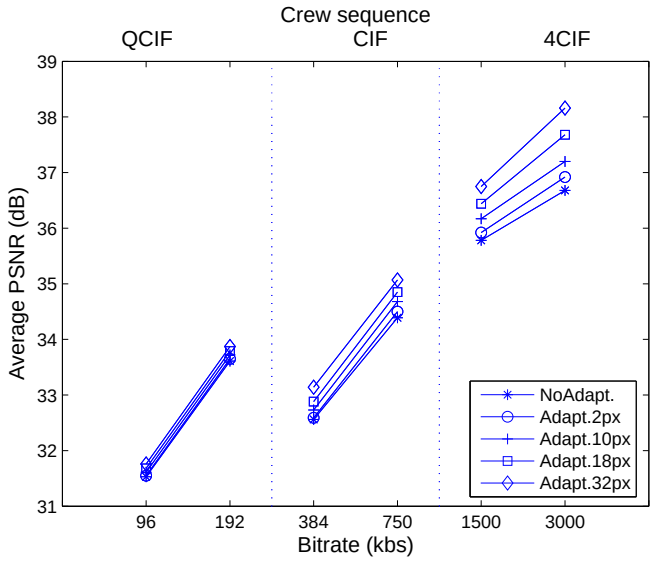


Figure 3.21: Rate-distortion comparison for Crew (4CIF, 60 Hz) sequence.

3.4 Video compression for multispectral satellite sequences

We have seen in section 2.3 that the lifting-based coding schemes are succesfully used for image [8] and video [211] coding. Now we will focus our interest on another type of data, namely multispectral and multitemporal satellite sequences, as nowadays, optical sensors cover often a localized area several times a year. Thus, it is essential to provide tools that are able to compress these multispectral and multitemporal sequences in order to store large amounts of data. This problem is starting to be adressed since remote-sensing data is significantly increasing.

As an example, one multispectral image taken by the Thematic Mapper (TM) sensor from the Landsat V satellite has the a size greater than 200 MB. New sensors have been built in order to achieve higher radiometric precision and also better spatial and spectral resolutions. Images composed of tens, or even hundreds, of spectral bands are very important due to the information they provide on the nature of the ground, but they are very hard to handle, because of the huge storage resources they require. These problems can be significantly reduced by using some form of data compaction or data compression [133]. Lossless coding is a reversible process in which the original data can always be recovered from the encoded data without any loss of information. This type of coding takes advantage of the statistical redundancy of data only and, as a result, the achieved compression ratio rarely exceeds $3:1$, which cannot solve many data-storage issues. A higher compaction is assured by lossy coding, which very often reaches compression ratios of $10:1$ and even more. However, in this case, we have to accept some loss of information in order to take advantage of the increased efficiency: the compression ratio is inversely proportional to the image quality.

Compression of hyperspectral and multispectral images has been extensively studied using several techniques, such as wavelet coding [21, 76], adaptive coding [10], DCT coding [85], DPCM [51, 159] or statistical models [66].

As in the case of video compression, where, in order to achieve data compression, one has to take advantage of redundancy in the temporal and spatial domains, the cod-

(a)

(b)

(c)

Figure 3.22: Detail from the Mobile (CIF, 30Hz) sequence at 128 kbs: (a) no adaptation, (b) 2-pixels adaptation, (c) 10-pixels adaptation.

(a)

(b)

(c)

Figure 3.23: Frame extracted from Foreman (CIF, 30Hz) sequence: (a) original, (b) 5/3 filter-bank temporal prediction and (c) nonlinear temporal prediction using a 10-pixel adaptation scheme.

ing techniques applied to multispectral data take advantage of the presence of two redundancy sources: spatial correlation among neighboring pixels in the same spectral band and spectral correlation among different bands at the same spatial location. Several

strategies can be adopted to separately or jointly exploit these sources of redundancy, but a cascade of spectral and spatial decorrelation is usually preferred, as in the case of three dimensional transform coding [98], which is based on the combination of two transforms that successively exploit inter-band and intra-band correlation. The Karhunen-Loève Transform (KLT) is usually preferred as a first step, due to its decorrelation capability, while for the second step, the Discrete Wavelet Transform and Discrete Cosine Transform (DCT) have been widely studied, also on account of their important role in international image-coding standards. In particular, the state-of-the-art JPEG2000 [8] standard can be successfully applied to intra-band coding, since it can heavily compress the less significant image planes generated by the KLT (i.e., those associated with smaller eigenvalues) with a negligible visual distortion. These planes are in fact characterized by very low dynamics, and their number usually increases with the number of bands, thus making the use of multidimensional transform coding attractive in high spectral-resolution imagery. Also, in [67], Principal Component Analysis (PCA) is deployed in JPEG2000 in order to provide spectral decorrelation, as well as spectral dimensionality reduction. However, when a multitemporal sequence of satellite images is available, as in our case, the correlation in the temporal direction also needs to be exploited.

In the following we propose to evaluate the performance of still-image (JPEG2000 [8]) and $t+2D$ video [210] compression (based on wavelet tools) on SPOT 1, 2 and 4 sequences. The results of this study have been presented in the proceedings of the IEEE ISSCS'05 [87] conference. For the video approach, our framework supposes $t+2D$ wavelet decomposition in the tempo-spatial domain and KLT decorrelation in the spectral domain. For the experiments, we have used the $5/3$ biorthogonal filter bank and DCT for the temporal filtering, due to their good compression efficiency.

3.4.1 Spectral decorrelation

Each multispectral image is composed of several bands that are correlated. The SPOT 1, 2, and 4 images exhibit some correlation among spectral bands. Instead of processing the bands separately, the interband correlation should be exploited to improve the compression. In our scheme, we propose the KLT for spectral decorrelation of multispectral images.

The KLT is applied among the spectral components taking each pixel of the scene as N-dimensional vector (N being the number of spectral bands). Let X be the column vector containing the N components for a given pixel and U be the mean vector $U = E[X]$. The covariance matrix C_x is defined as:

$$C_x = E[(X-U)(X-U)^t] \tag{3.38}$$

The KLT (T) is defined as the matrix that diagonalizes C_x in the following way:

$$C_y = TC_xT^t = \Lambda \tag{3.39}$$

C_y being the covariance of the transformed vector (Y) and Λ the diagonal matrix representing the eigenvalues of C_x. Y can then be obtained by the equation:

$$Y = T(X-U) \tag{3.40}$$

Since the transform diagonalizes the covariance matrix between spectral bands, the spectral correlation of the components is removed. The images in the transformed domain are sorted by decreasing order of importance (value of the eigenvalues). This energy

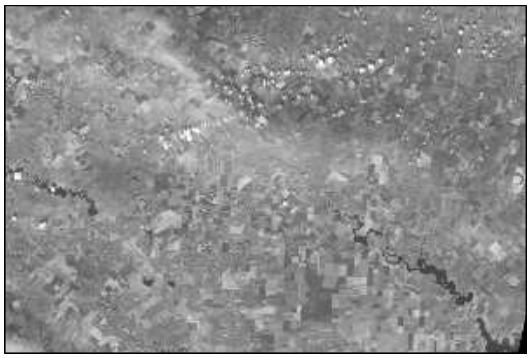


Figure 3.24: First component (highest energy) of the KLT.

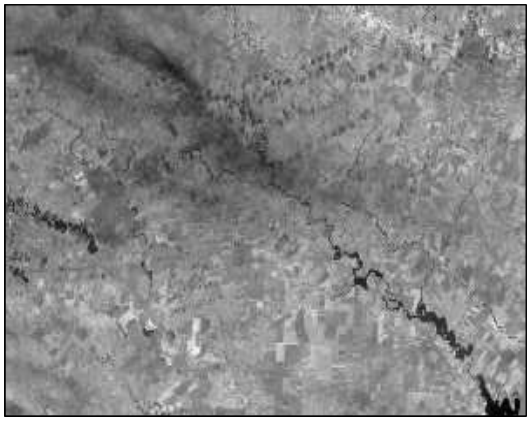


Figure 3.25: Second component of the KLT.

compaction along the spectral axis is suitable for the selection of the main spectral components for analysis as well as for image compression. Figs. 3.24, 3.25, and 3.26 present one example of components obtained after applying the KLT.

3.4.2 Temporal decorrelation

We consider a sequence of multispectral images that are supposed perfectly registered. The registration [22] is done in order to have precise and constant location for each pixel at any moment. Therefore, we can consider that no displacement between successive images remains in the sequence. In this approach, we treat the spectral bands independently, so there will be as many sequences as the number of spectral bands. Since after registration there is no displacement between successive input images, we may also consider that the sequences form a hyperspectral image [52].

In the following, the use of a $t + 2D$ wavelet decomposition is proposed for each spectral-band sequence in order to exploit both temporal and spatial redundancies as illustrated in Fig. 3.27. Due to the previous registration of the satellite sequences, the temporal transform is applied without performing motion estimation. It is already known [218] that long bidirectional filters, like the $5/3$ wavelet transform, perform better, in terms of compression efficiency, than the shorter ones (i.e. Haar filter bank). Therefore, we shall use in the following a temporal decomposition based on the $5/3$ biorthogonal

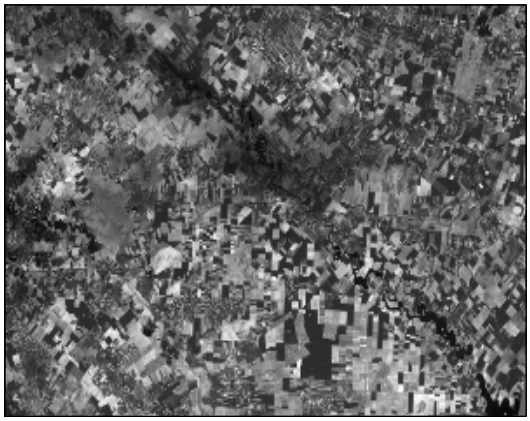


Figure 3.26: Third component of the KLT.

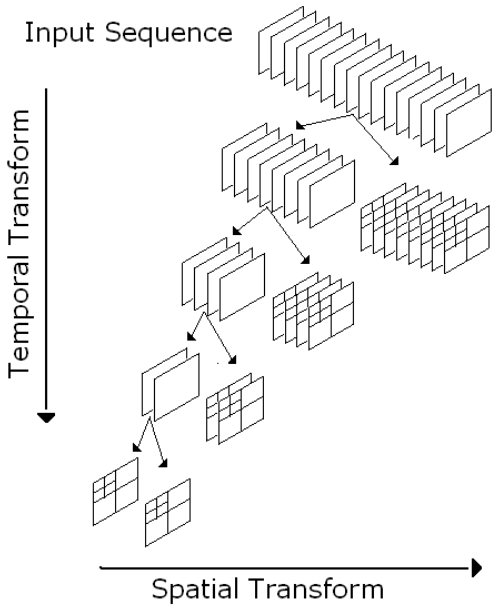


Figure 3.27: Spatio-temporal wavelet transform applied on a spectral-band sequence.

wavelets.

We denote by x_{2t-1}, x_{2t}, x_{2t+1}, x_{2t+2} the frames positioned in time at $2t-1$, $2t$, $2t+1$ and respectively $2t+2$ moments. As explained in section 2.3.1.3, in the $5/3$ filtering, the detail frame h_t is predicted from its two neighbours, i.e., x_{2t} and x_{2t+2}; in a similar way the approximation frame l_t is obtained from the frames h_{t-1}, h_t by the so-called update step. Recall that for one temporal level, the $5/3$ temporal filtering has the following form:

$$\begin{cases} h_t[n,m] = x_{2t+1}[n,m] - \dfrac{1}{2}(x_{2t}[n,m] + x_{2t+2}[n,m]) \\ l_t[n,m] = x_{2t}[n,m] + \dfrac{1}{4}(h_{t-1}[n,m] + h_t[n,m]) \end{cases} \tag{3.41}$$

Once the temporal subbands are obtained in the above described manner, they are further spatially decomposed with the $9/7$ biorthogonal filter bank (Fig. 3.27) and coded using an embedded zero-tree algorithm [210].

3.4.3 Joint spectral and temporal decorrelation

In the sections 3.4.1 and 3.4.2, separate spectral and temporal correlations have been highlighted. However, it is essential to jointly take into consideration both types of correlation. Hence, the idea is to first transform each group of three spectral images at a given temporal moment with the KLT and then the temporal decorrelation is achieved via wavelet or DCT on the sequences formed by these spectrally decorrelated components. The results of this methods are shown in Tab. 3.9.

3.4.4 Experimental results

The data to be compressed are sequences of SPOT images acquired by SPOT 1, 2 and 4. The data have been provided by CNES (Centre National d'Etudes Spatiales, France). Each image is composed of 3 spectral bands (the 4^{th} band of SPOT4 images is not treated here). The sequence has been extracted from images of size 3000×2000. We have reduced by cropping the size of images in order to consider that the signal is stationnary for transforms such as the KLT. The size of images is 300×200 and there are 35 images in the considered temporal sequence.

A radiometric correction, a sensing intercalibration, and a registration have been applied to the sequence. The sequence represents the reflectivity of a geographical location (the countryside at south-east of Bucharest) at several times irregularly sampled over one year. The reflectivity is between 0 and 1 and coded on 10 bits. Consequently, the distortion measure used for the presented results is slightly different from the usual one used in image compression, and we define it as follows:

$$PSNR = 10 \log_{10} \left(\frac{1024^2}{\text{MSE}} \right),$$

where MSE is the mean square quantization error.

We have performed the comparisons based on the compression results given by the state-of-the-art still image codec, JPEG2000 [8] and the Motion Compensated Embedded Zero-Tree Block-based (MC-EZBC) [210] video codec. Note that in the last one, even though no motion is estimated, the coding algorithm exploits the 3D nature of the decomposition, both in the bitrate allocation and during the entropy coding step.

Rate (bpp)	no transform	spectral KLT
Image 10	12.10	11.32
Image 15	10.32	9.99
Image 24	10.30	9.94
Image 34	11.75	11.32

Table 3.5: JPEG2000: comparison of lossles compression efficiency with and without component decorrelation.

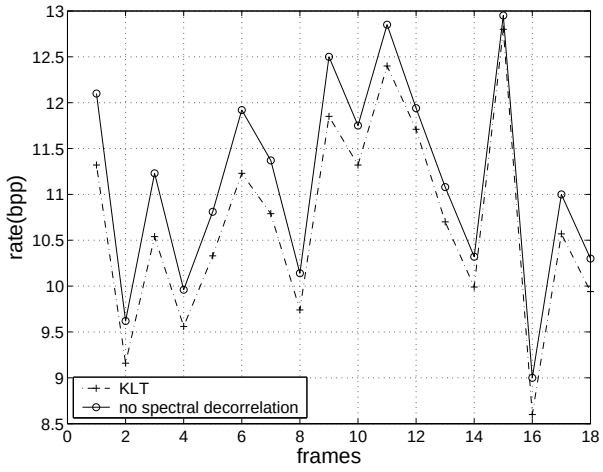


Figure 3.28: Rate comparison for spectral decorrelation transform over the multitemporal sequence.

After the spectral decorrelation, each of the three components of the multiband sequence is processed for compression using JPEG2000, based on EBCOT coding scheme [176] or using the MC-EZBC algorithm.

In Tab. 3.5, one can see that the rate of compression is 1:3 by using JPEG2000 in a lossless mode. One can observe that the KLT decorrelation decreases this rate by aproximatively 0.5 bpp. In Fig. 3.28, this gain in terms of rate is depicted for 16 images taken from the 35 frames sequence. A possible explanation of the reduced influence of the KLT is that satellite images are highly non-stationary. Indeed, in Tab. 3.6, while the size of image decreases, the gain in terms of rate increases. This is an indicator of the non-stationarity which cannot be exploited by a global KLT. The non-stationary effects are noticeable in the spectral joint histogram between the second and the third band (Fig. 3.29), in which one can see three modes of correlation. In Tab. 3.7, each band is processed separately, and no time decorrelation is applied. One can see that EZBC provides better results than EBCOT (2 bpp gain) for our images. However, JPEG2000 performs a lossless

Image size	no transform	spectral KLT	gain
100×100	11.52	10.56	0.96
500×500	10.17	9.75	0.42
1000×1000	10.14	9.74	0.40

Table 3.6: Image size influence on KLT and bitrate (bpp).

Rate (bpp)	JPEG2000	Intra Video Coding
PSNR (dB)	Lossless	> 60
Sequence Band1	3.25	2.44
Sequence Band2	3.60	2.84
Sequence Band3	5.27	4.65
Total	12.12	9.93

Table 3.7: Sequence compression without time decorrelation (JPEG2000 lossless vs. EZBC).

Rate (bpp)	JPEG2000 + WT	JPEG2000 + DCT	t+2D (EZBC)
PSNR (dB)	Lossless	> 70	> 60
Sequence Band1	3.22	3.78	2.46
Sequence Band2	3.55	4.07	2.84
Sequence Band3	4.82	4.42	4.42
Total	11.59	12.27	9.72

Table 3.8: Sequence compression with time decorrelation.

coding while EZBC does a nearly-lossless coding.

In Tab. 3.8, the $t + 2D$ wavelet decomposition is applied. The $t + 2D$ wavelet coding procedure based on the MC-EZBC codec has here one decomposition level for the $5/3$ filter, and the temporal subbands are spatially decomposed over five levels with the biorthogonal $9/7$ wavelets. In the first column, the lossless decorrelation transform proposed in JPEG2000 is applied along the temporal direction. Compared to the results without any time decorrelation, the gain in terms of rate is higher for EBCOT. Nevertheless, the DCT increases the rate unlike it was expected. Last, the results of JPEG2000 coding of temporally and spectrally decorrelated sequence are displayed in Tab.3.9. The use of temporal DCT and spectral KLT provides the best results in terms of bitrate reduction. One reason can be that the kernel of the temporal wavelet transform that was used is shorter than the support of the DCT, not allowing a good energy compaction.

3.4.5 Conclusion

In this section, we have performed a comparison between still-image and video approach for the compression of the images acquired by SPOT 1, 2, and 4. As it results from our experiments, EZBC intra-coding outperforms EBCOT in terms of rate gain in the case of nearly lossless compression. Also, it has been pointed out that lossless compression provides comparable results compared to lossy compression realized in the manner de-

Rate (bpp)	KLT	KLT and DCT	KLT and WT
PSNR (dB)	> 70	> 70	> 70
Sequence Band1	4.43	3.71	4.68
Sequence Band2	3.37	2.65	3.61
Sequence Band3	3.66	3.47	4.19
Total	11.46	9.83	12.48

Table 3.9: Sequence compression with time and spectral decorrelation.

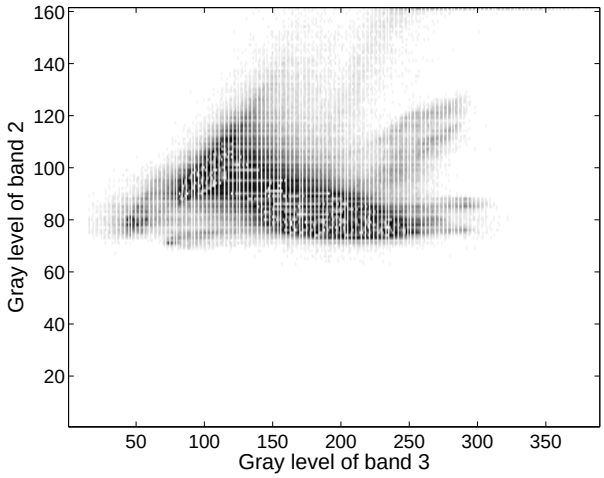


Figure 3.29: Joint histogram between the second and third components.

scribed above. In addition, it has been highlighted that temporal and spectral decorrelations can be exploited separately or jointly to improve the compression ratio. As expected, the combination of spectral and temporal decorrelation provides the best results in terms of bitrate reduction. However, the improvement achieved by these decorrelation methods does not decrease the rate significantly.

3.5 Conclusion

In this chapter, we have presented several of our contributions to the improvement of the temporal processing part of a $t+2D$ lifting-based video-coding scheme. Firstly we have proposed a modified MCTF scheme able to detect and correctly process the scene-cuts that may occur in a video sequence. Moreover, based on the proven efficiency of longer temporal filters, we have introduced a 5-band temporal lifting scheme which has shown its efficiency especially in the encoding of video-surveillance sequences. However, both contributions mentioned above are based on a linear lifting scheme. When the video sequences encounter complex motion transitions, these will be reflected in high-frequency bands, due to the inefficiency of the motion-estimation and compensation operations. So the next step of our research was to implement an adaptive temporal predictor, which has proven its efficiency in removing the motion-estimation artefacts. Further, knowing the coding efficiency of the MCTF-based schemes for video sequences, we have tested the $t+2D$ coding approach on multispectral and multitemporal image sequences, obtaining good results for nearly lossless compression.

Once the redundancy in the temporal direction of a video shot has been reduced using the above-mentioned methods, the next stage of the classical MCTF-based video codec consists of spatial decorrelation, in which the 2D redundancy will be reduced. We shall pass in the following to the presentation of our contributions to the spatial (2D) domain of the $t+2D$ video-coding framework.

Chapter 4

Spatial processing of video sequences

In a $t + 2D$ wavelet-based coding scheme, after reducing the temporal redundancy by MCTF-based techniques as presented in section 2.3.1 and chapter 3, the spatial correlation will be exploited through wavelet decompositions. Wavelet-based coding schemes have proven their efficiency for spatial coding, the most prominent example being the current still-image coder JPEG2000 [8]. The application of the wavelet transform for spatial processing of temporal subbands is most frequently based on a separable construction [125]. Lines and columns in the temporal subband are treated independently, and the basis functions are simply tensor products of the corresponding $1D$ functions. Such methods keep simplicity in design and computation but are not capable of properly capturing all the properties of a textured frame.

The reason for the inefficiency of the typical $2D$ wavelet transform resides mainly in the spatial isotropy of its construction; that is, filtering and subsampling operations are applied equally along the horizontal and vertical directions at each scale. As a result, the corresponding filters, obtained as the direct products of the $1D$ filters, are isotropic at all scales. The study regarding the frequency characteristics of the temporal subbands that we propose in section 4.1 concludes that, contrary to the approximation subbands, spectral information of temporal details is distributed almost uniformly over the frequency spectrum. As the MCTF process leads to a significant number of detail frames with respect to the total number of frames, an effective and parcimonious spatial representation of these frames is important for the overall coding efficiency. Wavelet packets are a potential solution for describing the high-pass temporal subbands; however, having a different decomposition for each frame could be very expensive from an encoding point of view.

In section 4.2, based on the above-mentioned study results, we propose to find a joint wavelet-packet representation for groups of frames, allowing a unique best-basis representation for several frames. Moreover, a computationally efficient best-basis algorithm (BBA) for biorthogonal decompositions is deduced for an entropy-based criterion.

Many anisotropic decompositions were proposed as a solution for coding textures with homogeneous energy distribution, such as bandelets [114], wedgelets [64], curvelets [63], contourlets [62] and other edge driven oriented wavelet transforms [100]. However, the implementation of these transforms usually requires oversampling and has higher complexity compared to the standard wavelet transform. In section 4.3, we propose to fully separate horizontal and vertical transforms of the classical WT. The decomposition

inherits the simplicity of processing and filter design from the standard WT, allowing perfect reconstruction and providing a highly anisotropic frequency representation.

4.1 Frequency characteristics of temporal subband frames

In this section, we make a comparative study of the spectral properties of the temporal detail and approximation subband frames. Such a study could be beneficial, as far as it helps us find a transform that can represent the temporal subbands better than the dyadic wavelets. Our study will be done in three stages: initially, we will analyze the behavior of the power-spectrum partition for temporal subband frames. Then, we will locate the maximum amount of energy inside the subbands, to finally characterize the distribution of spectral information in the various frequential subbands.

4.1.1 Estimation of power-spectrum density

We considered for our study two video sequences: Mobile and Foreman in CIF format (352×288) at 30 Hz, both decomposed over four temporal levels. Let us recall that a temporal decomposition over four levels of a sequence provides us 4 detail subband sequences (one for each temporal decomposition level) and a sequence containing the approximation subbands. Our spectral study was done then on these five sequences, in addition to the original sequence. Generally, frames belonging to the same level of temporal wavelet decomposition have close spectral properties. Thus, analyzing the spectrum of a sequence obtained on a certain level of decomposition amounts to looking at the average spectrum of all the frames belonging to this same sequence. In practice, we consider the power-spectrum densities (PSD) of a sequence by averaging the PSD of the frames belonging to that sequence in time.

Thus, for each level of temporal decomposition j, it is enough to consider the following function:

$$\ln\Big(\sum_{f_{ij}\in S_j} |FT(f_{ij})|^2\Big) \tag{4.1}$$

where:
S_j: level j temporal decomposition subband sequence.
f_{ij}: i^{th} frame in S_j sequence.
FT: spatial Fourier Transform.

For the Mobile (CIF, 30Hz) subband sequences, the PSD estimation provided us the curves presented in Fig. 4.1: One can notice that the spectral information reached its maximum, for each subband sequence, at low-pass frequencies. Also, in comparison to original and approximation subbands sequences, the amplitude decrease for the detail subbands is relatively slow. This means that the spectral information for these subbands is not localised only at low frequencies but is also present at high frequencies. Similar results were found in the study of Foreman (CIF, 30Hz) sequence (see Fig. 4.2).

4.1.2 Maximal spectral-information localization

In section 4.1.1 we have noted that spectral information, in the case of detail subband sequences, is not located only at low frequencies.

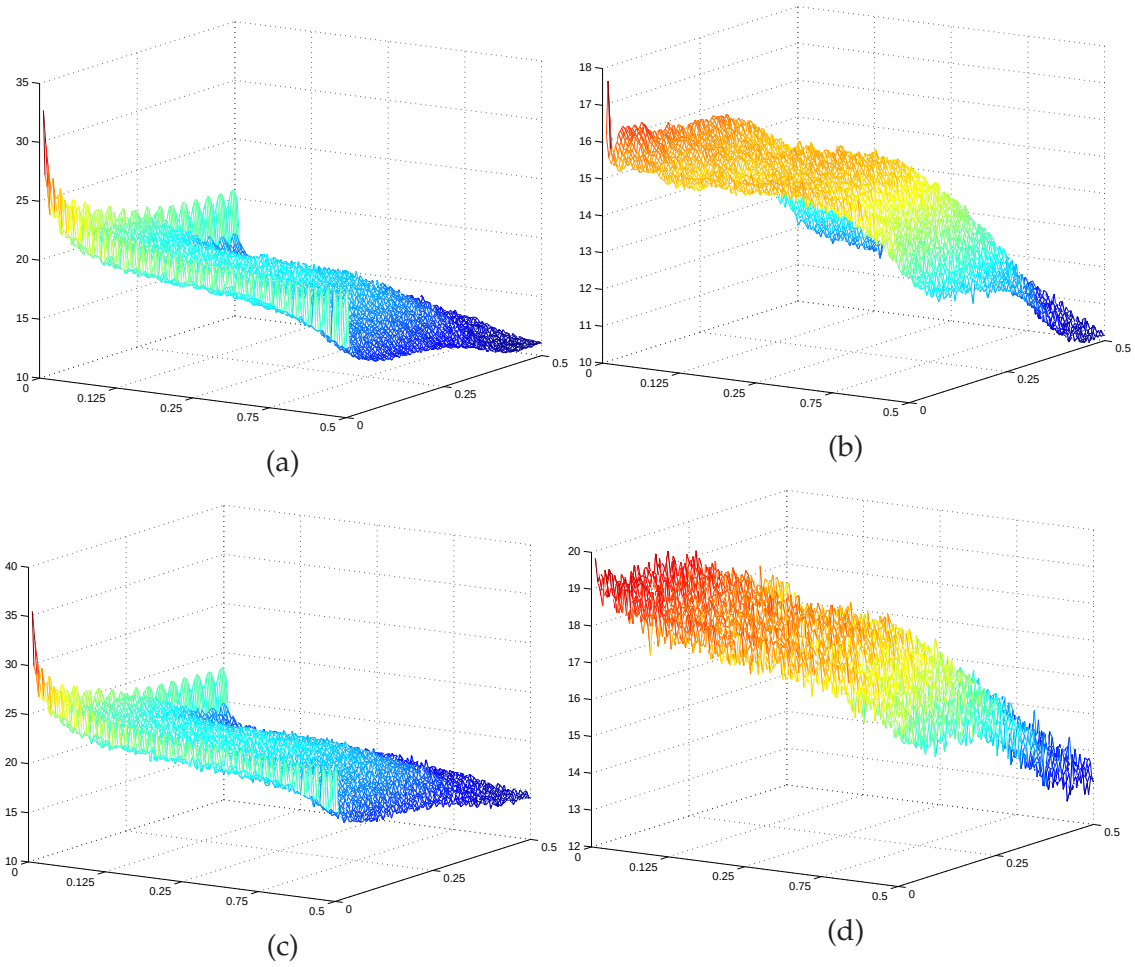


Figure 4.1: PSD estimation pour Mobile (CIF, 30Hz): original sequence (a), detail subbands for one temporal decomposition level (b), temporal approximation (c) and detail subbands (d) for 4 decomposition levels.

In the second stage of our study, we pursue locations which cover 90% of spectral energy. With this intention, we perform a zigzag scan (see Fig.(4.3)) of the positive space frequency quarter (because of the symmetries in the Fourier transform of real images) of the estimated PSD matrix for each temporal subband sequence, and then we plot the energy according to a scanning parameter l, where l has as maximum value $l_{max} = 256 * 257/2 = 32896$.

From Tab. 4.1 and Tab. 4.2 one can remark that:

- ⋆ for the temporal-approximation subband sequences as well as for the original sequence, the maximum to 90 % of energy is located in the spatial low-frequency subbands.
- ⋆ for the temporal-detail subband sequences, the maximum of energy is reached both in the spatial high- and low-frequency subbands.

For the Mobile (CIF, 30Hz) sequence, which is more textured than Foreman (CIF, 30Hz), the energy is distributed on more spectral coefficients (80 % instead of 70 %).

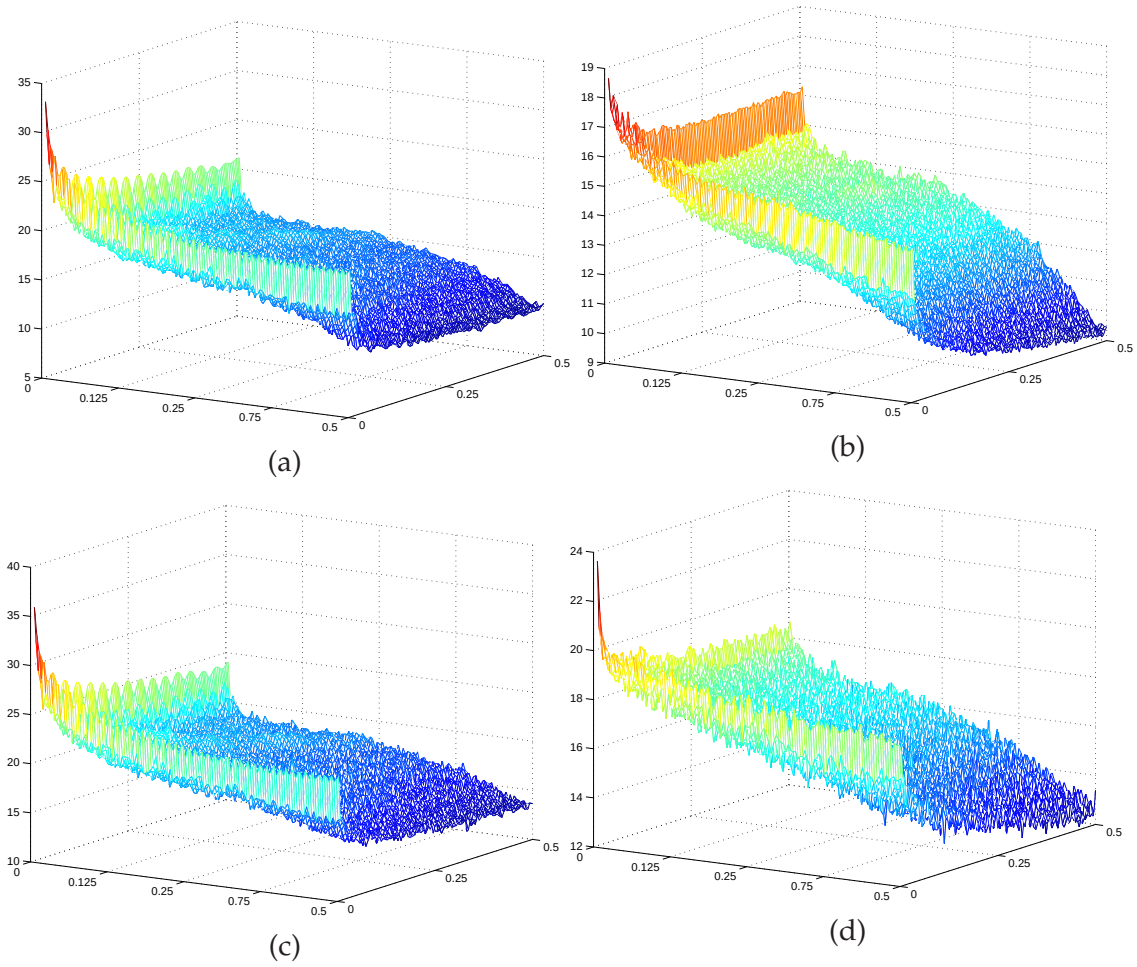


Figure 4.2: PSD logaritmic estimation pour Foreman (CIF, 30Hz) video sequence: original sequence (a), detail subbands for one temporal decomposition level (b), temporal approximation (c) and detail subbands (d) for 4 decomposition levels.

Mobile	$l_{90\%}/l_{max}$	$(i,j)_{90\%}$
Original	$3.34 * 10^{-4}$	(2, 3)
1^{st} Level Details	83.34	(79, 156)
2^{nd} Level Details	81.65	(173, 60)
3^{rd} Level Details	80.50	(86, 145)
4^{th} Level Details	81.18	(140, 92)
4^{th} Level Approximations	$3.43 * 10^{-4}$	(1, 5)

Table 4.1: Indices of maximum to 90% spectral energy covering for Mobile (CIF, 30Hz).

This illustrates quantitatively the graphs that we presented in section 4.1.1 concerning the PSD of various temporal sequences.

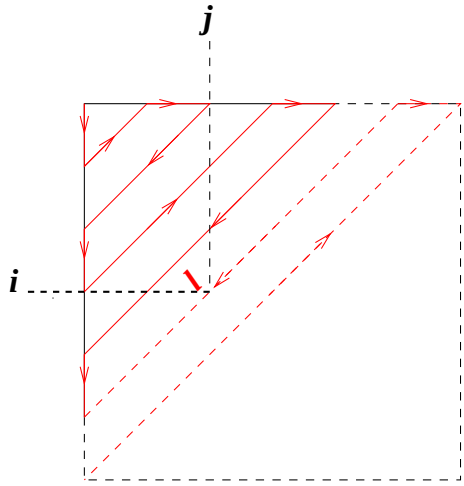


Figure 4.3: Zig-zag scanning of the estimated PSD matrix.

Foreman	$l_{90\%}/l_{max}$	$(i,j)_{90\%}$
Original	$2.73 * 10^{-4}$	(2, 3)
1^{st} Level Details	76.69	(29, 197)
2^{nd} Level Details	70.69	(182, 35)
3^{rd} Level Details	66.70	(206, 4)
4^{th} Level Details	70.51	(190, 26)
4^{th} Level Approximations	$2.43 * 10^{-4}$	(3, 2)

Table 4.2: Indices of maximum to 90% spectral energy covering for Foreman (CIF, 30Hz).

4.1.3 Distribution of spectral information

In order to better localize the spectral information, we have divided the spectrum of the temporal sequences into 4, 9, and finally 16 frequency bands, and we have calculated, each time, the quantities of energy present in the various subbands.

For Mobile (CIF, 30Hz) sequence, the distribution of spectral information over 4, 9, and respectively 16 subbands is presented in Tab. 4.3, Tab. 4.4, and Tab. 4.5 respectively.

$f_h \setminus f_v$	$[0, \frac{1}{4}[$	$[\frac{1}{4}, \frac{1}{2}]$
$[0, \frac{1}{4}[$	99.35%	0.34%
$[\frac{1}{4}, \frac{1}{2}]$	0.27%	0.05%

(a)

$f_h \setminus f_v$	$[0, \frac{1}{4}[$	$[\frac{1}{4}, \frac{1}{2}]$
$[0, \frac{1}{4}[$	48.29%	24.21%
$[\frac{1}{4}, \frac{1}{2}]$	19.58%	7.92%

(b)

$f_h \setminus f_v$	$[0, \frac{1}{4}[$	$[\frac{1}{4}, \frac{1}{2}]$
$[0, \frac{1}{4}[$	52.91%	19.58%
$[\frac{1}{4}, \frac{1}{2}]$	20.85%	6.67%

(c)

$f_h \setminus f_v$	$[0, \frac{1}{4}[$	$[\frac{1}{4}, \frac{1}{2}]$
$[0, \frac{1}{4}[$	99.29%	0.38%
$[\frac{1}{4}, \frac{1}{2}]$	0.28%	0.05%

(d)

Table 4.3: Average spectral-energy partition over 4 subbands for Mobile (CIF, 30Hz) sequence: original sequence (a), detail subbands for one temporal decomposition level (b), temporal detail (c) and approximation subbands (d) for 4 decomposition levels.

$f_h \setminus f_v$	$[0, \frac{1}{6}[$	$[\frac{1}{6}, \frac{1}{3}[$	$[\frac{1}{3}, \frac{1}{2}]$
$[0, \frac{1}{6}[$	98.56%	0.45%	0.09%
$[\frac{1}{6}, \frac{1}{3}[$	0.58%	0.21%	0.03%
$[\frac{1}{3}, \frac{1}{2}]$	0.06%	0.02%	0%

(a)

$f_h \setminus f_v$	$[0, \frac{1}{6}[$	$[\frac{1}{6}, \frac{1}{3}[$	$[\frac{1}{3}, \frac{1}{2}]$
$[0, \frac{1}{6}[$	23.29%	17.59%	7.72%
$[\frac{1}{6}, \frac{1}{3}[$	20.03%	17.01%	5.33%
$[\frac{1}{3}, \frac{1}{2}]$	4.68%	3.45%	0.9%

(b)

$f_h \setminus f_v$	$[0, \frac{1}{6}[$	$[\frac{1}{6}, \frac{1}{3}[$	$[\frac{1}{3}, \frac{1}{2}]$
$[0, \frac{1}{6}[$	28.72%	15.07%	6.76%
$[\frac{1}{6}, \frac{1}{3}[$	20.73%	12.70%	4.54%
$[\frac{1}{3}, \frac{1}{2}]$	6.84%	3.64%	1%

(c)

$f_h \setminus f_v$	$[0, \frac{1}{6}[$	$[\frac{1}{6}, \frac{1}{3}[$	$[\frac{1}{3}, \frac{1}{2}]$
$[0, \frac{1}{6}[$	98.48%	0.46%	0.12%
$[\frac{1}{6}, \frac{1}{3}[$	0.53%	0.22%	0.04%
$[\frac{1}{3}, \frac{1}{2}]$	0.06%	0.02%	0%

(d)

Table 4.4: Average spectral energy-partition over 9 subbands for Mobile (CIF, 30Hz) sequence: original sequence (a), detail subbands for one temporal decomposition level (b), temporal detail (c) and approximation subbands (d) for 4 decomposition levels.

$f_h \setminus f_v$	$[0, \frac{1}{8}[$	$[\frac{1}{8}, \frac{1}{4}[$	$[\frac{1}{4}, \frac{3}{8}[$	$[\frac{3}{8}, \frac{1}{2}]$
$[0, \frac{1}{8}[$	98.07%	0.40%	0.18%	0.04%
$[\frac{1}{8}, \frac{1}{4}[$	0.65%	0.22%	0.11%	0.01%
$[\frac{1}{4}, \frac{3}{8}[$	0.17%	0.07%	0.04%	0.01%
$[\frac{3}{8}, \frac{1}{2}]$	0.03%	0%	0%	0%

(a)

$f_h \setminus f_v$	$[0, \frac{1}{8}[$	$[\frac{1}{8}, \frac{1}{4}[$	$[\frac{1}{4}, \frac{3}{8}[$	$[\frac{3}{8}, \frac{1}{2}]$
$[0, \frac{1}{8}[$	14.42%	9.76%	9.09%	3.36%
$[\frac{1}{8}, \frac{1}{4}[$	13.18%	10.93%	9.26%	2.50%
$[\frac{1}{4}, \frac{3}{8}[$	9.59%	7.28%	5.71%	1.25%
$[\frac{3}{8}, \frac{1}{2}]$	1.56%	1.16%	0.76%	0.20%

(b)

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

(c)

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

(d)

Table 4.5: Average spectral energy-partition over 16 subbands for Mobile (CIF, 30Hz) sequence: original sequence (a), detail subbands for one temporal decomposition level (b), temporal detail (c) and approximation subbands (d) for 4 decomposition levels.

The results confirm those presented in the previous two studies. One can remark from Tab. 4.3- 4.5 ((b),(c)) that the spectral information is far from being localised only at low frequencies. Indeed, in comparison with the situation for temporal approximation (d) and original (a) sequences where more than 98% of energy is distributed in low-pass

bands, the quantity of energy captured at these frequencies does not exceed 30% of total energy for the temporal detail subband sequences.

Similar results were obtained for Foreman (CIF, 30Hz) sequence, where the average spectral energy repartition over 4, 9, and 16 subbands is presented in Tab. 4.6, Tab. 4.7, and Tab. 4.8 respectively.

$f_h \setminus f_v$	$[0, \frac{1}{4}[$	$[\frac{1}{4}, \frac{1}{2}]$
$[0, \frac{1}{4}[$	99.84%	0.03%
$[\frac{1}{4}, \frac{1}{2}]$	0.12%	0%

(a)

$f_h \setminus f_v$	$[0, \frac{1}{4}[$	$[\frac{1}{4}, \frac{1}{2}]$
$[0, \frac{1}{4}[$	60.64%	6.85%
$[\frac{1}{4}, \frac{1}{2}]$	27.94%	4.57%

(b)

$f_h \setminus f_v$	$[0, \frac{1}{4}[$	$[\frac{1}{4}, \frac{1}{2}]$
$[0, \frac{1}{4}[$	67.79%	8.49%
$[\frac{1}{4}, \frac{1}{2}]$	20.82%	2.90%

(c)

$f_h \setminus f_v$	$[0, \frac{1}{4}[$	$[\frac{1}{4}, \frac{1}{2}]$
$[0, \frac{1}{4}[$	99.84%	0.03%
$[\frac{1}{4}, \frac{1}{2}]$	0.12%	0%

(d)

Table 4.6: Average spectral energy-partition over 4 subbands for Foreman (CIF, 30Hz) sequence: original sequence (a), detail subbands for one temporal decomposition level (b), temporal detail (c) and approximation subbands (d) for 4 decomposition levels.

$f_h \setminus f_v$	$[0, \frac{1}{6}[$	$[\frac{1}{6}, \frac{1}{3}[$	$[\frac{1}{3}, \frac{1}{2}]$
$[0, \frac{1}{6}[$	99.71%	0.06%	0.01%
$[\frac{1}{6}, \frac{1}{3}[$	0.14%	0.01%	0%
$[\frac{1}{3}, \frac{1}{2}]$	0.06%	0.01%	0%

(a)

$f_h \setminus f_v$	$[0, \frac{1}{6}[$	$[\frac{1}{6}, \frac{1}{3}[$	$[\frac{1}{3}, \frac{1}{2}]$
$[0, \frac{1}{6}[$	46.25%	7.91%	2.25%
$[\frac{1}{6}, \frac{1}{3}[$	14.91%	4.79%	1.07%
$[\frac{1}{3}, \frac{1}{2}]$	15.69%	6.11%	1.03%

(b)

$f_h \setminus f_v$	$[0, \frac{1}{6}[$	$[\frac{1}{6}, \frac{1}{3}[$	$[\frac{1}{3}, \frac{1}{2}]$
$[0, \frac{1}{6}[$	50.50%	10.38%	3.38%
$[\frac{1}{6}, \frac{1}{3}[$	15.35%	4.03%	0.94%
$[\frac{1}{3}, \frac{1}{2}]$	11.19%	3.55%	0.67%

(c)

$f_h \setminus f_v$	$[0, \frac{1}{6}[$	$[\frac{1}{6}, \frac{1}{3}[$	$[\frac{1}{3}, \frac{1}{2}]$
$[0, \frac{1}{6}[$	99.71%	0.05%	0.01%
$[\frac{1}{6}, \frac{1}{3}[$	0.14%	0.01%	0%
$[\frac{1}{3}, \frac{1}{2}]$	0.07%	0.01%	0%

(d)

Table 4.7: Average spectral energy-partition over 9 subbands for Foreman (CIF, 30Hz) sequence: original sequence (a), detail subbands for one temporal decomposition level (b), temporal detail (c) and approximation subbands (d) for 4 decomposition levels.

4.1.4 Conclusion

In this section, we have studied the spectral properties of the subbands obtained by the wavelet temporal decomposition. We noticed that, contrary to the approximation subbands, spectral information of temporal details is distributed almost uniformly over the subbands, thus, far from being localized in the low frequencies. These differences suggest that the classical dyadic wavelet, which could be powerful for the spatial encoding of still images or temporal approximation subbands, would not be the best decorrelation scheme for the temporal detail subbands.

These observations motivate us to investigate the use of wavelet packets, whose frequency - selectivity properties are more suited to decomposition of the detail frames. We propose thus to pass in the following to the presentation of a wavelet-packet-based spatial decomposition.

$f_h \setminus f_v$	$[0, \frac{1}{8}[$	$[\frac{1}{8}, \frac{1}{4}[$	$[\frac{1}{4}, \frac{3}{8}[$	$[\frac{3}{8}, \frac{1}{2}]$
$[0, \frac{1}{8}[$	99.58%	0.08%	0.02%	0.01%
$[\frac{1}{8}, \frac{1}{4}[$	0.18%	0.01%	0%	0%
$[\frac{1}{4}, \frac{3}{8}[$	0.07%	0.01%	0%	0%
$[\frac{3}{8}, \frac{1}{2}]$	0.04%	0%	0%	0%

(a)

$f_h \setminus f_v$	$[0, \frac{1}{8}[$	$[\frac{1}{8}, \frac{1}{4}[$	$[\frac{1}{4}, \frac{3}{8}[$	$[\frac{3}{8}, \frac{1}{2}]$
$[0, \frac{1}{8}[$	38.07%	6.98%	3.33%	1.11%
$[\frac{1}{8}, \frac{1}{4}[$	11.83%	3.76%	1.98%	0.42%
$[\frac{1}{4}, \frac{3}{8}[$	9%	3.90%	1.81%	0.34%
$[\frac{3}{8}, \frac{1}{2}]$	10.37%	4.67%	2.07%	0.33%

(b)

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

(c)

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

(d)

Table 4.8: Average spectral energy-partition over 16 subbands for Foreman (CIF, 30Hz) sequence: original sequence (a), detail subbands for one temporal decomposition level (b), temporal detail (c) and approximation subbands (d) for 4 decomposition levels.

4.2 Joint wavelet packets for video coding

As mentioned in the introduction of this chapter, a weakness in the existing MCTF video codecs is related to the way the spatial filtering is done. Most of the coding schemes do not differentiate between the characteristics of the temporal approximation and detail frames and use for both of them a dyadic wavelet decomposition. It has been shown that the 9/7 filters give very good results in the case of still-image coding [19, 8], and it appears obvious to use them for the spatial decomposition of the approximation frames. Due to their significant amount of intermediate and high frequencies, the detail frames are, however, not suited to be decomposed with wavelets. Because, in the temporal subband coding paradigm, the proportion of detail frames with respect to the total number of frames is significant, an effective and parcimonious spatial representation of these frames is important.

Wavelet-packet subband structures [207] generalize the dyadic decomposition used by most classical wavelet-based coding schemes by iterating the decompositions on the high-pass subbands as well. Especially for images with highly textured content, the rate-distortion performance of adaptively generated wavelet-packet subband structures is superior to classical wavelet one.

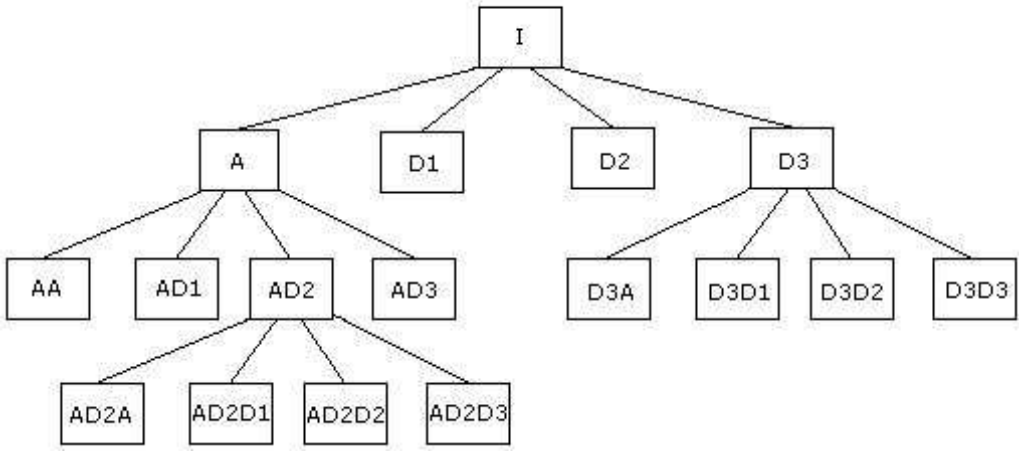


Figure 4.4: An example of wavelet packet decomposition tree.

Best-basis algorithms [49] are the most computationally efficient techniques to find wavelet-packet subband structures for a given image. This can be done by optimizing rate-independent information cost functions operating in the transform domain which provides suboptimal rate-distortion performance only. Techniques involving rate- distortion -based optimization have also been introduced [156, 162], often at a higher computational cost.

The idea of using 3D wavelet packets for video compression has been developed in some previous work [95, 179, 110]. Also, a hybrid video coder using the wavelet packet transform and motion compensation has been proposed by Cheng, Li and Kuo [42], where wavelet packets are used to represent the displaced frame differences (DFD), whose features are close to those of the detail frames in a temporal subband decomposition.

In this section, we will introduce *joint* wavelet packets for groups of frames, providing a unique best-basis representation for several frames, and not one basis per frame, as is classically done. Two main advantages are expected from this joint representation. On the one hand, bitrate is spared, since a single-tree description is sent instead of 31 per group of frames (GOP) - when a GOP contains, for example, 32 frames. On the other hand, this common description can characterize the spatio-temporal features of the given video GOP and this way can be exploited as a valuable feature for video classification and video-database searching.

Moreover, we provide insight into the modifications necessary in the best-basis algorithm (BBA) in order to cope with biorthogonal decompositions. A computationally efficient algorithm is deduced for an entropy-based criterion. This section work has been subject to publication into the SPIE-VCIP'05 [191] conference proceedings.

4.2.1 Wavelet packets

In $t+2D/2D+t$ subband video coding [104, 117, 93], wavelet decompositions are used both in the temporal and spatial dimensions, being spatio-temporal operators, as we have named them in section 2.3.1.3. Wavelet packets are a generalized family of multiresolution bases that include wavelets (see section 1.2.4). The idea is to choose the best-basis

by some criterion such as entropy [49], rate-distortion [156, 127, 167, 158], or other measure adapted to the application at hand. This method should provide results at least as good as, or better than, the dyadic wavelet basis for the chosen cost function. In this case, both low-frequency (approximations) and high-frequency (details) components are decomposed. In Fig. 4.4 an example of such a decomposition is shown. This type of analysis allows an image to be decomposed in 4^J different manners, where J is the maximal analysis depth. In order to define which is the best decomposition from the point of view of the least cost, an entropy criterion is frequently used [49].

4.2.1.1 Orthogonal wavelet packets. Best basis algorithm

Orthogonal wavelets generate orthonormal bases of $L^2(\mathcal{R}^2)$ as shown in section 1.2.1, and separable compactly supported wavelets are commonly used in image coding. There are several families of commonly used wavelets, such as Daubechies [58, 57] and Coifman [49]. In these decompositions, the length of the filter is related to the smoothness and regularity of the wavelet. For building wavelet packets, the same families of functions can be used, but the tree structure will be adapted to the input content in order to maximize an energy- or entropy-compaction criterion. The basis selection in each node of the tree has therefore to be done according to this criterion. For this, consider the projection of a signal $s(t)$ into two orthonormal bases $\mathcal{B}$ and $\mathcal{B}'$. If, for any real concave function $\mathcal{J}$ defined on $\mathbb{R}_+$, we have:

$$\mathcal{Q}(s,\mathcal{B}) = \sum_{k=1}^{\mathcal{K}} \mathcal{J}(|s_k|^2) \leq \sum_{k=1}^{\mathcal{K}} \mathcal{J}(|s'_k|^2) = \mathcal{Q}(s,\mathcal{B}') \tag{4.2}$$

where $\mathcal{Q}$ is the chosen cost function and s_k, s'_k the coefficients into $\mathcal{B}, \mathcal{B}'$ respectively, $k \in \{1 \ldots \mathcal{K}\}$, we say that the best decomposition of the signal is $\mathcal{B}$. One cost function or criterion usually used in the selection of the best basis is the entropy, given by $\mathcal{J}(u) = -u\ln(u)$. For a decomposition of a discrete signal in one of these bases, the entropy (also called Wickerhauser entropy) is:

$$\mathcal{E}(s,\mathcal{B}) = -\sum_{k=1}^{\mathcal{K}} \frac{s_k^2}{\sum_{k=1}^{\mathcal{K}} s_k^2} \ln\left(\frac{s_k^2}{\sum_{k=1}^{\mathcal{K}} s_k^2}\right) \tag{4.3}$$

A bottom-up pruning algorithm is used for the selection of the best orthogonal decomposition [49]. In the case of dyadic decomposition, on level j, we compare the cost associated to the best basis of the current node, $\mathcal{B}_{j,m}$, with that of its successors, $\mathcal{Q}(s,\mathcal{B}^*_{j+1,2m})$ and $\mathcal{Q}(s,\mathcal{B}^*_{j+1,2m+1})$. As the considered cost function is additive, if

$$\mathcal{Q}(s,\mathcal{B}_{j,m}) \leq \mathcal{Q}(s,\mathcal{B}^*_{j+1,2m}) + \mathcal{Q}(s,\mathcal{B}^*_{j+1,2m+1}), \tag{4.4}$$

we can conclude that the best basis for the node (j,m) is $\mathcal{B}^*_{j,m} = \mathcal{B}_{j,m}$, so the nodes $(j+1,2m)$ and $(j+1,2m+1)$ are pruned from the decomposition tree. Note that they are kept in this tree if the opposite relation holds in Eq. (4.4).

4.2.1.2 Biorthogonal wavelet packets. Modified best-basis algorithm

One disadvantage of orthogonal wavelets is their asymmetry, which is related to the use of nonlinear phase filters. When the orthogonality or compact-support restrictions are

relaxed, symmetry can be obtained, as is the case with biorthogonal wavelets discussed in section 1.2.2. In this case, the quadrature mirror filters used to calculate the discrete wavelet transform (DWT) are not an orthogonal pair, although they are orthogonal to another pair used to calculate the inverse DWT. In this manner, perfect reconstruction is preserved. Different types of biorthogonal filters have been developed [174] and have shown [201] superior performance compared to orthonormal decompositions in still-image coding. The use of biorthogonal pairs in JPEG2000 [8], for both lossless and lossy coding, also represents a strong motivation for developing a best-basis algorithm for biorthogonal wavelet-packet bases.

In the case of biorthogonal decompositions, we go back to Eq. (4.4), which can be written, in the most general case, as:

$$\mathcal{Q}(s, \mathcal{B}_{j,m}) \leq \mathcal{Q}(s, \mathcal{B}^*_{j+1,2m} \cup \mathcal{B}^*_{j+1,2m+1}) \tag{4.5}$$

This relation amounts to comparing the criterion in the father node with the criterion computed on the union of bases of its sons. Once the decision is taken, the algorithm iterates with the comparison of the criterion in the upper node with the criterion corresponding to all the leaves remaining under it in the bottom-up pruning process:

$$\mathcal{Q}(s, \mathcal{B}_{j,m}) \leq \mathcal{Q}(s, \bigcup_{j'>j,m'} \mathcal{B}^*_{j',m'}) \tag{4.6}$$

where the index m' corresponds to all the leaves under the node (j, m) kept in the tree at this step.

This means that, at each step of the algorithm, one has to recompute the criterion corresponding to all the leaves considered in Eq. (4.6) in the union of bases of these leaves. This is a computationally intensive task. Hopefuly, in the case of the entropy criterion (see Eq. (4.3)), one can take advantage of the entropy values already computed in the previous steps in order to obtain a recursive formula for the criterion corresponding to the union of leaves:

$$\mathcal{E}(s, \bigcup_{j'>j,m'} \mathcal{B}^*_{j',m'}) = \frac{\sum_{j',m'} \mathcal{E}(s, \mathcal{B}^*_{j',m'})}{\sum_{j',m'} \mathcal{N}_{j',m'}} + \ln(\sum_{j',m'} \mathcal{N}_{j',m'}) \tag{4.7}$$

where $\mathcal{N}_{j',m'}$ is the energy of the coefficients in the basis $\mathcal{B}^*_{j',m'}$. Note that, for the sake of simplicity, we wrote the above relations in the case of a binary tree (corresponding to the decomposition of a one-dimensional signal), but they can be extended to quadtrees (for image decomposition) in an obvious manner.

4.2.2 Joint Wavelet Packets

In video coding, wavelet-packet decompositions can be successfully applied on the temporal detail frames, which are the most appropriate from the frequency-partition point of view as shown by the results in section 4.1. In the following, we present an approach for finding the best basis for a group of frames, as well as two possibilities for its application to motion-compensated temporal filtering (MCTF) video coding.

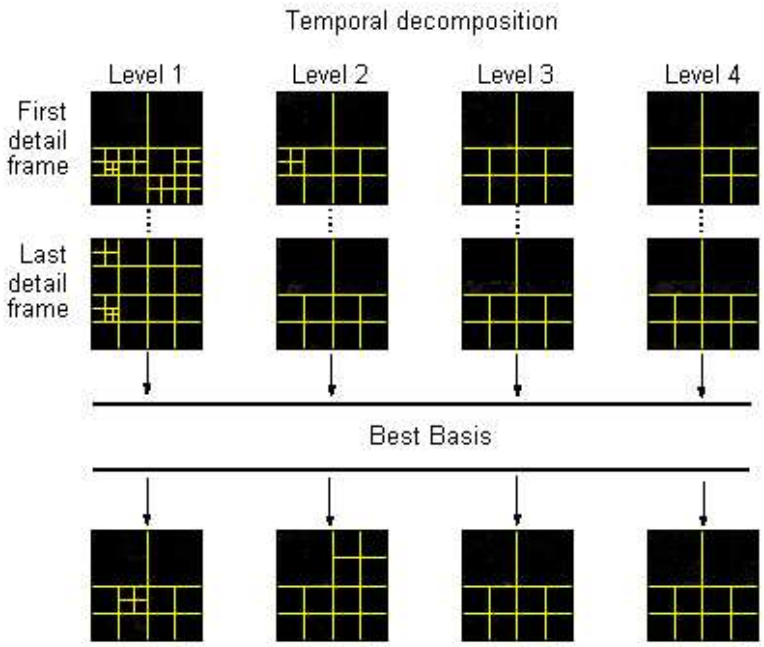


Figure 4.5: Joint wavelet-packet decomposition per temporal subband.

4.2.2.1 Joint Best Basis Algorithm

In section 4.2.1.1, we showed how the best decomposition tree can be found for a single image in the case of orthogonal wavelets. Now, we propose to extend the presented algorithm in order to find the best basis for a group of frames. For simplicity, we detail the algorithm for orthonormal bases, but an obvious extension to biorthogonal decompositions can be done by using the results presented in section 4.2.1.2.

As in the case of still images, the Wickerhauser entropy criterion can be used for the selection of the wavelet-packet decomposition of a group of frames, due to its additivity property and its ability to represent the sparsity of the signal. For a group of frames $\mathcal{S}_j$, we need to compute the entropy of the representation in the union of bases of the GOP. In the orthonormal case, the associated entropy can simply be found by:

$$\mathcal{E}(\mathcal{S}_j) = \sum_{i=1}^{n_j} \mathcal{E}(f_{i,j}) \tag{4.8}$$

where $f_{i,j}$, $i \in \{1 \dots n_j\}$ are the n_j frames in the GOP. In the biorthogonal case, the formula in Eq. (4.7) can be used. Once we obtain the entropy corresponding to $\mathcal{S}_j$, we can apply the best-basis algorithm explained previously in the orthonormal or biorthogonal case.

In the following, we present two possibilities for using the wavelet packets for the spatial decomposition of the temporal detail frames. The first approach is to use a different best-basis decomposition for all the frames belonging to a given level of temporal decomposition (as illustrated in Fig. 4.5). This method issues from the assumption that the frames belonging to a specific decomposition level present, more or less, the same frequency characteristics. The second approach is to have one wavelet-packet basis for

each encoding unit; i.e., a unique basis for all detail frames in a GOP (as it can be seen in Fig. 4.6).

For each frame $f_{i,j}$ of the sequence $\mathcal{S}_j$, the relative representation error when decomposing the frame in the joint basis, normalised with respect to the optimal entropy obtained if the frame is represented in its own best basis, is given by:

$$RelativeRepresentationError(f_{i,j}) = \frac{JointEntropy(f_{i,j}) - OptimalEntropy(f_{i,j})}{OptimalEntropy(f_{i,j})} \tag{4.9}$$

An average relative error over the jointly encoded frames is:

$$AverageRepresentationError(\mathcal{S}_j) = \frac{\sum_{f_{i,j} \in \mathcal{S}_j} RelativeRepresentationError(f_{i,j})}{n_j} \tag{4.10}$$

Moreover, it is interesting to have also the maximal representation error, found as:

$$MaximalRepresentationError(\mathcal{S}_j) = \max_{f_{i,j} \in \mathcal{S}_j} (RelativeRepresentationError(f_{i,j})) \tag{4.11}$$

In Tabs. 4.9-4.11, we present the mentioned representation errors, as well as the average joint entropies for both approaches (one basis per temporal level or one per GOP), for the Foreman (CIF, 30Hz), Mobile (CIF, 30Hz) and Harbour (4CIF, 60Hz) sequences.

As it can be seen, the error representations are relatively small, especially when a joint representation per temporal level is chosen. We can conclude that using a joint wavelet

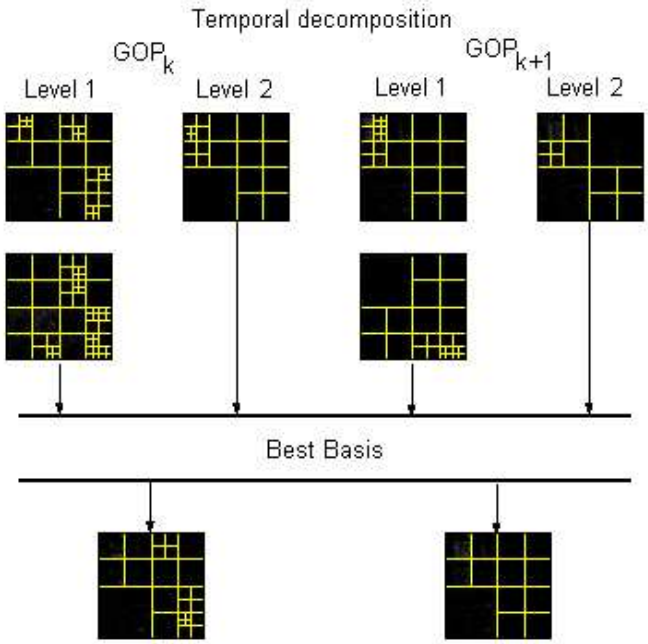


Figure 4.6: Joint wavelet-packet decomposition per GOP.

Foreman (CIF, 30Hz)	Level 1	Level 2	Level 3	Level 4	Level 5	GOP
Average Joint Entropy	108.33	66.05	55.99	67.07	47.48	87.02
Average Representation Error	4.2%	2.1%	0.7%	0.3%	0.1%	6.1%
Maximal Representation Error	42.1%	16.4%	9.3%	3.5%	2.4%	80.1%

Table 4.9: Average joint entropies and representation errors for the sequence Foreman (CIF, 30Hz).

Mobile (CIF, 30Hz)	Level 1	Level 2	Level 3	Level 4	Level 5	GOP
Average Joint Entropy	111.60	99.70	73.59	58.88	52.49	95.29
Average Representation Error	1.4%	3.4%	4.4%	1.9%	1.2%	5.7%
Maximal Representation Error	14.7%	28.5%	28.4%	15.7%	13.8%	73%

Table 4.10: Average joint entropies and representation errors for the sequence Mobile (CIF, 30Hz).

Harbour (4CIF, 60Hz)	Level 1	Level 2	Level 3	Level 4	Level 5	GOP
Average Joint Entropy	232.67	140.35	117.25	121.64	115.91	183.21
Average Representation Error	2.4%	0.9%	1.3%	0.5%	0.9%	4.8%
Maximal Representation Error	64.9%	12.2%	15.9%	8.2%	11.7%	80.4%

Table 4.11: Average joint entropies and representation errors for the sequence Harbour (4CIF, 60Hz).

packet decomposition for all the detail frames belonging to a given temporal level, or to a group of frames, does not change dramatically compared to the best basis of the individual frame.

4.2.3 Experimental results

For simulations, we have considered three representative video sequences: Foreman (CIF, 30 Hz), Mobile (CIF, 30 Hz) and Harbour (4CIF, 60 Hz), which have been selected for their different motion and texture characteristics. The tests have been made in the framework of the MSRA [117] video codec. For our simulations, we have used only the $t + 2D$ coding approach. Motion estimation is block-based and the motion-vector fields have been estimated with $1/4$-pixel accuracy. All video sequences used in our tests have been decomposed with a 5-level $5/3$ temporal decomposition which supports block-prediction modes. Temporal approximation and detail subband wavelet coefficients have been encoded using the 3D-ESCOT [117] algorithm which supports both wavelet and wavelet-packet spatial decompositions.

In our experiments, the temporal approximation subbands have been spatially decomposed over 5 levels with the biorthogonal $9/7$ wavelet. For the spatial decomposition of the temporal detail frames, we have compared the following representations:

- ⋆ dyadic biorthogonal wavelets with $9/7$ filter banks (Wavelet).
- ⋆ orthonormal wavelet packets using symmlet functions (Ortho.WP).
- ⋆ biorthogonal wavelet packets using $9/7$ filter banks (Biorthog.WP).

For the last two cases, in a first approach, we have run the simulations having a joint wavelet-packet decomposition for each of the 5 levels of temporal filtering. In a second

YSNR (dB) for Foreman (CIF, 30Hz) sequence					
Bitrate (kbs)	128	160	192	224	256
5-lev. dyadic Wavelet	31.331	32.325	33.098	33.784	34.337
Ortho.WP/TS	31.514	32.499	33.283	33.991	34.494
Ortho.WP/GOP	**31.578**	**32.667**	**34.342**	**34.101**	**34.548**
Biorthog.WP/TS	31.548	32.584	34.296	34.106	34.557
Biorthog.WP/GOP	**31.591**	**32.689**	**34.462**	**34.179**	**34.635**

Table 4.12: Rate-distortion comparison for the sequence Foreman (CIF, 30Hz). *Ortho/Biorthog* stands for an orthonormal/biorthogonal decomposition, respectively and *TS* for temporal subband.

YSNR (dB) for Mobile (CIF, 30Hz) sequence					
Bitrate (kbs)	192	224	256	320	384
5-lev. dyadic Wavelet	24.968	25.678	26.285	27.308	28.096
Ortho.WP/TS	25.317	26.063	26.728	27.698	28.473
Ortho.WP/GOP	**25.321**	**26.114**	**26.764**	**27.704**	**28.484**
Biorthog.WP/TS	25.728	26.311	27.242	28.117	29.102
Biorthog.WP/GOP	**25.891**	**26.532**	**27.521**	**28.268**	**29.271**

Table 4.13: Rate-distortion comparison for the sequence Mobile (CIF, 30Hz). *Ortho/Biorthog* stands for an orthonormal/biorthogonal decomposition, respectively and *TS* for temporal subband.

approach, we have constructed a wavelet-packet basis for all the detail frames in a GOP. In Tabs. 4.12-4.14, we compare the YPSNR results for the three sequences. The bitrate test-points for the sequences used in our simulations correspond to those defined for the MPEG standardization [215].

The obtained results confirm the remarks make in section 4.1. Due to their significant amount of medium and high frequencies, the most appropriate representation of the detail frames is not based on dyadic wavelets.

It can be easily noticed that in all the cases the decomposition of the temporal detail frames with wavelet packets performs better, achieving a gain between 0.1 and 0.9 dB over the classical dyadic filtering. Also, we can observe that the use of a biorthogonal wavelet-packet basis slightly improves the results as compared to the orthonormal basis

YSNR (dB) for Harbour (4CIF, 60Hz) sequence					
Bitrate (kbs)	1536	1780	2048	2560	3072
5-lev. dyadic Wavelet	30.834	31.228	31.726	32.489	33.035
Ortho.WP/TS	31.068	31.474	31.968	32.718	33.296
Ortho.WP/GOP	**31.111**	**31.476**	**31.969**	**32.733**	**33.312**
Biorthog.WP/TS	31.220	31.732	32.191	32.803	33.381
Biorthog.WP/GOP	**31.303**	**31.941**	**32.235**	**32.836**	**33.448**

Table 4.14: Rate-distortion comparison for the sequence Harbour (4CIF, 60Hz). *Ortho/Biorthog* stands for an orthonormal/biorthogonal decomposition, respectively and *TS* for temporal subband.

(a) (b)

Figure 4.7: Reconstructed frame from Mobile (CIF, 30Hz) sequence at 384 kbs: (a) 5-level 9/7 spatial decomposition, (b) 5-level joint 9/7 spatial decomposition at GOP level.

(we obtain a average gain around 0.1 dB over the results obtained with an orthonormal basis). The small differences between the results obtained with the wavelet-packet basis applied on each temporal subband and on each group of frames can be explained by the additional rate necessary in the former case needed to describe the decomposition tree at each level.

4.2.4 Conclusion

In this section, we have presented a method for building a joint wavelet-packet representation for groups of frames. The best-basis selection algorithm was adapted to this goal, and the method was illustrated by simulation results in the framework of motion-compensated temporal filtering coding. Best results were obtained when a single wavelet-packet basis was considered for coding all the detail frames in a GOP. We also highlighted the algorithmic modifications for the selection of the best basis when biorthogonal bases are used.

However, finding the best wavelet-packet representation, even for a single basis per group of frames, can be a computationally expensive task. This remark suggests a new direction: finding a simpler decomposition in line the observations presented in section 4.1. The fully separable wavelet and wavelet packet-based spatial decompositions that will be presented in the next section preserve the simplicity of the classical dyadic wavelet transform and provide a sparse representation of the discontinuities in temporal high-pass subband frames.

4.3 Fully separable wavelets and wavelet packets

As mentioned in section 4.1, the dyadic spatial wavelet transform (WT) may not be the most appropriate for exploiting the spatial redundancy of the detail subbands, since very often the power spectral density of these frames is not as concentrated at low frequencies as in the case of natural images.

Wavelet-packet [156, 191] transforms (WPT) may represent a solution for the decorrelation of such textures as they generalize the dyadic decomposition used by most classical $t + 2D$ video-coding schemes. For highly textured content, the rate-distortion performance of adaptively generated wavelet packets is superior to that of the classical dyadic wavelet. As shown in section 4.2, best-basis algorithms (BBA) are the computationally most efficient techniques for finding wavelet-packet subband structures for a given image. This can be done not only by optimizing rate-independent information-cost functions operating in the transform domain (like entropy), which provides suboptimal rate-distortion performance only, but also by exploiting rate-distortion objective functions in the BBA [157].

Moreover, anisotropic decompositions were proposed as a solution for coding textures with homogeneous energy distribution. In [200], a lattice-based, perfect- reconstruction, critically sampled anisotropic multi-directional wavelet transform, called directionlet is presented, preserving the simplicity of computations and filter design from the dyadic wavelet transform and providing a sparse representation of the discontinuities in images. This transform was successfully applied both for still-image compression and denoising [65]. The 3D anisotropic wavelet-packet bases proposed by Kutil [111] for video coding come as an extension of the directionlets mentioned above. He introduces the *bush* decomposition, based on the idea that video data has different characteristics in time and spatial dimensions, and anisotropic processing can represent shapes and features with different properties in different directions. In a 3D bush, any of the horizontal, vertical, or quadtree transform is allowed at each decomposition level. Continuing in this direction, in [100] an anisotropic multidirectional representation is proposed for oriented textures, which can be successfully used for coding the high-frequency temporal subbands .

In the following, we propose to fully separate the horizontal and vertical transforms of the classical wavelet decomposition, thus the decorrelation description for a texture will be accomplished by two concatenated binary trees. Moreover, this decomposition inherits the simplicity of processing and filter design from the dyadic wavelet transform and presents perfect reconstruction. This decomposition retains 1D filtering and subsampling operations but can provide a highly anisotropic frequency representation. Indeed, the concatenation of two uni-directional decompositions is not equivalent to the dyadic (quadtree) classical decomposition, where the separability is applied at each level. Generally, the finer 2D frequency separation allows to better capture the orientation of the spatial details. The Fully Separable Wavelet Transform (FSWT) (see section 1.2, Fig. 1.4) may extend both classical dyadic wavelet or wavelet-packet decompositions, preserving at the same time its low-complexity characteristics. The contributions of this section have been presented in the proceedings of the SPIE-VCIP'06 [188] conference.

4.3.1 Fully separable wavelet transform

Generally, in the processing of still images, an isotropic 2D wavelet decomposition is used. It results from the tensor product of 1D wavelet bases of the form:

$$\{\phi_{J,k}(t),\ k \in \mathbb{Z}\} \cup \bigcup_{j \leq J} \{\psi_{j,k}(t),\ k \in \mathbb{Z}\}\ , \tag{4.12}$$

where J is the maximum number of decomposition levels. This alternation between horizontal and vertical decompositions at each level leads to *square* subbands, i.e. the mul-

tiresolution decomposition of Mallat [125]. Thus, for one given subband, the number of decomposition levels in the horizontal direction is the same as the number of decomposition levels in the vertical direction. This process is justified by the properties of natural still images: their texture features are often quite similar in all directions.

The construction of the dyadic 2D WT and its corresponding quadtree is shown in Fig. 4.8 (a,b). The resulting subbands from the quadtree decomposition are denoted at each level by LL, LH, HL, HH. Fig. 4.8 (c,d) represent the FSWT and its corresponding concatenated binary horizontal and vertical trees. The low-pass and high-pass subbands resulting from the 1D transform (horizontal or vertical filtering) are denoted by L and H, respectively.

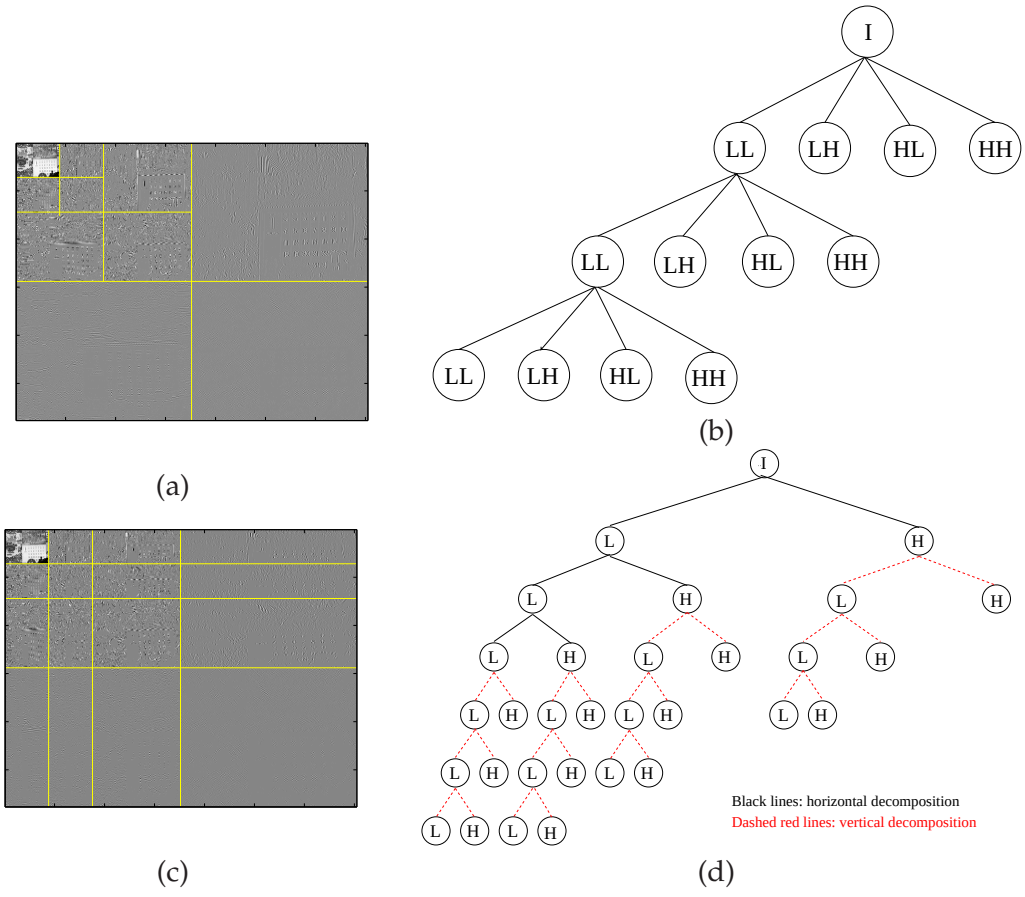


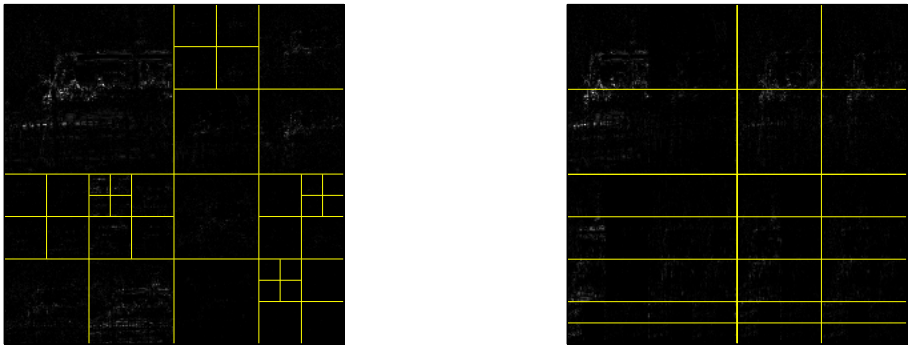
Figure 4.8: Three level spatial-decomposition scheme for Mobile(CIF, 30Hz) sequence: dyadic wavelet transform (WT) (a) and its decomposition quadtree (b); fully separable wavelet transform (FSWT) (c) and the two decomposition binary trees (vertical decomposition represented by full-line and the horizontal one by dashed-line)(d).

A 2D fully separable wavelet basis ($\mathcal{B}^{FS}$) consists of tensor products of *all* possible pairs of wavelets and scaling functions, i.e.:

$$\begin{aligned}\mathcal{B}^{\mathcal{FS}} &= \left(\{\phi_{J,k}(x)\}_k \cup \{\psi_{j,k}(x)\}_{j\leq J,k}\right) \otimes \left(\{\phi_{J,k}(y)\}_k \cup \{\psi_{j,k}(y)\}_{j\leq J,k}\right) \\ &= \{\phi_{J,k_1}(x)\phi_{J,k_2}(y)\}_{k_1,k_2} \cup \bigcup_{j_1\leq J} \{\psi_{j_1,k_1}(x)\phi_{J,k_2}(y)\}_{k_1,k_2} \cup \\ &\bigcup_{j_2\leq J} \{\phi_{J,k_1}(x)\psi_{j_2,k_2}(y)\}_{k_1,k_2} \cup \bigcup_{j_1,j_2\leq J} \{\psi_{j_1,k_1}(x)\psi_{j_2,k_2}(y)\}_{k_1,k_2}\end{aligned} \tag{4.13}$$

This means that one can find the FSWT expansion of a multidimensional function by simply operating on each coordinate axis separately, using a one-dimensional wavelet transform. In contrast, a multiresolution wavelet basis ($\mathcal{B}$) consists of tensor products of all possible pairs of wavelet and scaling functions *at the same scale*:

$$\mathcal{B} = \{\phi_{J,k_1}(x)\phi_{J,k_2}(y)\}_{k_1,k_2} \cup \bigcup_{j\leq J} \{\psi_{j,k_1}(x)\phi_{j,k_2}(y), \phi_{j,k_1}(x)\psi_{j,k_2}(y), \psi_{j,k_1}(x)\psi_{j,k_2}(y)\}_{k_1,k_2} \tag{4.14}$$



(a) (b)

Figure 4.9: Four level spatial-decomposition scheme on temporal detail frames for Bus (CIF, 30Hz) sequence: (a) wavelet-packet transform (WPT); (b) fully separable wavelet-packet transform (FSWPT).

4.3.2 Fully separable wavelet packet transform

As mentioned in section 4.2.1, wavelet packets are a generalized family of multiresolution bases that includes dyadic wavelets. In this case, both the low-frequency (approximations) and the high-frequency (details) components are decomposed (see Fig. 4.9 (a)). The selection of the best-basis decomposition for the fully separable wavelet packet transform is performed similary to the algorithm presented in section 4.2.1.2. After a full decomposition of the image following the two spatial directions, the bottom-up pruning BBA is first applied on the horizontal direction and the selection of the best-basis for each node is done according to the following comparison of the cost functions on the binary tree of the H decomposition, between the node and its children nodes:

$$\mathcal{Q}(f, \mathcal{B}^{\mathcal{FS}}{}_{j_1,j_2,k_1,k_2}) \leq \mathcal{Q}(f, \widetilde{\mathcal{B}}^{FS}_{j_1+1,j_2,2k_1,k_2} \cup \widetilde{\mathcal{B}}^{FS}_{j_1+1,j_2,2k_1+1,k_2}), \forall j_2 \tag{4.15}$$

Once the best-basis $\widetilde{\mathcal{B}}^{FS}$ for the horizontal direction is obtained, we proceed with a BB selection in the vertical direction, keeping fixed the optimal horizontal tree and using the same cost criterion for the vertical tree pruning. Thus if:

$$\mathcal{Q}(f, \widetilde{\mathcal{B}}^{FS}_{j_1,j_2,k_1,k_2}) \leq \mathcal{Q}(f, \widetilde{\widetilde{\mathcal{B}}}^{FS}_{j_1,j_2+1,k_1,2k_2} \cup \widetilde{\widetilde{\mathcal{B}}}^{FS}_{j_1,j_2+1,k_1,2k_2+1}), \forall j_1 \tag{4.16}$$

we can conclude that the best-basis for the node (j_1, j_2, k_1, k_2) is $\widetilde{\mathcal{B}}^{FS} = \widetilde{\widetilde{\mathcal{B}}}^{FS}$.

The result of this algorithm is illustrated in Fig. 4.9 (b). Concerning the computational complexity of these fully separable transforms, it was shown in [147] that they have linear complexity, $\mathcal{O}(N)$, as in the case of the classical dyadic transform.

4.3.3 Experimental results

For the experiments, we have considered three representative video sequences: Mobile (CIF, 30 Hz), Bus (CIF, 30 Hz) and City (4CIF, 60 Hz). The tests have been made in the framework of the Vidwav [212] video codec. Motion estimation is block-based and the motion-vectors field have been estimated with $1/4$-pixel accuracy. All video sequences used in our tests have been temporally decorrelated with a 5-level motion-compensated 5/3 decomposition. Spatio-temporal wavelet coefficients have been encoded using the 3D-ESCOT [216] algorithm which supports classical dyadic wavelet and wavelet-packet spatial decompositions. The bitrate test points for the sequences used in our simulations correspond to those defined for the MPEG standardization [215].

YSNR (dB) for Bus (CIF, 30Hz) sequence					
Filter/Bit rate	256	320	384	448	512
9/7	27.47	28.38	29.19	29.86	30.50
FS-9/7	27.55	28.45	29.27	29.92	30.53
5/3	27.26	28.28	28.99	29.67	30.30
FS-5/3	27.29	28.26	29.08	29.69	30.31
JWPT-9/7	27.56	28.46	29.26	29.93	30.60
JFSWPT-9/7	**27.61**	**28.56**	**29.39**	**30.06**	**30.73**
JWPT-5/3	27.45	28.50	29.23	29.91	30.52
JFSWPT-5/3	27.47	28.55	29.24	29.93	30.54

Table 4.15: Rate-distortion comparison for the sequence Bus (CIF, 30Hz).

In the experimental framework, the temporal approximation subbands have been spatially decomposed over 5 levels with the biorthogonal 9/7 and 5/3 filter banks. For the spatial decomposition of the temporal frames, we have compared the following representations:

- ⋆ dyadic WT with 9/7 and 5/3 filter banks (9/7 and 5/3).
- ⋆ FSWT with 9/7 and 5/3 filter banks (FS-9/7 and FS-5/3).
- ⋆ joint wavelet packets [191] using 9/7 and 5/3 filter banks (JWPT-9/7 and JWPT-5/3).

YSNR (dB) for City (4CIF, 60Hz) sequence					
Filter Bit rate	1024	1280	1536	1792	2048
9/7	34.25	34.65	35.08	35.48	35.77
FS-9/7	34.38	34.74	35.19	35.61	35.89
5/3	34.07	34.44	34.86	35.15	35.43
FS-53	34.19	34.47	34.87	35.28	35.55
JWPT-9/7	34.31	34.70	35.14	35.54	35.85
JFSWPT-9/7	**34.32**	**34.70**	**35.14**	**35.54**	**35.84**
JWPT-5/3	34.10	34.46	34.89	35.16	35.44
JFSWPT-5/3	34.11	34.44	34.89	35.16	35.43

Table 4.16: Rate-distortion comparison for the sequence City (4CIF, 60Hz).

- ⋆ joint fully separable wavelet packets using 9/7 and 5/3 filter banks (JFSWPT-9/7 and JFSWPT-5/3).

In Tabs. 4.15-4.16 we compare the YPSNR results for the Bus (CIF, 30Hz) and City (4CIF, 60Hz) sequences. For the same tree depth, we observe generally better results for a fully separable decomposition than for the dyadic wavelet representation. Moreover, this difference is more visible for a reduced number of decomposition levels (see Fig. 4.10), where the finer 2D frequency separation allows better capture of the orientation of the spatial details, achieving an average gain of 0.25 dB over their dyadic counterparts. For a higher number of decomposition levels, there is not very much information in the highest frequency subbands.

4.3.4 Conclusion

We have presented in this section an evaluation of fully separable wavelet and wavelet packet transforms for texture encoding in a motion-compensated subband video codec. As shown by the simulations results, the finer 2D frequency separation given by the fully separable transforms allows better capture of the orientation of the spatial details, resulting in a better representation of the video texture, in comparison to classical dyadic decompositions.

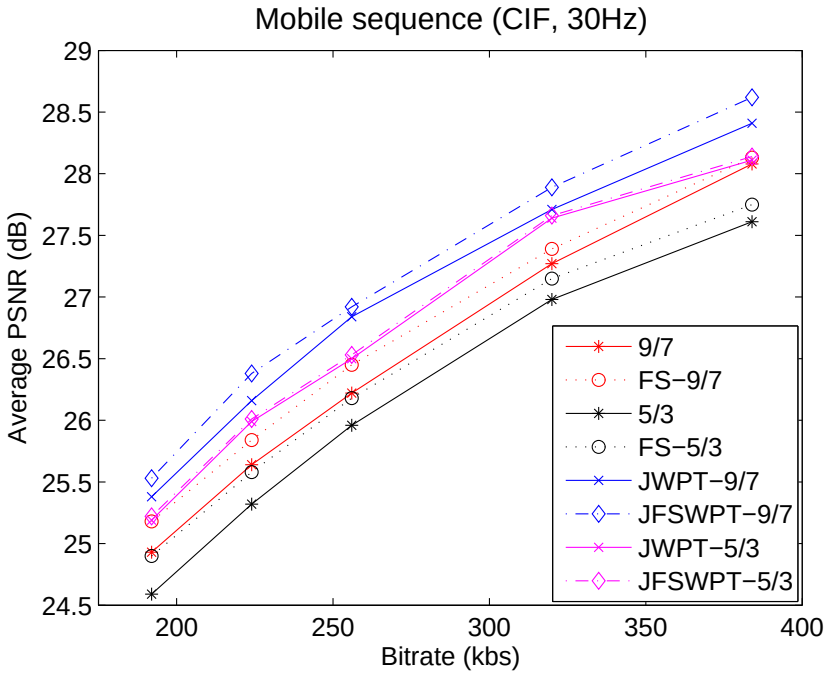


Figure 4.10: Rate-distortion comparison for 2-level spatial decomposition for the Mobile (CIF, 30Hz) sequence.

4.4 Conclusion

In this chapter, we have presented our contributions to the spatial processing stage of the $t + 2D$ video coding framework. Following the study regarding the properties of the temporal decomposition subbands, we have concluded that the dyadic wavelet decompositions are not the most appropriate for representing the residual temporal subbands. We have thus proposed a joint wavelet-packet representation for the detail temporal subbands, as well as an improved and efficient best-basis algorithm for biorthogonal wavelets. This way, a unique best basis representation is found for several frames, and not one basis per frame, as classically done. Moreover, fully separable wavelet and wavelet-packet transforms have been studied for spatial decorrelation of the temporal subbands, these decompositions leading to a better video-texture representation, due to an efficient capture of spatial detail orientation.

After the spatial-coding module, the resulting coefficients are passed to a quantizer. We present in the following the design of a graph-cut solvable energy functional for the Lagrangian rate-distortion optimization problem. Three graph-based solutions are described, by modelling several aspects of energy interactions for the minimization of a non-orthogonal system. This way optimal rate-distortion truncation of scalable streams is achieved, including trade-offs at various rates.

Chapter 5

Rate distortion optimization using graph cuts

Min-cut algorithms on graphs emerged as an increasingly useful tool for problems in vision. Typically, the use of graph cuts is motivated by one of the following two reasons. Firstly, graph cuts allow geometric interpretation: under certain conditions a cut on a graph can be seen as a hypersurface in the N-D space embedding the corresponding graph. Thus, many applications in vision and graphics use min-cut algorithms as a tool for computing optimal hypersurfaces. Secondly, graph cuts also work as a powerful energy minimization tool for a fairly wide class of binary and non-binary energies that frequently occur in early vision. In some cases, graph cuts produce globally optimal solutions.

More generally, there are iterative graph-cut techniques that produce provably good approximations which have been empirically shown to correspond to high-quality solutions in practice. Thus, another large group of applications use graph-cuts as an optimization technique for low-level vision problems based on global-energy formulations.

This chapter starts by explaining the general theoretical properties that motivate the use of graph cuts by giving some examples of graph-cut solutions already existing in computer graphics (section 5.1). In a second part (section 5.2), we design a graph-cut-solvable energy functional for the Lagrangian rate-distortion-optimization problem. Moreover, we study three possible solutions by modeling several aspects of energy interactions for the minimization of a non-orthogonal functional. We present our results in the context of subband image compression; although, the proposed graph-cut rate distortion optimization method could also be jointly used with existing coding algorithms for video compression.

5.1 Graph-cuts in computer vision

Many computer-vision problems can be formulated in terms of energy minimization. In the last few years, minimum cut/maximum network-flow algorithms [107, 30] have emerged as an elegant and increasingly useful tool for exact or approximate energy minimization. In the following, we review several concepts from graph theory and introduce the graph-cut energy-minimization technique. We also present several examples of graph-cut solutions for some signal-processing applications.

5.1.1 Graph representation

In the following, we present a short overview of graph-based algorithms, and we introduce the graph-cut and multiway min-cut definitions.

5.1.1.1 Definition

A graph is a mathematical representation that is useful for solving many kinds of problems. Fundamentally, a graph consists of a set of vertices and a set of edges, where an edge is a link between two vertices in the graph. Generally, a graph G is given by a pair (V, E), where V is a finite set, and E is a binary relation on V. V is called a vertex set whose elements are called vertices or graph nodes. E is a collection of edges, where an edge is a pair (u, v) with $u, v \in V$. In a directed graph, edges are ordered pairs, connecting a source vertex to a target vertex. In an undirected graph, edges are unordered pairs and connect two vertices in both directions, hence in an undirected graph (u, v) and (v, u) are two ways of writing the same edge (symmetry).

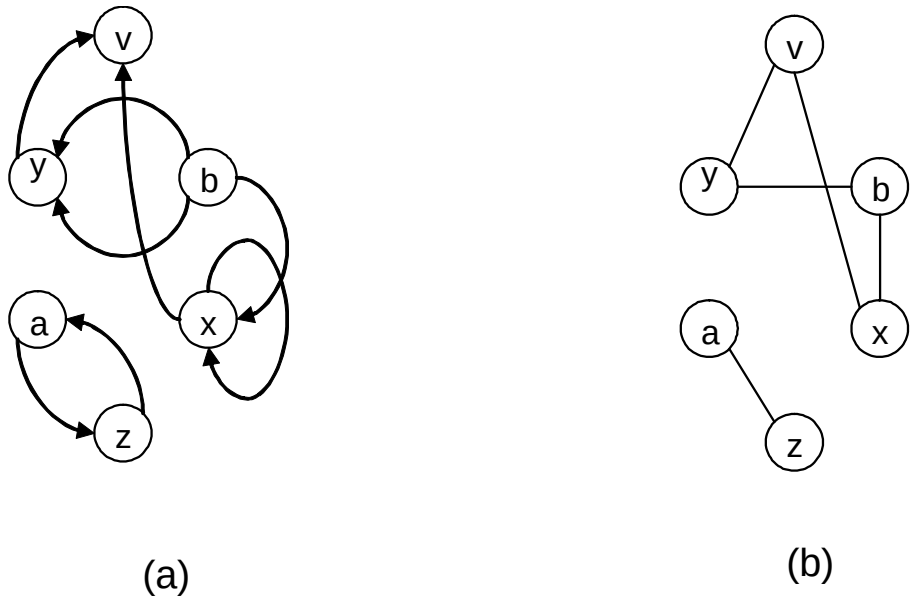


Figure 5.1: Example of: (a) directed and (b) undirected graph.

This graph definition is purely abstract; it does not say what a vertex or edge represents. They could be cities with connecting roads, or web pages with hyperlinks, or pixels. These details are left out of the definition of a graph for an important reason; they are not a necessary part of the graph abstraction. By leaving out the details, we can construct a theory that is reusable that can help us solve a number of different kinds of problems.

The primary property of a graph to consider when deciding which data-structure representation to use is sparsity, the number of edges relative to the number of vertices in the graph. A graph where E is close to V^2 is a dense graph, whereas a graph where $E = \alpha V$ with α much smaller than V is a sparse graph. For dense graphs, the adjacency-matrix representation is usually the best choice, whereas for sparse graphs, the adjacency-list representation is a better choice. Also the edge-list representation is a space efficient

choice for sparse graphs that is appropriate in some situations.

5.1.1.2 Graph-based algorithms

Breadth-First Search Breadth-first search (BFS) is a traversal through a graph that touches all the vertices reachable from a particular source vertex. In addition, the order of the traversal is such that the algorithm will explore all of the neighbors of a vertex before proceeding to the neighbors of its neighbors (level search). One way to think of breadth-first search is that it expands like a wave emanating from a stone dropped into a pool of water. Vertices in the same wave are at the same distance from the source vertex. A vertex is considered discovered the first time it is encountered by the algorithm. It becomes visited after all its level neighbors have been are explored.

Depth-First Search A depth-first search (DFS) visits all the vertices in a graph. When choosing which edge to explore next, this algorithm always chooses to go deeper into the graph. That is, it will pick the next adjacent unvisited vertex until reaching a vertex that has no unvisited adjacent vertices. The algorithm will then backtrack to the previous vertex and continue along any as-yet unexplored edges from that vertex. After DFS has visited all the reachable vertices from a particular source vertex, it chooses one of the remaining undiscovered vertices and continues the search. This process creates a set of depth-first trees which together form the depth-first forest.

Minimum Spanning Tree For a weighted graph $G = (V, E, W)$, where W is the set of weights associated to the graph edges, the minimum-spanning-tree problem is defined as follows: find an acyclic subset T, $T \in E$ that connects all of the vertices in the graph and whose total weight is minimized, where the total weight is given by:

$$w(T) = \sum_{(u,v)\in T} w(u, v)$$

where $w(u, v)$ is the weight on the edge (u, v). T is called the minimum spanning tree.

Shortest-paths algorithms One of the classic problems in graph theory is to find the shortest path between two vertices in a graph. Formally, a *path* is a sequence of vertices $(v_0, v_1, ..., v_k)$ in a graph $G = (V, E)$ such that each vertex is connected to the next vertex in the sequence (so the edges (v_i, v_{i+1}) for $i = 0, 1, ..., k - 1$ are in the edge set E). In the shortest-path problem, each edge is given a real-valued weight. We can therefore talk about the weight of a path:

$$w(p) = \sum_{i=1}^{k} w(v_{i-1}, v_i)$$

Usually, the weights are given by heuristics, in order to speed-up the search process. For example, a weight or cost for passing from vertex u to v could be:

$$\delta(u, v) = \min w(p) \quad if \quad \exists(u \rightarrow v)$$
$$\delta(u, v) = \infty \quad otherwise.$$

where $(u \rightarrow v)$ means that there is a path from u to v. A shortest path is thus any path whose weight is equal to the shortest path weight.

There are several variants of the shortest-path problem. Above, we defined the single-pair problem, but there is also the single-source problem (all shortest paths from one vertex to every other vertex in the graph), the equivalent single-destination problem, and the all-pairs problem.

Network-flow algorithms A flow network is a directed graph $G = (V, E)$ with a source vertex s and a sink vertex t. Each edge has a positive real-valued capacity function c, and there is a flow function f defined over every vertex pair. The flow function must satisfy three contraints:

$$f(u,v) \leq c(u,v), \qquad \forall (u,v) \in V \times V \quad (Capacity \quad constraint)$$

$$f(u,v) = -f(v,u) \qquad \forall (u,v) \in V \times V \quad (Skew \quad symmetry)$$

$$\sum_{v \in V} f(u,v) = 0 \qquad \forall u \in V - \{s,t\} \quad (Flow \quad conservation)$$

The flow of the network is the net flow entering the sink vertex t (which is equal to the net flow leaving the source vertex s).

$$|f| = \sum_{u \in V} f(u,t) = \sum_{v \in V} f(s,v)$$

The maximum-flow problem is to determine the maximum possible value for $|f|$ and the corresponding flow values for every vertex pair in the graph.

5.1.1.3 Maximum-flow / minimum-cut problem equivalence

The $s-t$ minimum-cut problem is intimately related to the maximum-flow problem. The input is the same as for the maximum-flow problem. The goal is to find a partition of the nodes that separates the source and sink so that the total capacity of edges going from the source side to the sink side is minimum. Formally, we define an $s-t$ *cut* $[S,T]$ to be a partition of the nodes V = $S \cup T$ such that $s \in S$ and $t \in T$. The cost of a cut is defined to be the sum of the capacities of "forward" arcs in the cut:

$$C[S,T] = \sum_{u \in S, v \in T} c(u,v)$$

The goal is to find an $s-t$ cut of minimum capacity. It is easy to see that the value of any flow is less than or equal to the capacity of any $s-t$ cut. Any flow sent from s to t must pass through every $s-t$ cut, since the cut disconnects the terminal nodes s from t. Since flow is conserved, the value of the flow is limited by the capacity of the cut. A max-flow min-cut equivalence demonstration as well as a simple algorithm for finding the min-cut are given by Ford and Fulkerson in [75].

5.1.1.4 Multiway minimum cut

Given an undirected graph $G = (V, E)$ with non-negative edge costs and a set of T special nodes in the graph (called terminals, $T \subset V$), the multiway minimum cut is given by the cheapest multiway cut, i.e., a subset of the edges whose removal disconnects each

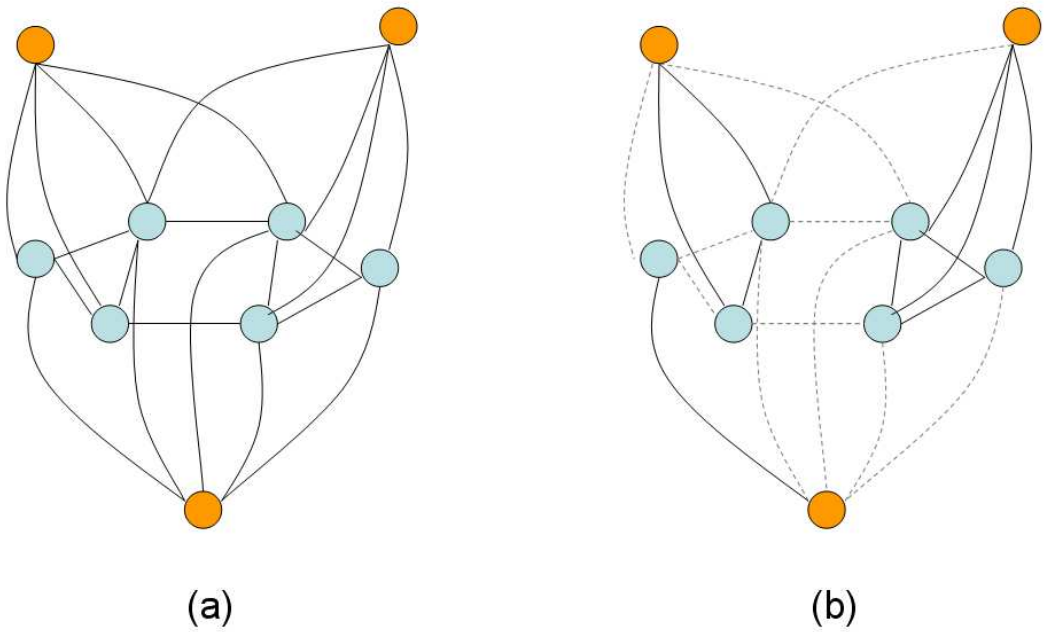


Figure 5.2: Three-way graph cut example: (a) initial graph, (b) induced three-way cut graph (cut-edges in dashed gray lines, terminal nodes in orage, non-terminal nodes in blue).

terminal from the rest (Fig. 5.2). This is one of several generalizations of the classical undirected $s - t$ cut problem.

Dahlhaus *et al.* [54] initiated the study of multiway cut. For $T = 2$, the problem is identical to the undirected version of the extensively studied $s - t$ min-cut problem of Ford and Fulkerson [75] and thus has polynomial-time algorithms (see, e.g. [14]). Prior to this method, the best approximation algorithm for $T \geq 3$ was due to the above mentioned paper [54]. For fixed T in planar graphs, the problem is solvable in polynomial time [121]. For trees, there are linear-time algorithms [46].

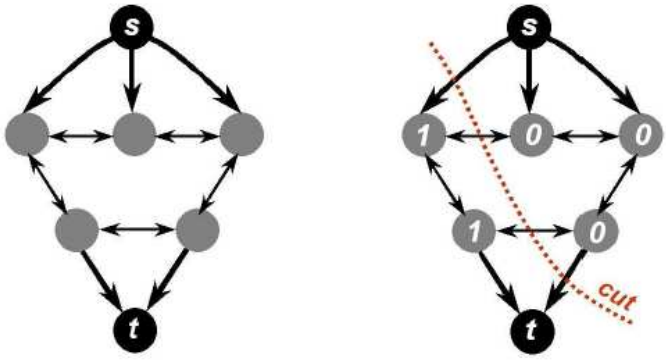
5.1.2 Energy minimization using graph-cuts

After the above review of the graph-cut definition, we present in the following the problem statement for energy minimization, as well as several signal-processing applications involving graph-cut optimizations.

5.1.2.1 Binary energy function model

We show in the following that the minimum-graph-cut algorithm is a powerful optimization tool which is inherently binary. This is particularly useful in vision applications because it can be used for enforcing piecewise coherence.

Recall that a graph cut is a partition of all non-terminal nodes into two sets, S and T. Let each non-terminal node $v \in V$ represent a binary variable, and let us associate 0 with the source s and 1 with the sink t (Fig. 5.3). Then we can identify any cut with a particular



(a) A graph $\mathcal{G}$ (b) A cut on $\mathcal{G}$

Figure 5.3: Binary energy graph cut example: (a) initial graph, (b) induced min cut on graph.

assignment to binary variables (i.e. nodes) as follows: if node $v \in S$ then its binary value is 0, and if node $v \in T$ then its binary value is 1. This procedure establishes a one-to-one correspondence between the set of all variable assignments and the set of graph cuts. In Fig. 5.3(b), we label each node with the binary value corresponding to the given cut. If N is the number of non-terminal nodes in the graph, then there are 2^N possible graph cuts. The cost of a given cut is:

$$C[S,T] = \sum_{u \in S, v \in T} c(u,v)$$

and this cost can be used as the objective function to evaluate the quality of the partition $S \cup T = V$ or, equivalently, as the energy of the variables assigned to the non-terminal nodes. Then, the minimum cut corresponds to the optimal variable assignment for the nodes. Thus, the power of the graph-cut algorithm is that it can search efficiently through the exponential space of solutions and find the optimal solution in polynomial time. In this sense, graph cuts are similar to dynamic programming which also can find the optimum in an exponential space of solutions in polynomial time. The significant disadvantage of dynamic programming is that it can be applied only to tree graphs, while graph cuts can be applied to arbitrary graphs.

5.1.2.2 Multi-terminal energy-function model

Previously, we have seen that graph cuts provide an inherently binary optimization. In the following, we will show that they can also be used for multi-label or multi-terminal energy minimization. This minimization is sometimes exact, but usually it is approximate.

Many problems in vision can be formulated as labeling problems. We have already seen an example of a binary-labeling problem in section 5.1.2.1. We now state it in a more general form. In a labeling problem, we have a set of units X which can represent pixels, voxels, or any other set of entities. We also have a finite set of labels T which is now allowed to be of size larger than 2. Labels can represent any property that we wish to assign to units, for example intensity, stereo disparity, motion vectors, and so on. The

goal is to find a labeling f which is a mapping from X to labels T. Let us use f_n to denote the label assigned to unit n and f to denote the collection of such assignments for all units in X.

For each problem, one can derive a set of constraints which should be respected as much as possible by the optimal labeling f. Usually these constraints are derived from the observed data and from a priori knowledge. The data constraints are typically expressed as individual preferences of each unit $n \in X$ for labels in T. To formalize them, for each unit n, we use a function $D_n(t) : T \to \mathbb{R}$. D being part of the functional we want to minimize, the smaller the value of $D_n(t)$, the more likely is the label t for unit n. If for some n, $D_n^m = \min_{t \in T} D_n(t) < 0$, then we subtract D_n^m from $D_n(t)$ for all $t \in T$. This does not change our energy formulation. Thus, from now on, we assume, without loss of generality, that $D_n(t) > 0$, for all n, t.

The a priori knowledge can be generally complex, but in graph-cut-based optimization, we are esentially limited to constraints which impose different types of spatial smoothness on the labeling f. In order to enforce these smoothness constraints, we define a neighborhood system $\mathcal{N}$ on the units X. A neighborhood system contains pairs of units which are immediate neighbors. The simplest example of a neighborhood system is the four-connected grid. We consider a function $\mathcal{V}_{n_1,n_2}(t_{n_1}, t_{n_2})$ which assigns a positive cost if the neighbour units n_1, n_2 are given different labels t_{n_1}, t_{n_2}. Different choices of $\mathcal{V}$ lead to different assumed types of smoothness.

Now we are ready to formulate the labeling problem in terms of energy minimization. The optimal labeling f is the one which minimizes the following energy function:

$$E(f) = \sum_{n \in X} D_n(f_n) + \sum_{n_1,n_2 \in \mathcal{N}} \mathcal{V}_{n_1,n_2}(f_{n_1}, f_{n_2}) \tag{5.1}$$

As mentioned in 5.1.1.4, there are several algorithms for computing the minimum multiway cut and thus minimizing a functional like Eq. (5.1). Sometimes the global minimum is reached [32, 86, 121], in other cases a local minimum within a known factor of the global is found [33].

Label-expansion algorithm One of the most effective algorithms for minimizing discontinuity- preserving energy functions is the expansion-move algorithm introduced in [33]. This algorithm can be used whenever $\mathcal{V}$ is a metric on the space of labels $\mathcal{T}$. $\mathcal{V}$ is called a *metric* on the space of labels $\mathcal{T}$ if it satisfies:

$$\begin{aligned} \mathcal{V}(\alpha, \beta) = 0 \quad &\Leftrightarrow \quad \alpha = \beta \\ \mathcal{V}(\alpha, \beta) &= \mathcal{V}(\beta, \alpha) \geq 0 \\ \mathcal{V}(\alpha, \beta) &\leq \mathcal{V}(\alpha, \gamma) + \mathcal{V}(\gamma, \beta) \end{aligned}$$

for any labels $\alpha, \beta, \gamma \in \mathcal{T}$.

Consider a labeling f and a particular label α. A new labeling f' is defined to be an α-expansion move from f if $f'_n \neq \alpha$ implies $f'_n = f_n$. This means that the set of units assigned to the label α has increased when going from f to f'.

The expansion-move algorithm cycles through the labels in a certain order (fixed or random) and finds the lowest energy α-expansion move from the current labeling via graph cut. If this expansion move has lower energy than the current labeling, then it becomes the current labeling. The algorithm terminates with a labeling that is a local minimum of the energy with respect to expansion moves; more precisely, there is no α-expansion move, for any label α, with lower energy.

Label-swap algorithm When $\mathcal{V}$ is a *semimetric* on the labels space $\mathcal{T}$, that is $\forall \alpha, \beta \in \mathcal{T}$:

$$\mathcal{V}(\alpha, \beta) = 0 \quad \Leftrightarrow \quad \alpha = \beta$$
$$\mathcal{V}(\alpha, \beta) = \mathcal{V}(\beta, \alpha) \geq 0$$

an $\alpha - \beta$ label-swap algorithm is proposed [33].

Consider now a possible labeling f and all the pairs of labels (α, β); f' is defined to be an $\alpha - \beta$-swap from f if $f_{n_1} = \alpha$, $f_{n_2} = \beta$ swaps to $f_{n_1} = \beta$, $f_{n_2} = \alpha$, where the f' label assignment implies a lower energy that f. As in the expansion-move, the algorithm parses all the label pairs and finds the best $\alpha - \beta$ partitioning via graph cuts. The algorithm ends when no label exchange in the current partitioning can assure a lower energy.

It has been proven [33] that both the label-expansion and label-swap algorithms finish in $O(|T|)$ cycles. Generally, they assure a local minimum with respect to the swap or expansion space for a general label space. If the label space can be linearly ordered, the global minimum can be reached [31].

5.1.2.3 Graph-cut applications in computer vision

Many computer-vision problems require energy minimization. Each of these methods consists in modeling a graph for an energy type such that the minimum cut minimizes globally or locally that functional. We present in the following several fields in which graph-cut energy minimization has been applied.

Image segmentation Image segmentation consists in grouping similar pixels together to form a set of coherent image regions given a single image. Pixel similarity can be measured in terms of location, intensity, color, or texture. Graph-cuts seem to be a natural mechanism for segmentation as graph-cuts imply data partitioning. The basic idea of this approach is the following:

- ⋆ each image pixel is viewed as a vertex of a graph;
- ⋆ the similarity between two pixels is viewed as the edge weight between these two vertices;
- ⋆ segmentation is achieved by cutting edges in the graph to form a good set of disconnected components.

Good image-segmentation results obtained via graph-cuts results were presented in [34, 29, 170]. Fig. 5.4(b) illustrates the result of the graph-cut segmentation algorithm presented in [199] on the "Peppers" (128×128) image.

Motion segmentation The motivation of the motion-segmentation problem is given by the fact that motion regions (pixels with similar motion) usually correspond to distinct objects in the scene.

Motion-vector fields are obtained by establishing pixel correspondences between images. Based on the motion vectors, the pixels are then grouped into motion regions, thus producing a segmentation of the image (Fig. 5.5). In [163, 204], moving object segmentation is considered as an image-labeling process. All pixels grouped together by the same label and geometrical neighbours are considered to describe a moving object which is

(a) (b)

Figure 5.4: Image segmentation example [199]: (a) original image, (b) segmentation induced by graph-cuts.

extracted from the scene. As in the case of image segmentation, each image pixel is considered a vertex in the graph, the labels/terminals being given by the possible motion vectors so that pixels are assigned labels corresponding to their motion flow between consecutive frames.

Image restoration Many applications (i.e. satellite imaging, medical imaging, astronomical imaging, etc.) require image quality better than that of the initial date, since images are often distortioned due to aquisition or transmission constraints. The goal of image restoration is to ameliorate image quality.

Good image-restoration results obtained via graph-cuts results were recently presented in [55, 56]. Moreover, in [33, 155], a pixel-labeling solution for image restoration was proposed. The graph is constructed in grid manner according to pixel positions where the weights between neighbours are directly proportional to the degree of similarity between pixels (i.e., similar intensities, small weight value and different intensities, high weight value). The labels are given by an intensity-value set. Fig. 5.6(b) illustrates the result of a graph-cut denoising algorithm, where the pixel-interaction energy is modeled in a discontinuity-preserving way [33].

Stereo video In stereo-video applications, one problem is to find pixels in images of different cameras (for example, the input images of Fig. 5.7(a,b)), that correspond to the same 3D point of the observed scene. The goal of most stereo-matching algorithms is to compute one disparity estimate for each pixel in a reference image, often chosen as the left input image. These disparity estimates are often interpreted as the inverse distances to observed objects (the ground-truth data in Fig. 5.7(c)). The disparity of a pixel in the reference image describes the distance, measured in pixels, between the pixel under consideration and its corresponding pixel in the other input image.

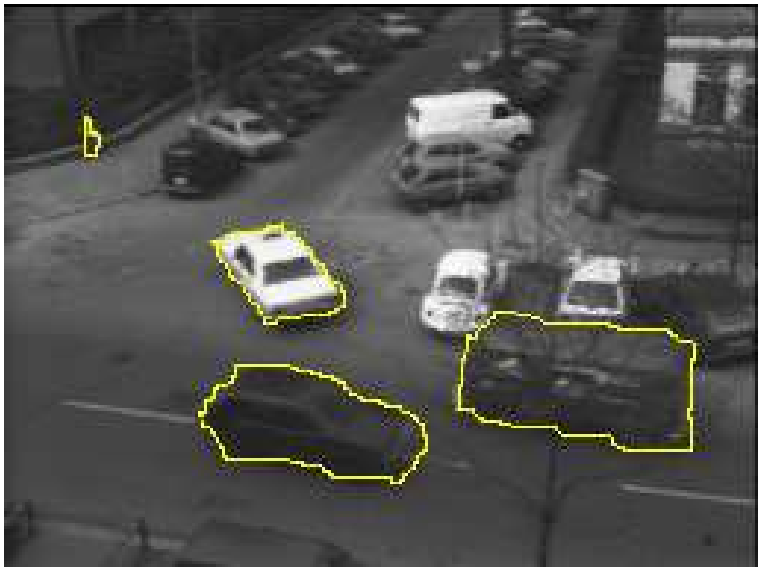


Figure 5.5: Motion segmentation example [163] on Taxi video sequence.

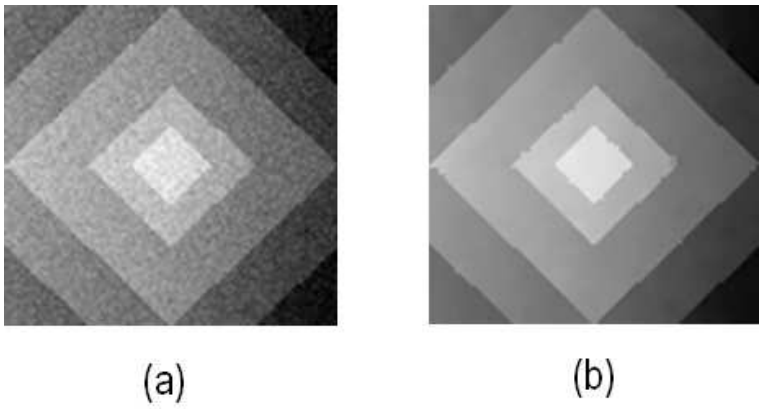


Figure 5.6: Image restoration example [33]: (a) noisy image; (b) restored image using graph-cuts.

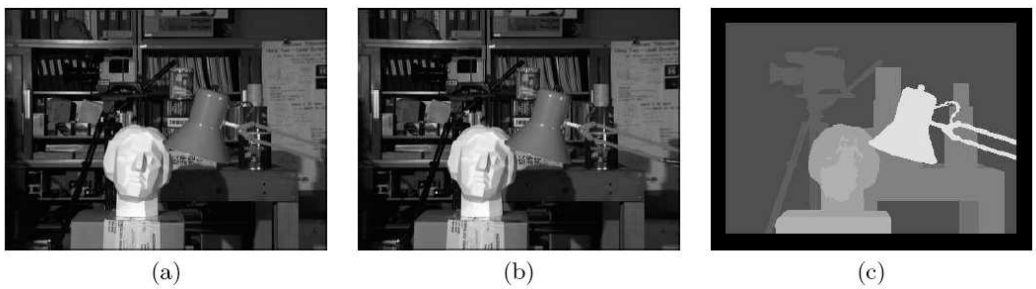


Figure 5.7: Tsukuba benchmark stereo pair [32]: (a) the left input image; (b) the right input image; (c) ground truth data, where the intensity represents the disparity, meaning that lighter gray represents objects closer to the cameras.

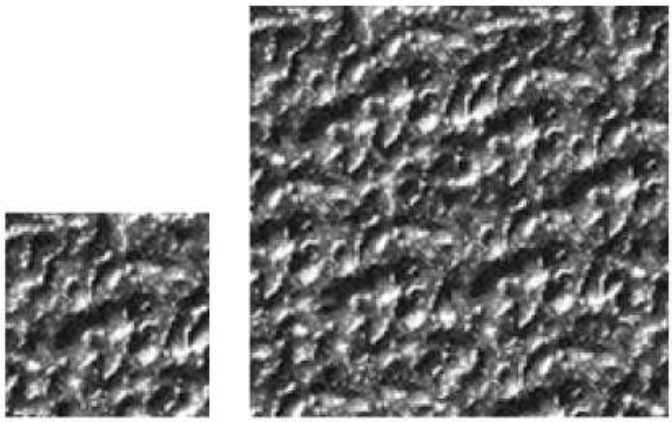


Figure 5.8: Texture synthesis example: (a) original image; (b) new image obtained by texture synthesis using graph-cuts [112].

Traditionally, most methods for finding correspondences between images from two cameras are based on correlation of local windows or on matching of sparse features. These global methods often optimise an energy function describing a relationship between image measurements and a prior model of the observed world. The energy function may, for example, include a matching cost for assigning a pixel a certain depth value, and a smoothness cost for assigning different depth values to neighbouring pixels. The first method using graph cuts to solve the stereo correspondence problem was introduced by Roy and Cox [160]. Their result was a local coherence constraint that suggests that, locally, a disparity map is smooth, which means that pixels close to each other in any direction have similar disparities. Later Ishikawa and Geiger [97] solve the problem using a discontinuity-preserving function for neighbour pixels. Another approach for solving the correspondence problem as the maximum flow in a directed graph is taken by Boykov, Veksler and Zabih [32], where the correspondence is modeled using a Potts model for each pair of neighbouring pixels.

Texture synthesis Texture-synthesis techniques are used to reproduce the pattern contained in a sample image into either a larger 2D image or a surface in 3D. A variety of techniques have been employed for 2D image-texture synthesis. For example, in Fig. 5.8, given the input image (a), a good texture-synthesis algorithm should be able to generate more of the texture in order to create an image like (b). The example was created through graph-cut texture synthesis [112] which uses patches of texture to create its output image.

The overlapping region between two patches is set up as a graph where each pixel in the overlap is represented by a node, the weights being intensity-based functions. Graph-cut algorithms for texture synthesis were succesfully used in both the image- and video-processing fields [112, 219].

5.2 Still-image compression using graph cuts

Nowadays, the majority of image-compression algorithms use wavelet transforms, attempting to exploit all the signal redundancy that can appear within and across the different subbands of a spatial decomposition. However, efficiency of a coding scheme highly depends on bit allocation. A general subband-based image-compression scheme has three major modules: a spatial-decorrelation module, in which the spatial image redundancy is reduced; a quantization module, in which the best quantizers in terms of rate-distortion optimization are selected such that the bitstream matches the bandwidth capacity; and an entropy-coding module.

In this section we present a rate-distortion optimization based on graph cuts, which can compress efficiently the coefficients of a critically sampled or even redundant, non-orthogonal transform. As shown in section 5.1.2.3, good energy-optimization results based on graph cuts were obtained in image restoration [155, 33], as well as in stereo video [32], motion segmentation [163], texture synthesis in image and video [112, 219], etc. We propose to use the graph-cut mechanism for the minimization of the rate-distortion Lagrangian function and thus find the optimal set of quantizers satisfying the imposed constraints. To this aim, we have designed a specialized graph able to represent a subband decomposition taking into consideration the correlations between subbands in a multiresolution approach.

In the following, we express the Lagrangian functional as a discrete sum accumulating the contribution of each coding unit (subband or block) in terms of rate and distortion induced by the quantization. Moreover, the graph model is *planar*[1] [121], and the energy function we intend to optimize is convex, so the minimum graph cut can be found in polynomial time. As it will be shown by the experimental results, the method gives good compression results compared to the state-of-the-art JPEG2000 codec, as well as an improvement in visual quality.

5.2.1 Graph design

Consider the graph $G = (V, E, W)$ with positive edge weights W, which have not only two, but a set of terminal nodes, $Q \in V$. Recall that a subset of edges $\mathcal{E}_C \in E$ is called a *multiway cut* if the terminal nodes are completely separated in the induced graph $G(\mathcal{E}_C) = (V, E - \mathcal{E}_C, W)$ and no proper subset of $\mathcal{E}_C$ separates the terminals in $\mathcal{E}_C$. If C is the cost of the multiway cut, then the multi-terminal min-cut problem is equivalent to finding the minimum-cost multiway cut. For our optimization problem, the terminals are given by a set of quantizers Q, and the coding units give the rest of the vertices $V - Q$. The edges and their weights/capacities will be defined in the following depending on the coding strategy (subband or block coding) and the distortion functional.

In [33], Y. Boykov *et al.* find the minimal multiway cut by succesively finding the min-cut between a given terminal and the other terminals. This approximation guarantees a local minimization of the energy function that is close to the optimal solution for both concave and convex energy functionals. As the rate-distortion Lagrangian lies on a convex-decreasing curve (i.e. $D(R)$), we propose to use the method in [33] for its optimization.

[1] A planar graph is a graph that can be drawn so that no edges intersect in the plane.

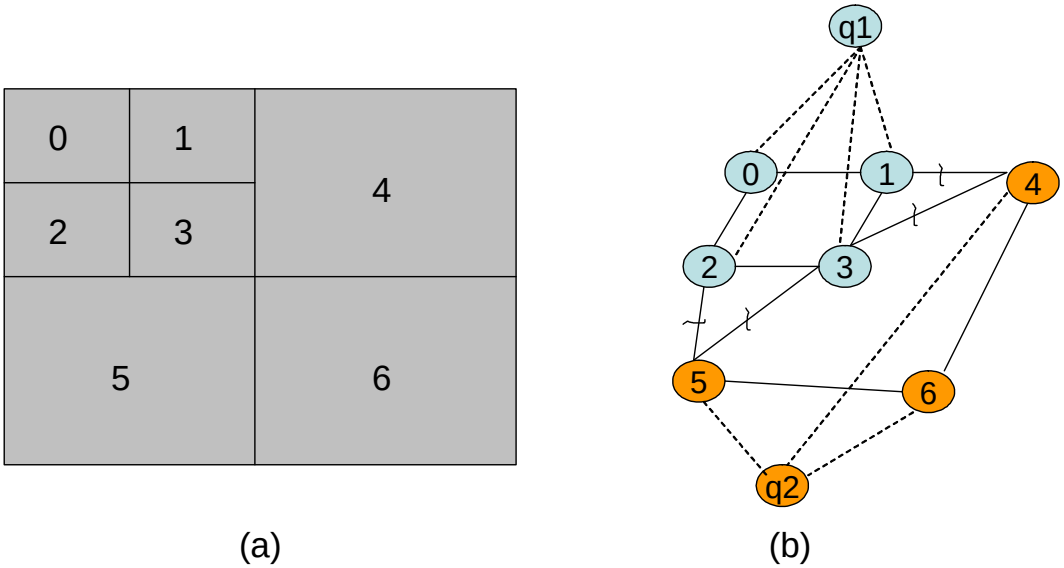


Figure 5.9: Two-level spatial decomposition (a) and the corresponding graph-cut partition for two quantizers (b) (q_1 partition in blue, q_2 partition in orange, where the regular edges are with full lines and the terminal edges with dashed ones).

5.2.2 Lagrangian rate-distortion functional

Consider the problem of coding an image at a maximal rate R_{max} with a minimal distortion D. Each image consists of a fixed number of coding units (spatial subbands or blocks of coefficients), each of them coded with a different quantizer q_i, $q_i \in Q$ (Q being the quantizers set). Let $D_i(q_i)$ be the distortion of the coding unit i when quantized with q_i, and let $R_i(q_i)$ be the number of bits required for its coding. The problem can now be formulated as: find $\min \sum_i D_i(q_i)$, such that $\sum_i R_i(q_i) = R \leq R_{max}$.

In the Lagrange-multiplier framework, this constrained optimization is written as the equivalent problem:

$$\min \sum_i \left(D_i(q_i) + \lambda R_i(q_i) \right), \quad R \leq R_{max} \tag{5.2}$$

where the choice of λ measures the relative importance between distortion and rate for the optimization and which can be determined using a binary search. The advantage of problem formulation in Eq. (5.2) is that the sum and the minimum operator can be exchanged to:

$$\sum_i \min \left(D_i(q_i) + \lambda R_i(q_i) \right), \quad R \leq R_{max} \tag{5.3}$$

This formulation obviously reveals that the global optimization can now be carried out independently for each coding unit, making an efficient implementation feasible.

5.2.2.1 Rate estimation

For the rate estimation of the quantized coding units we consider a non-contextual arithmetic coder [132], which uses a zero-order entropy model, where in the M quantized

coefficients of a given coding unit are random i.i.d. variables following a Gaussian distribution. Thus, the zero-order entropy (H) estimation in bits/variable (i.e., coefficient) is obtained as:

$$H = -\sum_{i=1}^{M} p_i \log_2 p_i, \tag{5.4}$$

where p_i is the probability of the i^{th} coefficient. The resulting entropy estimate per coding unit is weighted by the size of the coding unit in order to obtain the total entropy of the quantized image.

5.2.2.2 Distortion estimation

The distortion D between the original image x and the quantized one, $\widehat{x}$ is estimated in the following as the L^2 norm, i.e. :

$$D = \|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|^2 . \tag{5.5}$$

This model will be futher developed, in order to obtain a good distortion estimate in the spatial domain, rather than in the transform domain, as is usually done for orthonormal transforms.

5.2.3 Multiresolution-based graph modeling

In the following, two distortion models for our graph-cut based rate-distortion Lagrangian minimization are proposed, both methods being developed with the aim of encoding the coefficients of a critically sampled, or even redundant, non-orthogonal transform. In the first approach, we consider a first-order approximation model for the distortion, to finally represent it more accurately in the second approach which takes into consideration the cross-correlation distortion terms in a more complicated case. We show how the distortion can be approximated and then estimated in the spatial domain, allowing a graph modeling of the subband interactions. Moreover, in a third approach, the graph design is developed to model the coding units at a finer level of representation.

5.2.3.1 Graph design with first-order distortion at the subband level

As previously mentioned, the distortion D between the original image x, and the quantized image $\widehat{x}$ can be written as the L^2 norm, i.e. $D = \|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|^2$. For orthonormal transforms, this norm can be equivalently estimated in the transform domain. However, for arbitrary transforms (biorthogonal, redundant, non-linear etc.) this property does not hold any more. In the following, we focus on this more complicated case and show how the distortion can be approximated and then estimated in the spatial domain, allowing a graph modeling of the subband interactions.

If, in the reconstructed image $\widehat{x}$, we highlight the contribution of each subband, $\widehat{x} = \sum_i \widehat{x_i}$, where $\widehat{x_i}$ is the contribution per subband (i.e., the reconstructed image when only the i^{th} subband is quantized and the other subbands are set to zero), then we can also write the image in a similar way, $x = \sum_i x_i$. However, here x_i is completely arbitrary.

In the case of a linear basis, it may become $x_i = \sum_k \langle x, \widetilde{e}_{k,i} \rangle \, e_{k,i}$, where $\widetilde{e}_{k,i}$, $e_{k,i}$ are the analysis, respectively synthesis, elements of the biorthogonal basis. Then we have:

$$D = \left\| \sum_i (\widehat{x}_i - x_i) \right\|^2 = \sum_{i,i'} \langle \widehat{x}_i - x_i, \widehat{x}_{i'} - x_{i'} \rangle \tag{5.6}$$

In a first approximation, we can consider only the diagonal terms of the above development, i.e.:

$$D_I \cong \sum_i \|x_i - \widehat{x}_i\|^2 = \sum_i D_i(q_i) \tag{5.7}$$

which amounts to estimating the distortion between the contribution to the image and to the quantized image of only the i^{th} subband. This means we can reconstruct the image only from i^{th} subband coefficients (the others being set to zero) to get x_i and from the quantized coefficients of the i^{th} subband to get $\widehat{x}_i$.

In [33] are presented two graph-cut based algorithms able to reach a minimum for an energy function of the form:

$$E(f) = E_{data}(f) + E_{smooth}(f) \tag{5.8}$$

where E_{smooth} is a smoothness constraint, while E_{data} measures the distortion introduced by the f partitioning with respect to the original data. Without taking into consideration a rate constraint, one can easily associate E_{data} with D (i.e., $E_{data} = D$). Because E_{data} can be arbitrarily chosen, with only a positivity constraint, we add to it the rate factor, that is:

$$E_{data} = \sum_i (D_i(q_i) + \lambda R_i(q_i)) \tag{5.9}$$

We can define:

$$E_{smooth} = \sum_{n_1, n_2 \in \mathcal{N}} \mathcal{V}_{n_1,n_2}(q_{n_1}, q_{n_2}) \tag{5.10}$$

where $\mathcal{N}$ represents the 2D neighborhood system of the nodes and $\mathcal{V}_{n_1,n_2}(q_{n_1}, q_{n_2})$ measures the cost of assigning the quantizers q_{n_1}, q_{n_2} to the adjacent nodes n_1, n_2. We define $\mathcal{V}$ as the Potts interaction penalty, i.e. :

$$\mathcal{V} = \beta T(q_{n_1} \neq q_{n_2}) \tag{5.11}$$

where T is a boolean operator (e.g., its value equals 1 if its argument is true and 0 otherwise), and β is a real constant which enforces or diminishes the smoothing. As can be seen, the definition of E_{smooth} is consistent, as for two strongly correlated subbands the same quantizer choice is imposed. Moreover, it is a metric on the quantizer space, so the α-expansion algorithm [33] can be used for minimizing E.

A simple graph $G = (V, E)$ model for this energy functional can be obtained by seeing as the regular (planar) vertices the decomposition subbands (X), which are connected between them following their 2D geometrical position (thus $(E - XQ)$-regular links), and each terminal node $q \in Q$ being connected to all the non-terminal vertices (thus (XQ)-terminal links) (Fig. 5.9 shows the quantizer asigmenment after the cut, i.e., the output of the algorithm). So one can distinguish two connection types: one between regular vertices and the other one between the terminal nodes and the regular vertices. For a terminal-node edge (i.e., link between a quantizer q and a given coding unit vertex i), the

cost is given by the distortion induced by that quantizer to the image and the number of bits needed to transmit the quantized subband i, $D_i(q_i)+\lambda R_i(q_i)$. For a regular edge (i.e., a link between two neighbor vertices), the cost is 0 if the two nodes are quantized with the same quantizer or β otherwise. Moreover, this cost is dynamically computed for each possible partitioning f of the graph.

5.2.3.2 Graph design with cross-correlation distortion

Recall that we have written the distortion D between the original image, x, and the quantized one, $\widehat{x}$, as the L^2 norm, i.e. $D = \|\boldsymbol{x} - \widehat{\boldsymbol{x}}\|^2$. In a first approximation, we have considered only the diagonal terms, i.e.:

$$D_I \cong \sum_i \|x_i - \widehat{x}_i\|^2 \tag{5.12}$$

which amounts to estimating the distortion between the contribution to the image and to the quantized image of only the i^{th} subband.

In a second approximation, one can also consider the *cross-correlation* terms, i.e.:

$$D \cong D_I + \sum_i \sum_{i' \in \mathcal{N}(i)} \langle \widehat{x}_i - x_i, \widehat{x}_{i'} - x_{i'} \rangle \tag{5.13}$$

where $\mathcal{N}(i)$ is a neighborhood of i, containing closely correlated subbands. Indeed, given the limited support of the wavelets, the closer in scale and frequency are the subbands, the higher the correlation among them. In practice, this neighborhood is described by the geometrical position of the subbands in a multiresolution decomposition, where only the vertical and horizontal directions are considered (for example, in Fig. 5.9 and Fig. 5.10, the neighborhood relations are indicated by the black-marked edges in the graph). Eq. (5.13) can be written as:

$$D = \sum_i \underbrace{\|x_i - \widehat{x}_i\|^2}_{D_i} + \sum_i \sum_{i' \in \mathcal{N}(i)} \underbrace{\langle \widehat{x}_i - x_i, \widehat{x}_{i'} - x_{i'} \rangle}_{D_{i,i'}} \tag{5.14}$$

The second term involves the highest complexity (inverse transforms plus inner products between error images), which can however be divided by two, noting that $D_{i,i'} = D_{i',i}$ and therefore:

$$D \cong \sum_i \left(D_i + 2 \sum_{i > i'} D_{i,i'} \right) \tag{5.15}$$

For $D_{i,i'}$ we need to calculate the error between the image reconstructed from the i^{th} subband (x_i) and its equivalent reconstructed from the quantized i^{th} subband ($\widehat{x}_i$), the same from a neighboring subband i' and then compute the inner product.

The minimization of the energy function defined above is equivalent to the best partition of quantizers per subbands. The graph we have designed for solving this problem has as vertices the set of spatial subbands and the set of quantizers as terminal nodes, where the subbands are linked following the neighborhood system $\mathcal{N}$. Each terminal node is connected to all non-terminal nodes, considering all quantization possibilities for the spatial subbands. (i.e., $G = (V, E)$, where $V = X \cup Q$ and $E = E_N \cup E_Q$, E_N denoting the regular edges between subband vertices in the neighbourhood system $\mathcal{N}$ and E_Q the

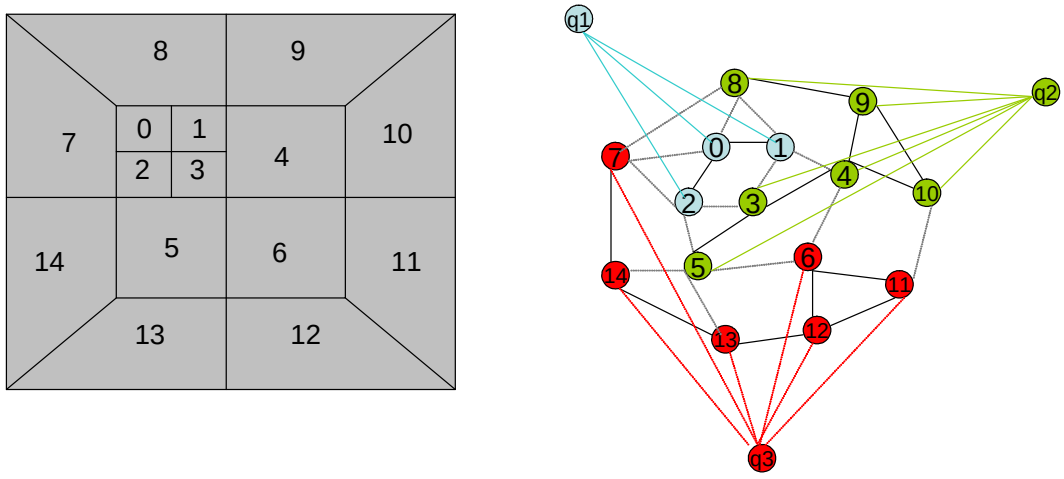


Figure 5.10: Contourlet decomposition with three levels (left) and three-way graph-cut repartition (right) (q_1 partition in blue, q_2 partition in green, q_3 partition in red, where the regular edges are with full black lines, terminal links in colors and the cut-edges in gray dash-lines).

terminal links between subband nodes and quantizers). One can distinguish two connextion types: E_N and E_Q. We define the weights for the quantizers links E_Q in terms of the rate-distortion cost; so, the weight associated to the edge connecting subband x to quantizer q is defined as $w_{x,q} = D_x(q) + R_x(q)$. For a E_N link, the associated weight is given by the cross-correlation distortion, i.e.: $w_{x_i,x_{i'}} = \langle \widehat{x}_i - x_i, \widehat{x}_{i'} - x_{i'} \rangle$, $i' \in \mathcal{N}(i)$. So the function we want to minimize can be written as:

$$\min \sum_i \left(\underbrace{\|x_i - \widehat{x}_i\|^2 + \lambda R(i)}_{E_{data}(i)} + \underbrace{\sum_{i' \in \mathcal{N}(i)} \langle \widehat{x}_i - x_i, \widehat{x}_{i'} - x_{i'} \rangle}_{E_{smooth(i)}} \right) \tag{5.16}$$

Now we establish the correspondence between our graph and the multiway cut. In Fig. 5.10 is illustrated an induced graph $G(\mathcal{E}_C) = (V, E - \mathcal{E}_C)$ corresponding to a multiway cut $\mathcal{E}_C$ on G. One can remark that it should be exactly one terminal link to each subband node in the induced graph.

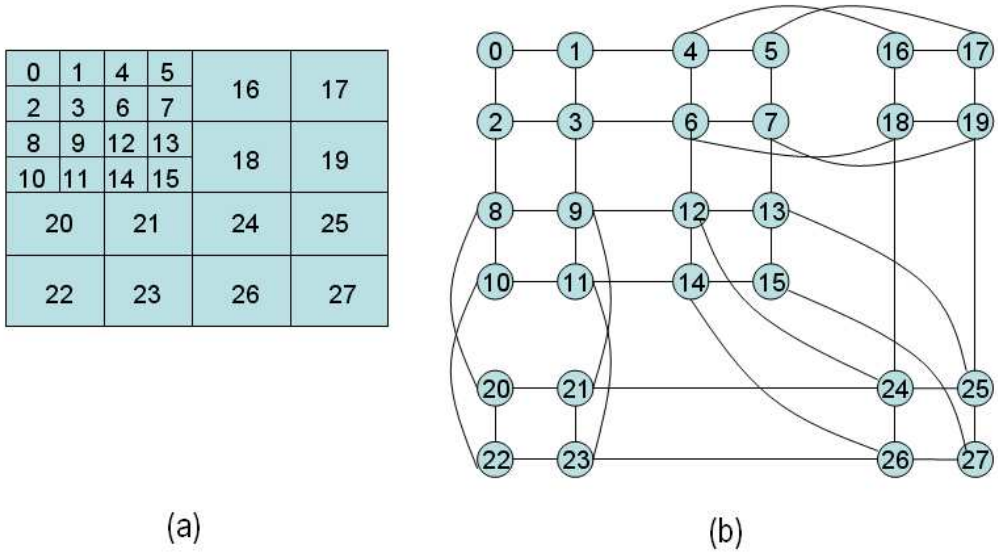


Figure 5.11: Block graph design: (a) two-level wavelet decomposition with four-blocks subband division and (b) regular vertices network design.

5.2.3.3 Graph design with cross-correlation distortion at the block level

In the following, we propose to extend the subband level distortion estimation presented in section 5.2.3.2 to the block level (Fig. 5.11). This extension comes naturally, as the smaller the coding unit, the more correlated in amplitude are the coefficients within it. At block level, Eq. (5.16) becomes:

$$\min \sum_{i=1}^{X} \sum_{j=1}^{N_b} \underbrace{\|x_{i,j} - \widehat{x}_{i,j}\|^2 + \lambda R(i,j)}_{E_{data}(i,j)} + \underbrace{\sum_{i' \in \mathcal{N}(i)} \left\langle \widehat{x}_{i,j} - x_{i,j}, \widehat{x}_{i',j} - x_{i',j} \right\rangle + \sum_{j' \in \mathcal{N}(j)} \left\langle \widehat{x}_{i,j} - x_{i,j}, \widehat{x}_{i,j'} - x_{i,j'} \right\rangle}_{E_{smooth}(i,j)} \quad (5.17)$$

where X, respectively N_b represents the number of subbands, respectively blocks in each subband, $x_{i,j}$ denotes the image reconstucted only from the j^{th} block of the i^{th} subband, $\left\langle \widehat{x}_{i,j} - x_{i,j}, \widehat{x}_{i',j} - x_{i',j} \right\rangle$ represents the cross-correlation distortion induced by the block j in neighbour subbands $i' \in \mathcal{N}(i)$ and $\left\langle \widehat{x}_{i,j} - x_{i,j}, \widehat{x}_{i,j'} - x_{i,j'} \right\rangle$ measures the correlation between neighbour blocks in a given subband i, $j' \in \mathcal{N}(j)$.

As expected, our graph will have this time $B = X \times N_b$ regular vertices. The neighbourhood system, $\mathcal{N}$, contains now both multiresolution correlation links E_{N_M}(i.e., edges between same positioned blocks in neighbour subbands as defined in section 5.2.3.2) as well as position correlation links E_{N_G}(i.e., edges between neighbour blocks in a 4-connected subband grid). The geometrical model can be described as: $G = (V, E)$ where $V = B \cup Q$ and $E = E_N \cup E_Q$, $E_N = E_{N_M} \cup E_{N_G}$ and Q/E_Q represent the quantiz-

ers set/the links between block nodes and quantizers. For the terminal links, E_Q, the weights are given by the direct costs in terms of distortion and rate induced by the quantization (i.e., the edge between block b and quantizer q, (b, q), has the associated weight $w_{b,q} = D_b(q)+R_b(q)$). The capacity between two regular neighbour blocks ($(b_i, b_{i'}) \in E_{N_M}$ or $(b_j, b_{j'}) \in E_{N_G}$) is defined as the cross-correlation distortion induced by the current quantization of these blocks.

5.2.4 Application to subband image compression

In the following, we propose to apply the proposed graph-cut minimization models to subband image compression. Some results are drawn in the framework of classical separable wavelet image coding, as well as for a geometrical transform, namely the contourlet decomposition [62]. Note that the method can be applied to almost any existing decomposition (subbands, blocks, critically sampled / redundant etc.).

5.2.4.1 Wavelet subband image compression

Due to their energy compaction efficiency, the biorthogonal filter banks are the most used in image compression [8]. This is the reason for which we consider in our simulation framework both the $5/3$ and $9/7$ filter banks for the spatial decomposition.

Experimental results For our simulations, we have considered two representative test images: Barbara (512x512 pixels) and Mandrill (512x512 pixels), which have been selected for the difficulty to encode their texture characteristics.

We have used dead-zone scalar quantization, with $q \in \{2^0, \ldots, 2^{10}\}$. The dead-zone has twice the width of the other quantization intervals. All the images have been decomposed over five spatial levels with the floating-point $5/3$ and $9/7$ filter banks. Note that for rate estimation in the allocation algorithm we have used a simple (non-contextual) arithmetic coder [132], while JPEG2000 codec [8] uses a highly optimized contextual coder.

As it can be remarked from Fig. 5.12 and Fig. 5.13, the results obtained with the $9/7$ wavelet subband decompostion of JPEG2000 are between 0.5 and 1.5 dB higher than those obtained with the proposed graph-cut rate-distortion algorithm. This situation can be explained by the fact that the $9/7$ filter bank is very close, from an energy partition point of view, to an orthonormal decomposition while the $5/3$ filter bank is quite far from this situation. As illustrated in Fig. 5.14 and Fig. 5.15, our method seems to better cope with such non-orthogonal decompositions than the classical weighting, based on the synthesis filter-bank characteristics, performed in JPEG2000.

One can remark that distortion approximation at subband level taking into account the cross-correlation among subbands always leads to better results than the simple model without cross-correlation terms, by using a more realistic correlation model. Moreover, the finer level of represention for the coding units, the higher the correlation among these units, as it can be remarked from stated results, having an average gain of 0.25 dB over the preceeding rate-distortion curve obtained with a subband-based cross-correlated distortion model.

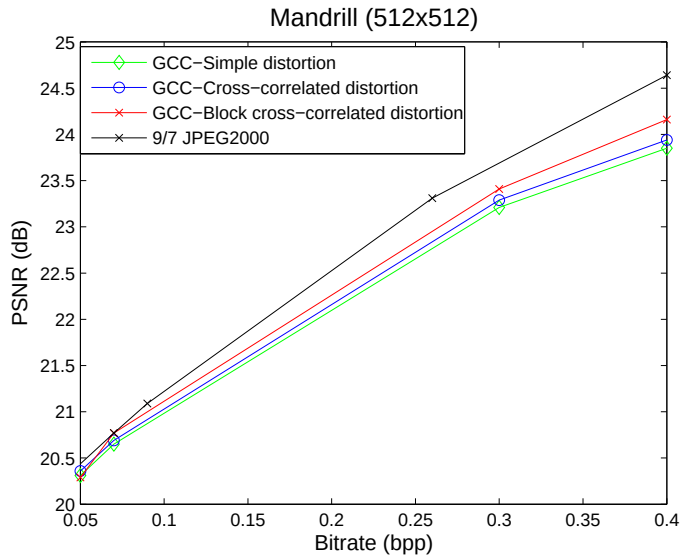


Figure 5.12: Rate-distortion comparison for Mandrill image with 9/7 wavelet subband decomposition

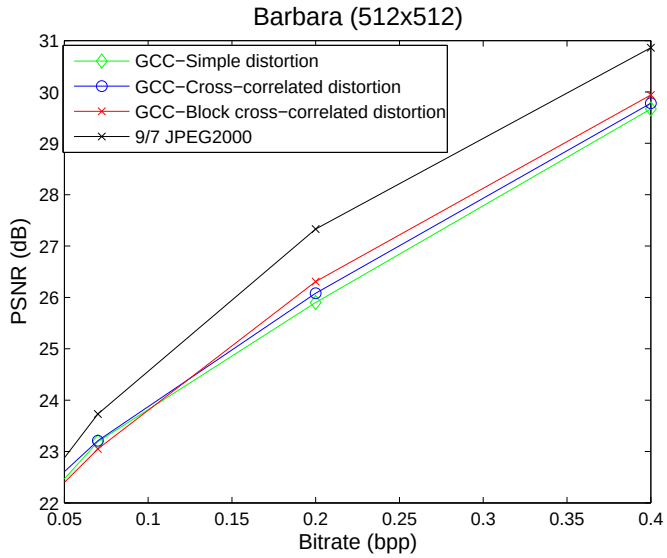


Figure 5.13: Rate-distortion comparison for Barbara image with 9/7 wavelet subband decomposition

5.2.4.2 Contourlet subband image compression

The drawback of separable wavelets is the limited orientation selectivity, as they fail to capture the geometry of the image edges. In order to overcome the problem of edge representation, Minh N. Do and Martin Vetterli have defined a new family of geometrical wavelets, called contourlets [62]. With contourlets, one can represent the class of smooth images with discontinuities along smooth curves in a very efficient and sparse way. The theory of geometrical wavelets has progressed in many directions, giving the definitions

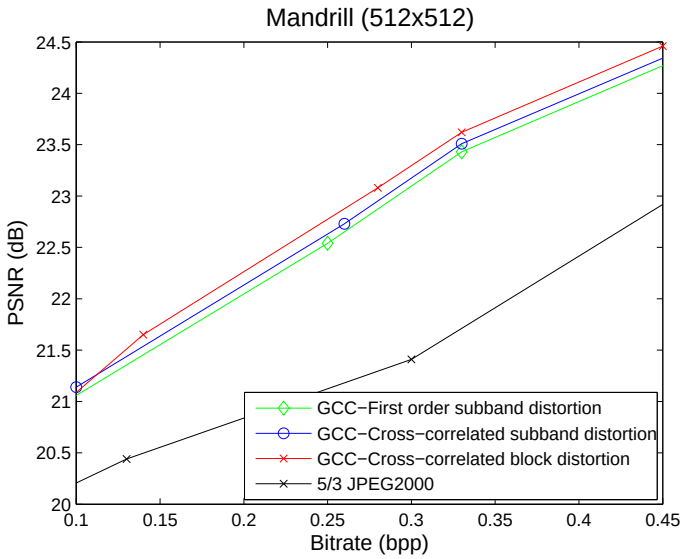


Figure 5.14: Rate-distortion comparison for Mandrill image with 5/3 wavelet subband decomposition

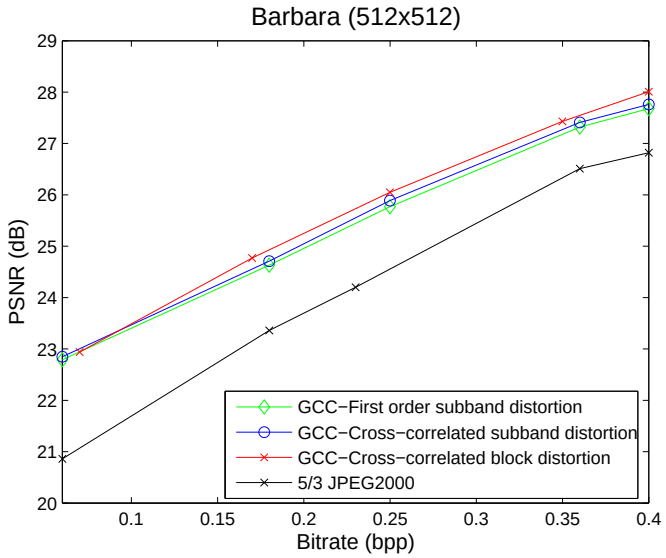


Figure 5.15: Rate-distortion comparison for Barbara image with 5/3 wavelet subband decomposition

of wedgelets [64], beamlets [96], curvelets [63], directionlets [200] and others [48, 151], as well as their corresponding fast transforms. All these new decompositions have been successfully used in image segmentation and noise removal, as well as in image compression: as shown in [203], the codec based on wedgelets gives better performance in image compression than the JPEG2000 standard at very low rate.

The contourlet transform [62] preserves the interesting features of classical wavelets, namely multiresolution and local characteristics of the signal, and, at the expense of a spatial redundancy, it better represents the directional features of the image. As shown

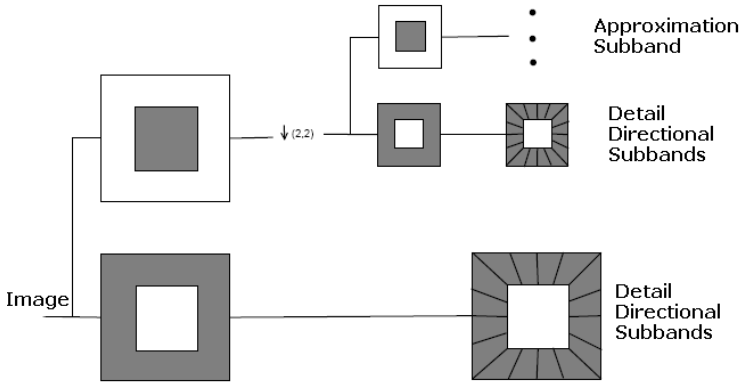


Figure 5.16: Contourlet filter bank

in Fig. 5.16, the transform is a multiscale and directional decomposition using a combination of a Laplacian pyramid (LP) and a directional filter bank (DFB). Bandpass images from the LP are passed to a DFB so that directional information can be retrieved. As its redundancy is given only by the LP transform, it has an upper limit on the redundancy of $4/3$, which makes the scheme more appropriate for compression than other geometrical transforms. Another reason for which we have considered this scheme is that contourlets can be approximated with fewer coefficients than wavelets; that is, for a contourlet basis, the approximation error for keeping only the M most significant coefficients is:

$$\|f - f_{M_{contourlet}}\| = O((logM)^3 M^{-2}) \tag{5.18}$$

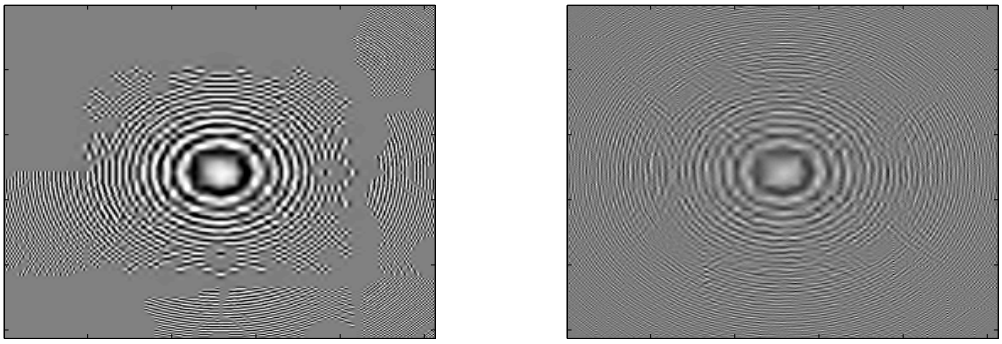
which is smaller than that obtained on a separable wavelet basis:

$$\|f - f_{M_{wavelet}}\| = O(M^{-1}). \tag{5.19}$$

As shown in [37], the efficiency of the pyramidal directional filterbank (PDBF) with respect to classical wavelets tends to decrease on natural images when the number of coefficients increases. Because at low decomposition levels a better energy compaction is needed, we have decided to use in our simulation framework a hybrid approach for spatial decorrelation.

Experimental results For our simulations, we have considered four representative test images: Zoneplate (512x512 pixels), Circles (512x512 pixels), Barbara (512x512 pixels) and Mandrill (512x512 pixels), which have been selected for the difficulty to encode their texture characteristics. We have used dead-zone scalar quantization, with $q \in \{2^0, \ldots, 2^{10}\}$ and a 5-level contourlet where the coarsest three decomposition levels consist of a 9/7 separable wavelet transform (i.e., 3 directions), and the finest two levels are represented with a 16- and 32-band biorthogonal directional filter. The efficiency of this hybrid scheme has been proved in [37] and in [25].

Fig. 5.17 presents the Zoneplate image compressed at 0.2 bpp with JPEG2000 and the graph-cut allocation algorithm using the simple distortion approach coupled with contourlet decomposition. One can remark that both numerical and visual quality are improved. For the same transmission rate, our method surpasses JPEG2000 by more than 1.5 dB, even though it employs a redundant transform. Similar results are also depicted in Fig. 5.18 for the Mandrill image.



(a) (b)

Figure 5.17: Zoneplate (512x512) image compressed at 0.2 bpp: (a) JPEG2000 compression (PSNR = 11.12dB), (b) 1^{st} graph-cut rate allocation method (PSNR = 12.66dB).

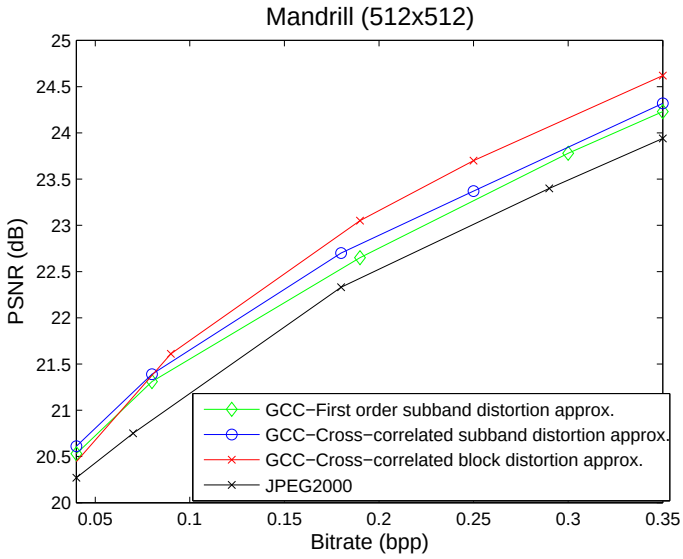


Figure 5.18: Rate-distortion comparison for Mandrill image with contourlet subband decomposition

Fig. 5.21 illustrates the compression results obtained with the subband-based graph-cut allocation methods and JPEG2000 at 0.1 bpp. Our allocation is based on the contourlet subband decomposition. One can note better visual quality results of both graph-cut compression methods with respect to JPEG2000. Moreover, a 0.5 dB PSNR improvement due to the cross-correlation distortion approach with respect to the simple distortion approach can be noticed. Note that, for rate estimation in the allocation algorithm, we have used a simple (non-contextual) arithmetic coder [132], while JPEG2000 codec uses highly optimized contextual coder.

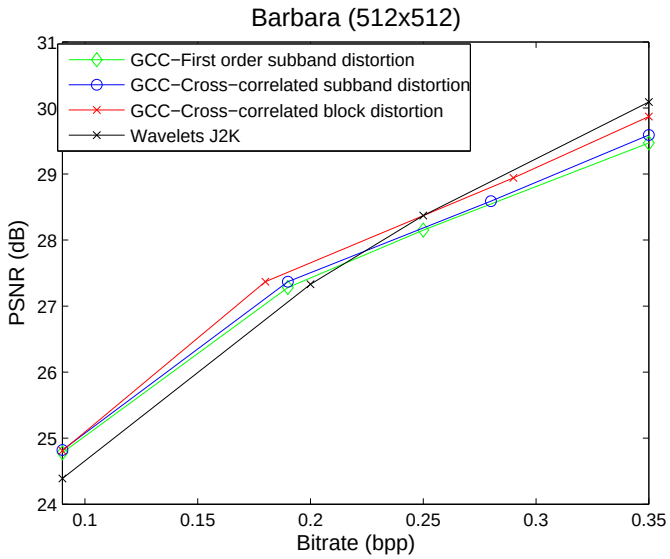


Figure 5.19: Rate-distortion comparison for Barbara image with contourlet subband decomposition

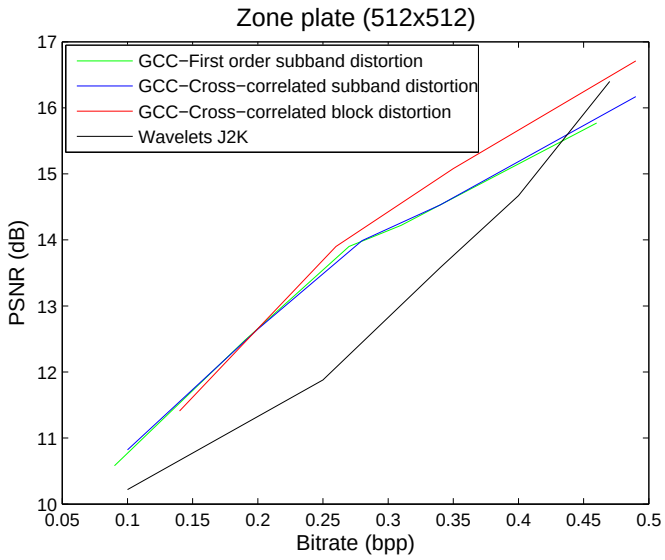
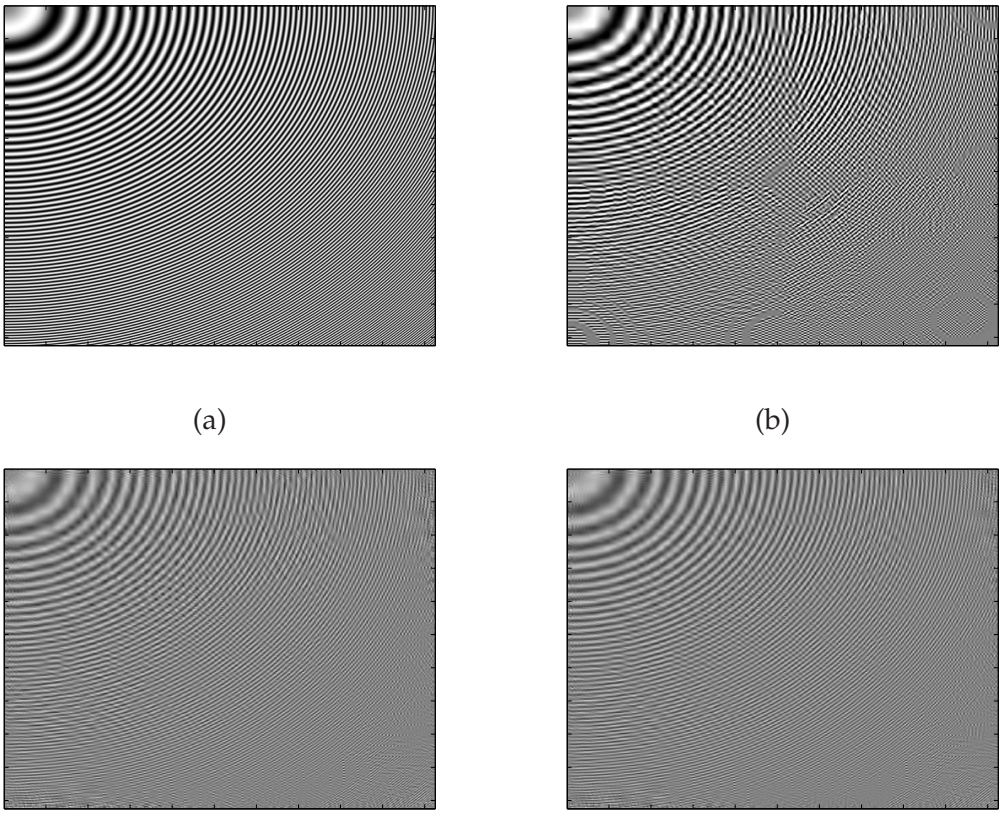


Figure 5.20: Rate-distortion comparison for Zone-plate image with contourlet subband decomposition

5.3 Conclusion

In this chapter we have presented a graph-cut method for rate-distortion optimization in image coding using decompositions which are not necessarily orthonormal. As shown by experimental results, it can efficiently encode contourlet coefficients at low bitrates, improving both the visual and numerical quality. Moreover, the proposed method can be further used with vector quantizers.



(a) (b)

(c) (d)

Figure 5.21: Circles (512x512) image at 0.1 bpp: (a) original, (b) JPEG2000 compression (PSNR=14.19 dB), (c) simple-distortion graph-cut method (PSNR=14.64 dB) , (d)cross-correlation distortion graph-cut method (PSNR=15.13 dB).

Conclusion and perspectives

This work contributes to the development of a $t + 2D$ wavelet-based video codec. More specifically, our research focused on the construction and optimization of MCTF schemes, the analysis of different spatial decomposition methods for better representing the temporal subbands in order to enhance both the objective and subjective quality of reconstructed sequences, and, last but not least, the improvement of entropy coding by designing graph-cut-solvable energy functionals for the rate-distortion-optimization problem. Below we summarize the contributions of the thesis work, and then we propose some directions for future research.

Synthesis of thesis contributions

I—Temporal video processing

Scene-cut processing in MCTF

Typical decompositions for temporal decorrelation are dyadic ones and use Haar and 5/3 filter banks. Generally, the same filtering is made along the sequence based on the assumption that the frames are highly correlated. However, this assumption no longer holds when the video involves scene changes, as in the case of action movies, music-video clips etc. The inefficiency of the motion estimator leads to poor predict/update stages, which, combined with the sliding-window implementation of the temporal filters, propagates the prediction/update errors through the decomposition levels.

We have proposed an improved version of the 5/3 MCTF coding scheme able to detect and process the scene-cuts appearing in video sequences. The lifting structure of the filters has been modified such that the filtering does not encompass the scene-cut. Moreover, the coding units were reduced near the scene-cut to accommodate this change. The experiments made in the framework of MC-EZBC video codec [210] show that our method gives an average YSNR gain of about 1.5 dB on several test video sequences and higher for frames close to the scene-cut.

A 5-band MCTF lifting scheme

Longer filters are preferred for temporal decomposition due to their efficiency in removing temporal redundancy. However, when the filters are too long, it is very likely that they will encompass several (different) scenes and, thus, loose their decorrelation efficiency. Moreover, the longer the temporal filter, the fewer the temporal decomposition levels needed to obtain some key (approximation) frames so useful for video database search and storage.

We have thus introduced a 5-band temporal-lifting structure allowing flexible scalability factors by multiples of five in a MCTF video codec. For the proposed structure, we have developed two implementation schemes: one using the mirroring method (so that the filter bank does not span other GOPs) and the other one using a sliding window. Both 5-band schemes have been implemented and integrated into the framework of the MSRA 3D-wavelet video codec [212]. The proposed method has similar performance to that of the dyadic Haar and 5/3 filters and the non-linear 3-band decomposition scheme. Also, it gives better coding efficiency for the temporal approximation subbands leading then to an improved temporal scalability. It can be used successfully in certain applications, such as the encoding of video-surveillance sequences where the motion activity is weak in most cases and few approximation frames are needed for long-term storage.

LMS-based adaptive prediction for scalable video coding

In wavelet video coding, the most used method for motion estimation is block-based, and even with a bidirectional temporal prediction, block artefacts are still present. In order to avoid such artefacts, motion-compensation solutions, such as weighted average update operator or overlapped block motion compensation, have been proposed to alleviate this problem.

We have proposed an LMS-based adaptive prediction for the temporal prediction step in scalable video coding. Pixels of temporal detail subband frames are optimally predicted using a set of pixels from the neighbouring subband frames. We have illustrated our proposal on bidirectional prediction, but the set of pixels for adaptation can be chosen from any number of frames involved in a longer-term prediction. The proposed method has been developed in the framework of the MSRA 3D-wavelet based video coder. The experimental results show that, even for two-pixel adaptation, visual quality of the reconstructed frames is improved. A trade-off between compression efficiency and additional complexity coming from a larger adaptation window can be done according to the target application. Significant PSNR improvements have been obtained for sequences with high contrast between various segments within the sequence and more particularly for varying illumination conditions.

Video compression for multi-temporal and multispectral satellite sequences

MCTF coding efficiency is strongly related to the correlation of the data being processed. Based on this assumption, we tested $t + 2D$ video coding principles on multi-temporal and multispectral data sequences. In order to achieve data compression, coding techniques applied to multi-temporal and multispectral data take advantage of the presence of two redundancy sources: spatial correlation among neighboring pixels in the same spectral band and temporal correlation among different bands at the same spatial location.

We have thus proposed to evaluate the performance on SPOT (1, 2 and 4) sequences using still image (JPEG2000) and video (t + 2D) compression techniques based on wavelet tools. As it results from the experiments, EZBC intra-coding outperforms EBCOT in terms of rate gain in the case of nearly lossless compression. Also, it has been pointed out that lossless compression provides comparable results with lossy compression realized in the manner described above. In addition, it has been highlighted that temporal and spectral decorrelations can be exploited separately or jointly to improve the compression

ratio. As expected, the combination of spectral and temporal decorrelation provides the best results in terms of bitrate reduction.

II—Spatial processing of video sequences

Joint wavelet packets

The results of a study regarding the spectral properties of the subbands obtained by the wavelet temporal decomposition have shown that, contrary to the approximation subbands, frequency information of temporal details is distributed almost uniformly over the subbands, that is, far from being localized in the low frequencies. These differences suggest that dyadic wavelets, which are powerful for spatial encoding of still images or temporal approximation subbands, would not be the best decorrelation scheme for the temporal detail subbands. These observations have motivated us to investigate the use of wavelet packets, whose frequency selectivity properties are more suited to decompose the detail frames.

We have thus proposed a method for building a *joint* wavelet-packet representation for several frames. A best-basis selection was adapted to this goal and the method has been illustrated by simulation results in the framework of MCTF coding for the entire GOPs and for the detail frames at a given temporal level. The method has been implemented in the framework of the MSRA 3D-wavelet video codec, the best results being obtained when a single wavelet-packet basis was considered for coding all the detail frames in a group of frames. We have also proposed and implemented the algorithmic modifications for the selection of the best basis when biorthogonal bases are used.

Fully separable wavelets and wavelet packets

Finding the best wavelet-packet representation, even for groups of frames, can be a computationally expensive task. This remark has led us to look for a simpler spatial decomposition of the temporal subbands.

We have thus presented an evaluation of fully separable wavelet and wavelet-packet transforms for texture encoding in a motion-compensated subband video codec. The finer 2D frequency separation given by the fully separable transforms allows better capture of the orientation of the spatial details, resulting in better representation of the video texture in comparison to classical quadtree decompositions.

III—Rate-distortion optimization using graph-cuts

Many computer-vision problems can be formulated in terms of energy minimization. On one hand, in the last few years, minimum-cut/maximum-network-flow algorithms have emerged as an elegant and increasingly useful tool for exacting or approximating energy-minimization problems. On the other hand, the majority of compression algorithms that use wavelet transforms try to exploit all the signal redundancy that can appear inside, and, across the different subbands of a spatial decomposition. However, the efficiency of a coding scheme highly depends on rate allocation.

We have thus designed a graph-cut-solvable energy functional for Lagrangian rate-distortion optimization for subband coding of non-orthogonal systems. Moreover, we have presented three possible solutions by modeling several aspects of energy interactions for the minimization of a non-orthogonal functional. The presented method has

good coding efficiency especially at low bitrates, improving both the visual and numerical quality of the reconstructed images.

Perspectives

In this thesis work, we have studied and proposed several spatio-temporal operators able to give a parcimonious multiresolution representation of the video sequences. Nevertheless, a number of topics can be identified that still require further investigation, and may lead to even better compression performance for the $t + 2D$ class of video-coding algorithms. These include:

- ⋆ Adaptive temporal operators, such as adaptive update step, where the Human Visual System is used to evaluate the impact in terms of visual quality at the low-pass subbands, since, in MCTF, the predicted residue is further used to update the temporal low-pass frames and may cause annoying ghost artefacts if the predicted residues are generated by inaccurate motion prediction.

- ⋆ Application and optimization of non-block-based motion compensation for adaptive temporal-lifting schemes. A pel-recursive motion-estimation algorithm can be used together with the LMS-based adaptive temporal prediction to obtain a more coherent motion flow at pel-level and thus reduce more of the blocking artefacts. The LMS-adaptive prediction can also be combined with revertible schemes optimizing the update step, such as the weighted average update operator or the OBMC.

- ⋆ An update-first method should be considered for the LMS-based prediction strategy. Because the adaptive step takes place at prediction, and the high-pass subbands are further used in the lifting scheme for reinforcing the signal in the low-pass bands, the eventual prediction errors can pass easily from one decomposition level to another. By inversing the lifting-scheme steps, the prediction errors do not propagate through the decomposition tree.

Additionally, the common description which characterizes the spatio-temporal features of a video GOP given by the proposed joint wavelet-packet transforms can be exploited as a valuable feature for video classification and video-database searching. Also, the fewer temporal decomposition levels needed by the 5-band filter bank for obtaining some high-quality key (approximation) frames are very useful for fast video-database search and storage, and together with the joint description of the temporal subbands provided in the proposed joint wavelet-packet scheme, could be seen as a starting point for a fast indexing system for video sequences.

Moreover, the efficiency of the 5-band motion compensated lifting scheme can be substantially improved if the scene-cut detection and processing algorithm is jointly considered in the temporal decorrelation stage, as the longer the temporal filter, the greater the probability of encompassing several uncorrelated video shots.

As the proposed graph-cut rate-distortion optimization gives promising results for the coding of non-orthogonal wavelet decompositions, it could be used jointly with existing coding algorithms for rate allocation between spatio-temporal subbands in a video codec. Moreover, the rate estimation method could be replaced with a higher-order

entropy-based algorithm or a real (dummy) estimation of the rate, so an improvement of the results for orthogonal systems can be expected.

Nevertheless, graph-cut-based algorithms can be conceivable for the optimization of the motion-vector fields, where the best motion vector is obtained by the minimization in terms of distortion and rate (as classically done), and also by the neighbourhood coherence which could be given by: the distance between adjacent motion vectors, the distortion introduced by blocking artefacts or, moreover, the rate difference needed to encode the motion vector.

Appendix A

Joint Source-Channel Coding

This annexe presents the thesis work on joint source-channel coding described in two journal papers. A robust joint source-channel coding scheme for transmission of video sequences over Gaussian channels using uncoded and coded index assignment via Reed-Muller is decribed in a first part [70], continued by the presentation of a coding system designed for the video transmission over flat Rayleigh fading channels [69].

A.1 Joint Source - Channel Coding with Partially Coded Index Assignment

A.1.1 Introduction

Shannon's separability theorem is often used to justify the independent design of source- and channel-coding subsystems. However, in real-time video systems, the separability principle may not be applicable due to the high complexity for both the source and channel coders potentially entailed by the theorem. Consequently, there has been increasing interest in joint source-channel coding (JSCC) to provide efficient performance with complexity lower than tandem schemes.

Many prior JSCC techniques can be partitioned into two main categories: 1) *source-optimized channel coding*, wherein channel coding is optimized with respect to the source; and 2) *channel-optimized source coding*, wherein source coding is optimized with respect to the channel. In source-optimized channel coding, a quantizer—most generally, a vector quantizer—is designed for a noiseless channel. In the absence of explicit channel coding, vector quantization (VQ) can be made robust by applying a good index assignment (IA) to map quantization indices to channel codewords so as to minimize the impact of channel noise (e.g., [153]). On the other hand, when an explicit channel coding is used, careful attention is paid to optimally partition given resources between the source and channel coder (e.g., [35, 169, 43, 91]). In channel-optimized source coding, the VQ and IA are simultaneously optimized for a specific channel such that very efficient clean-channel performance is obtained while providing robustness in the presence of noise (e.g., [77]).

In both the source-optimized channel coding and channel-optimized source coding categories of JSCC, the traditional approach is to cascade the channel code after the source code, such that the channel code adds redundancy to the transmission to combat channel errors and effectively increases the end-to-end transmission rate. An alternative category of JSCC, which can be considered to be *channel-constrained source coding*, was introduced in [171]. In such an approach, VQ is trained for minimum quantization distortion under constraints arising from the channel. The main result is that the channel distortion of a binary symmetric channel (BSC) is minimized if the source codebook can be expressed as a linear transform [105], that is, if the IA labeling is linear. Such linear IA includes direct mapping of VQ indices to channel codewords as well as coded IA wherein the VQ indices are mapped through a channel code. The use of the channel code in this latter approach effectively constrains the VQ source codewords to reside in the space of channel codewords. This marks a substantial departure from the traditional use of channel coding to add redundancy—and, consequently, increased transmission rate—as is the case in schemes that concatenate source and channel coding (e.g., [35, 169, 43, 91]).

In [26], linear transforms constructed from lattice constellations with "maximum component diversity" were used to build structured VQ codebooks which minimized simultaneously the source and channel distortions for Gaussian sources. In this paper, we develop a JSCC scheme in the channel-constrained source-coding category for the coding of video wherein the source distribution is not Gaussian. Specifically, we describe a scalable video-coding system constructed from $t + 2D$ motion-compensated temporal filtering (MCTF) coupled with JSCC using the structured VQ of [26]. The VQ indices are mapped to channel codewords either directly in an uncoded form, or through coded IA based on Reed-Muller codes, with the encoder adaptively deciding between the coded and uncoded IA on a subband-by-subband basis. Consequently, with coded

IA, the source codewords themselves are constrained to belong to the channel code, and there is no rate increase due to the incorporation of the channel code. We compare our proposed coding scheme to a source-optimized channel-coding technique featuring unstructured VQ of MCTF coefficients coupled with the IA mapping of [153], as well as to the more traditional approach to error resilience consisting of concatenating a source coder (the prominent MCTF-based coder MC-EZBC [39]) with a channel coder (convolutional codes). We find that the proposed JSCC system consistently outperforms the other two schemes as the channel noise level increases.

A.1.2 Index Assignment for Gaussian Sources

Let vector $\boldsymbol{x}$ be the input to a vector quantizer which produces an n-bit binary codeword, the quantization index of the vector. The source codebook can then be viewed as a function of $\boldsymbol{b} = \begin{bmatrix} b_1 & \cdots & b_n \end{bmatrix}^T \in \{+1, -1\}^n$, where $\boldsymbol{b}$ represents the IA of $\boldsymbol{x}$. Under the assumption of a maxentropic quantizer, the total distortion is $D = D_s + D_c$, where D_s is the source distortion due to quantization, and D_c is the channel distortion dependent on the IA.

In [105], the channel distortion of a BSC is proved to be minimized by IA in the form of a linear labeling, while, in [26], a linear labeling that minimizes simultaneously the source and channel distortions is constructed. In the case of a zero-mean Gaussian source, this linear labeling is constructed using a subset of lattice constellations with "maximum component diversity". Specifically, let $\boldsymbol{U}_n$ be an $n \times n$ generator matrix of a maximum-component-diversity lattice constellation as described in [81]. Its construction is based on number-field theory, and it is expressed by the standard embeddings in $\boldsymbol{R}^n$ of the ideal ring of the totally real subfield of cyclotomic fields. The rows and the columns of $\boldsymbol{U}_n$ are denoted by L_{in} and C_{nj}, respectively, where $1 \leq i, j \leq n$. If J is some subset of $\{1, \ldots, n\}$, then $C_{nj}(J)$ is the j^{th} column of $\boldsymbol{U}_n(J)$, which is a matrix of only the rows of $\boldsymbol{U}_n$ corresponding to the indices in J. Using $\boldsymbol{U}_n$, one can linearly map $\text{BPSK}_n = \{-1, +1\}^n$ onto a new set $\boldsymbol{U}_n \cdot \text{BPSK}_n$. Allowing n to increase while J remains fixed, we get a codebook $S_n(J)$ with codewords $\boldsymbol{y}^{(l)}$, $\boldsymbol{y}^{(l)} = \sum_{j=1}^{n} b_j^{(l)} C_{nj}(J)$, where $\boldsymbol{b}^{(l)} = \begin{bmatrix} b_1^{(l)} & \cdots & b_n^{(l)} \end{bmatrix}^T \in \text{BPSK}_n$, and $1 \leq l \leq 2^n$. In order to obtain a family of matrices $\boldsymbol{U}_n$ such that $S_n(J)$ is an asymptotically Gaussian source dictionary that minimizes D_s as $n \to \infty$, $\boldsymbol{U}_n$ must be orthogonal with coefficients going uniformly to 0 as $n \to \infty$ [26]. In this case, the linear mapping $\boldsymbol{b} \in \text{BPSK}_n \to \left(\boldsymbol{G}_{d,n}\boldsymbol{b} \in S_n(J)\right)$, where $d \times n$ matrix $\boldsymbol{G}_{d,n} = \boldsymbol{U}_n(J)$, $d = |J|$, allows the construction of a source dictionary that is asymptotically Gaussian. Similar properties are achieved by selecting columns of the matrix $\boldsymbol{U}_n$, and we shall denote the $n \times r$ matrices constructed this way as $\boldsymbol{G}'_{n,r}$, where $r \leq n$.

The above discussion assumes that the uncoded IA $\boldsymbol{b}$ is transmitted directly on the channel. In the alternative case that an error-correcting code is used, $\boldsymbol{b}$ ranges in $\boldsymbol{m}(\boldsymbol{c})$ where $\boldsymbol{c}$ is one of the 2^k possible binary codewords belonging to the (n, k) linear code $\mathcal{C}$. The function $\boldsymbol{m}(\cdot)$ maps $\boldsymbol{c} = \begin{bmatrix} c_1 & \cdots & c_n \end{bmatrix}^T$ onto $\boldsymbol{m}(\boldsymbol{c}) = \begin{bmatrix} m(c_1) & \cdots & m(c_n) \end{bmatrix}^T$, where $m(0) = 1$ and $m(1) = -1$. The codebook for this coded case has codevectors $\boldsymbol{y}^{(l)}$ given by $\boldsymbol{y}^{(l)} = \boldsymbol{G}_{d,n}\boldsymbol{b}^{(l)}$, where $\boldsymbol{b}^{(l)} = \boldsymbol{m}(\boldsymbol{c}^{(l)})$, $\boldsymbol{c}^{(l)} \in \mathcal{C}$, and $1 \leq l \leq 2^k$.

A.1.3 Coding of Spatio-Temporal Subbands

We now apply the JSCC scheme described above to a scalable video coder. The resulting system first applies $t + 2D$ MCTF in the form of a motion-compensated temporal wavelet transform applied to a group of frames (GOF) followed by a spatial wavelet transform of the temporal subbands. Next, an optimal bit-allocation procedure allocates rate between the spatio-temporal subbands, after which the spatio-temporal coefficients are vector quantized. Finally, a linear IA mapping between the source codebook and the coded symbols sent on the channel is applied to provide resilience to channel noise.

A.1.3.1 Index Assignment for non-Gaussian Sources

Due to the fact the coefficients of the $t+2D$ MCTF subbands are not Gaussian, the coding scheme of Sec. A.1.2 cannot be applied directly. However, the marginal distribution of the subband coefficients has been shown to be well-modeled by a mixture of two Gaussians [73]; thus, we classify vectors drawn from the spatio-temporal subbands into two vector classes and apply the IA approach of Sec. A.1.2 to each class independently.

For vectors from the temporally lowpass (approximation) frames, it was observed in [73] that classification according to vector magnitude, such that the vectors are partitioned into a low-variance and a high-variance class, results in an approximately Gaussian distribution within each class. Similarly, vectors from the temporal highpass (detail) frames are classified into two classes using the stochastic model of spatio-temporal dependencies introduced in [73]. This permits accurate classification based on only the coefficients already decoded, without requiring transmission of side information. Following this model, we assume that the conditional probability of a coefficient is Gaussian with variance depending on a set of its spatio-temporal neighbors; i.e., the conditional probability of coefficient x is $f(x|\sigma_x^2) = \frac{1}{\sqrt{2\pi}\sigma_x} \exp\left(-\frac{x^2}{2\sigma_x^2}\right)$, where the variance is $\sigma_x^2 = \sum_i w_i \left|p_i(x)\right|^2 + \alpha$, such the $p_i(x)$ are coefficients neighboring x in the same spatio-temporal subband, w_i are weight parameters, and α is an offset parameter. The spatio-temporal neighborhood is a set of causal coefficients that will have already been received by the decoder when the current coefficient is decoded. Estimation of the parameters (w_i and α) of this model is done as in [73].

A.1.3.2 Quantization and Bit Allocation

For each vector class described above, we design a VQ codebook by minimizing cost $\gamma = \mathrm{E}\left[\min_l \left\|\boldsymbol{x} - \beta \boldsymbol{G}_{d,n} \boldsymbol{b}^{(l)}\right\|^2\right]$, where, in the case that the IA is uncoded, $\boldsymbol{b}^{(l)}$ ranges over the set of 2^n possible codewords of a BPSK_n, and, in the case that the IA includes an error-correcting code, $\boldsymbol{b}^{(l)}$ ranges over the set of 2^k possible codewords of an (n, k) code $\mathcal{C}$. β is a parameter which scales the lattice constellation $\boldsymbol{G}_{d,n}$ to the source dynamics. In order to find β, as well as the codebook with vectors $\boldsymbol{y}^{(l)} = \beta \boldsymbol{G}_{d,n} \boldsymbol{b}^{(l)}$, an iterative optimization algorithm (similar to that of shape-gain VQ) is used. A similar optimization is applied when using the matrix $\boldsymbol{G}'_{n,r}$ for VQ; in this case, an (r, k) code $\mathcal{C}'$ is used for the coded IA.

The channel distortion D_c is minimized due to the linearity of the IA labeling, and its value is fixed for a given channel-noise variance. Consequently, an iterative bit-allocation algorithm is applied to allocate VQ rate between the spatio-temporal subbands in an optimal fashion. This bit-allocation algorithm, which originates in [72], takes into account

a nonnegativity constraint on the rate allocated to each subband. The algorithm indicates the size of the $\boldsymbol{G}_{d,n}$ or $\boldsymbol{G}'_{r,n}$ matrix which minimizes the end-to-end distortion. The choices of $\boldsymbol{G}_{d,n}$ or $\boldsymbol{G}'_{r,n}$ are, however, limited in practice by computational-complexity issues and dependences between the spatio-temporal coefficients. That is, it is known that the spatio-temporal coefficients exhibit strong correlation with their spatial or spatio-temporal neighbors; thus, in order to exploit these relationships, the dimensions d in $\boldsymbol{G}_{d,n}$ or n in $\boldsymbol{G}'_{n,r}$ should be a power of 4. However, in order to attain high or low coding rates, we permit these values to be 2 if need be. In addition,to keep the complexity low, we limit the dimensions n in $\boldsymbol{G}_{d,n}$ and r in $\boldsymbol{G}'_{n,r}$ to be no greater than 16.

A.1.3.3 Partially Coded Index Assignment

We initially applied the VQ and IA described above in an uncoded fashion, i.e., without the use of any error-correcting codes in the IA mapping. However, when transmitting over a Gaussian channel with low SNR, we remarked that for some subbands, especially those with high energy, the total distortion D was very high compared to the source distortion D_s obtained for when the channel was noiseless. We conclude that, in this situation, the channel distortion D_c must be dominant. In order to improve performance, we replace the uncoded IA with coded IA incorporating an error-correcting code for these subbands. We choose Reed-Muller codes due to their symmetry, their widespread use in lattice construction, and their error-correcting capability.

For coded IA, we restrict the mapping space to be the space of the binary vectors belonging to the Reed-Muller code. We additionally constrain the source-coding rate to be the same rate as dictated by the bit-allocation algorithm in the uncoded case. We then choose the (η, k) Reed-Muller code in light of the trade-off between the following considerations: 1) the error-correction capability of the code; 2) the blocklength η of the code must be $\eta = n$ of $\boldsymbol{G}_{d,n}$, or $\eta = r$ of $\boldsymbol{G}'_{n,r}$, as appropriate, in order that the bitrate does not increase; and 3) the dimension of the code k should be close to n or r so that the number of 2^k possible codewords is close to the 2^n or 2^r possible codewords of the uncoded case, in order to minimize the increase to the source distortion. In this way, the end-to-end distortion decreases without changing either the source-coding rate of the uncoded case or the total bitrate.

The encoding algorithm consists of the following steps. In each subband, we calculate D_s in a noiseless environment as well as the end-to-end distortion D for the given noisy channel as it would be obtained with an uncoded IA. If the difference between D and D_s is high (which means that D_c is significant), we restrict the IA to be codewords of a Reed-Muller code selected in regard to the considerations discussed above. Otherwise, the IA maps directly to uncoded codewords. At the decoder side, soft-decision decoding via the Viterbi algorithm with the BCJR trellis [129] is applied to the coded codewords, while hard-decision decoding is applied to the uncoded codewords. We note that the encoder sends a small amount of side information to the decoder (β and the sizes of $\boldsymbol{G}_{d,n}$ and $\boldsymbol{G}'_{n,r}$ for each vector class for each subband, as well as the coded/uncoded state for each subband); it is assumed that this side information is highly protected so as to arrive at the decoder uncorrupted.

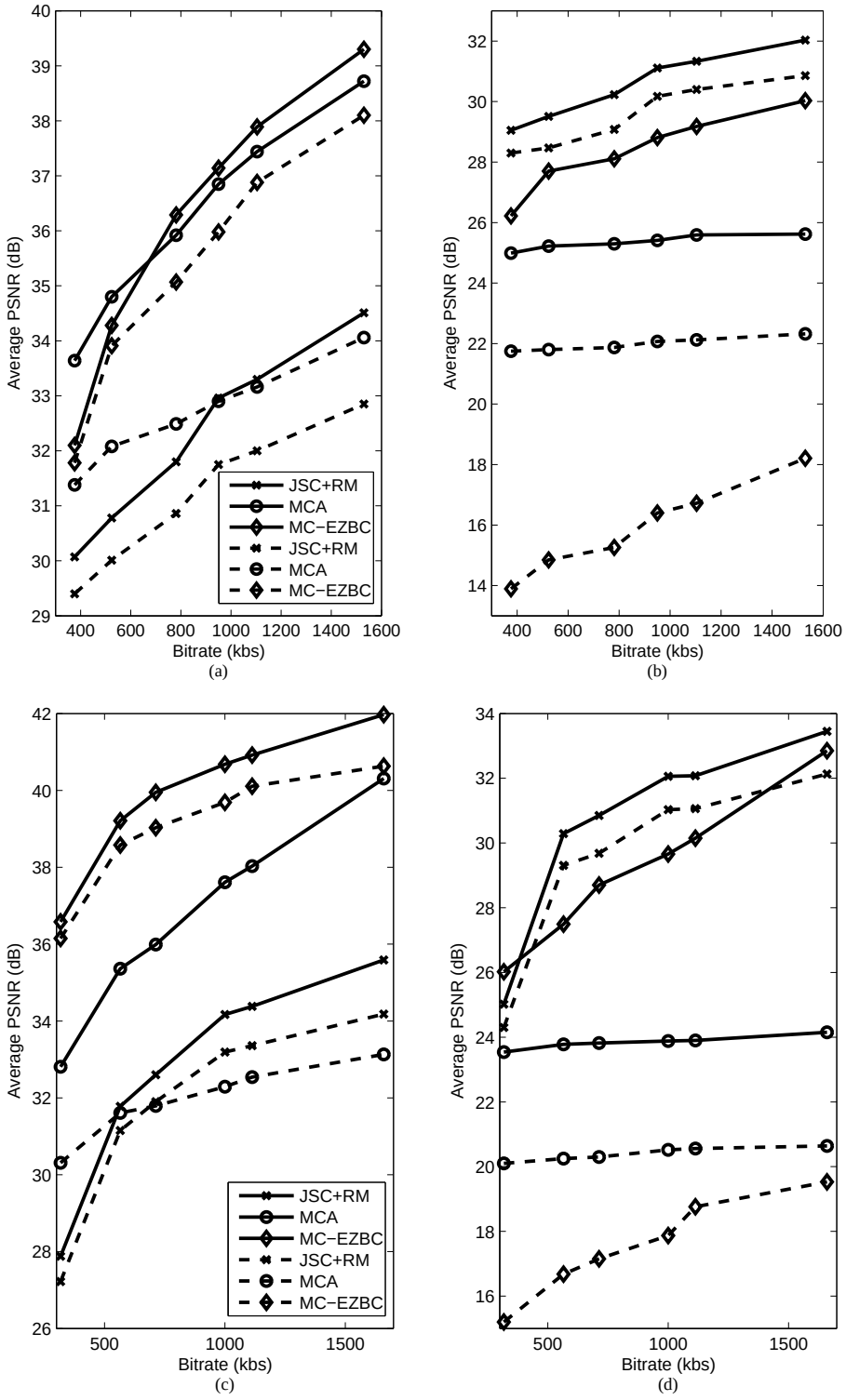


Figure A.1: (a) Foreman, channel SNRs of ∞ (solid lines) and 6.75 dB (dashed lines); (b) Foreman, channel SNRs of 4.33 dB (solid lines) and 3.00 dB (dashed lines); (c) Hall Monitor, channel SNRs of ∞ (solid lines) and 6.75 dB (dashed lines); (d) Hall Monitor, channel SNRs of 4.33 dB (solid lines) and 3.00 dB (dashed lines).

A.1.4 Experimental results

To experimentally evaluate the effectiveness of the system described in the previous section, we perform simulations using CIF test sequences at 30 fps. The video-coding system uses a Haar MCTF decomposition applied on GOFs of 16 frames, with 4 temporal and 2 spatial resolution levels. The spatial transform uses the popular biorthogonal $9/7$ filters.

As we remarked in Sec. A.1.3.3, a Reed-Muller code is incorporated into the IA for subbands of high energy. Our bit-allocation algorithm dictates that these high-energy subbands are coded using $\boldsymbol{G}_{d,16}$ or $\boldsymbol{G}'_{n,16}$. Thus, in consideration of the three trade-offs discussed in Sec. A.1.3.3, we choose the $\mathrm{RM}\{2,4\}$ with $\eta = 16$ and $k = 11$. However, compared to the uncoded case, we expect the source distortion to increase, as now the dimension of the mapping space has decreased to 2^{11} possible codewords, instead of 2^{16} as in the uncoded case. On the other hand, the end-to-end distortion of the entire scheme in a noisy environment will decrease significantly compared to the uncoded case, and the total bitrate remains the same.

We compare our proposed scheme (which we denote as "JSC+RM") to two other video coders using the same scalable MCTF transform structure. The first technique belongs to the class of source-optimized channel coding. In this scheme, a VQ source coder is designed to minimize source distortion for the noiseless channel, while a good, albeit suboptimal, IA is applied to increase the error resiliency of the quantizer. For the VQ source coder, we apply the locally optimal generalized Lloyd algorithm (GLA) to produce unstructured VQ codebooks with locally minimal source distortion D_s. The GLA VQ codebooks are of the same dimensions as dictated by the bit-allocation algorithm of our proposed scheme.We then follow with the Minimax Cover Algorithm (MCA) [153], which is an IA using a minimax error criterion that is designed against worst-case performance without sacrificing average performance. We refer to this coder as "MCA."

The second video coder to which we compare corresponds to an implementation of MCTF concatenated with traditional error-control coding. We employ the prominent MC-EZBC [39] coder, and, to provide error resilience, we packetize the MC-EZBC bitstream while applying rate punctured convolutional (RPC) codes to the resulting packets. Specifically, each packet contains the information corresponding to a single spatial resolution level from a single temporal subband frame. Hence, the packets have unequal length and are coded unequally by RPC codes. If the decoder fails to decode a received packet, the packet is dropped. The RPC codes have $R_p = 2/3, 3/4$, and $7/8$ with memory $m = 6$ and mother code $R = 1/2$ [88]. The most important information is protected by $2/3$ codes, the medium spatio-temporal frequencies by $3/4$ codes, and the finest details by $7/8$ punctured codes. We refer to this second coder as "MC-EZBC."

Fig. A.1 presents the results obtained using the three different coding schemes for the "Hall monitor" and "Foreman" sequences at different bitrates over a Gaussian channel with four different channel-noise levels. Note that, for the noiseless channel, the IA is entirely uncoded for our JSC+RM scheme. In Fig. A.1 we observe that both MCA and MC-EZBC yield performance superior to JSC+RM when the channel is noiseless. This is as expected, as these two algorithms are designed for noiseless channels. In particular, we expect MCA to outperform JSC+RM due to the unstructured nature of the codebooks generated by GLA, whereas the JSC+RM codebooks are highly structured. On the other hand, when the channel becomes very noisy (e.g., channel SNRs of 4.33 dB and 3 dB), the performances of MCA and MC-EZBC drop dramatically while JSC+RM remains quite close to its noiseless performance. Indeed, JSC+RM consistently outperforms both MCA

and MC-EZBC for the very noisy channel.

A.1.5 Conclusion

In this paper, we presented an approach to the JSC coding of scalable MCTF video. The proposed system is based on structured VQ coupled with linear IA in the form of uncoded IA, or coded IA via Reed-Muller codes. We compared the performance of the proposed system to that of a source-optimized channel coding using unstructured VQ codebooks without an explicit channel coder, as well as to that of the prominent MCTF-based MC-EZBC coder protected unequally with RPC codes, in the more traditional paradigm of concatenated source and channel codes. As the channel noise increases, the proposed coding system retains end-to-end distortion performance close to that of the noiseless channel as well as consistently outperforms the other two schemes for very low channel SNR.

As a final observation, we note that, at the encoder, the complexity of our scheme is similar to that of the unstructured-VQ coder, except we avoid the IA post processing of [153] in the creation of VQ codebooks. No additional encoder complexity occurs due to the use of coded IA, since the VQ source codewords belong to the space of channel codewords, unlike concatenated source-channel schemes that require subsequent channel-coding processing. On the other hand, at the decoder, the complexity of our proposed scheme is comparable to that of concatenated schemes as Viterbi decoding is required in both cases.

A.2 Rotated Constellations for Transmission over Rayleigh Fading Channels

A.2.1 Introduction

Video transmission over fading channels may suffer substantial quality degradation due to the nature of the channel. On a fading channel, errors occur in reception when the channel attenuation is large. However, if the receiver can be supplied with several replicas of the same information signal transmitted over independently fading channels, the probability that all the signal components fade simultaneously is reduced considerably. Several strategies for such diversity reception have been developed, including frequency diversity, time diversity, and diversity techniques based on multiple antennas; an example of the latter class applied to video is [120]. In addition to these common diversity approaches, we can also speak of *modulation diversity* [27] wherein special multidimensional signal constellations having lattice structure are used. Such constellations provide the receiver with an order of diversity dependent on the number of dimensions of the signal constellation. The diversity order, L, of a signal set of dimension n is the minimum number of distinct components between any two constellation points. Given a $\mathbb{Z}^n$-lattice constellation, the desired modulation diversity is obtained by applying a suitable rotation using algebraic number-theoretical tools [27, 24]. In this letter, we consider rotated $\mathbb{Z}^n$-lattices with full diversity (i.e., $L = n$) in order to achieve reliable transmission over flat Rayleigh fading channels.

Previously, we have used such rotated $\mathbb{Z}^n$-lattices as the basis of joint source-channel coding (JSCC). In [26], it is shown that structured vector quantization (VQ) resulting from these rotated lattices combined with linear labeling simultaneously minimizes the channel and the source distortion for Gaussian sources. In [72], we extended this coding scheme, originally devised for Gaussian sources, to video sequences whose source distribution is decidedly not Gaussian. Specifically, we developed a scalable video-coding system constructed from $t + 2D$ motion-compensated temporal filtering (MCTF) coupled with the structured VQ of [26] with several modifications to accommodate the non-Gaussianity of the spatiotemporal subbands. In [70], we coupled this JSCC video coder with partially coded index assignment for robust transmission over a binary symmetric channel, and, in [71], we paired the coder with rate-compatible punctured convolutional (RCPC) codes for the flat Rayleigh channel with independent fadings. This latter system was shown to be robust in the presence of fading; however, a trade-off was required between the compression efficiency and the added redundancy. Additionally, it was clear that the performance of a BPSK constellation without protection drops dramatically over a fading channel.

In this letter, we apply modulation diversity to our JSCC video coder and demonstrate that, by rotating the given constellation in order to increase its diversity order, we achieve robustness to fading without adding redundancy. In addition, we compare the proposed JSCC scheme to the more traditional approach to error resilience consisting of concatenating a state-of-the-art source coder (the prominent MCTF-based coder MC-EZBC [39]) with a channel coder (RCPC codes) applied optimally for unequal error protection (UEP). Whereas performance of this latter MC-EZBC coder degrades quickly as channel fading increases, the proposed JSCC coder maintains performance close to that of the noiseless channel.

A.2.2 JSCC for Video

A.2.2.1 JSCC via VQ and Linear Labeling

Let a d-dimensional vector $\boldsymbol{x}$ be the input to a vector quantizer producing an n-bit binary codeword $\boldsymbol{b}$ which is the index of the vector used for signal reconstruction at the receiver. The source codebook can be viewed as a function of $\boldsymbol{b} = \begin{bmatrix} b_1 & \cdots & b_n \end{bmatrix}^T \in \{+1, -1\}^n =$ BPSK$_n$ representing a VQ index assignment. Under the assumption of a maxentropic quantizer, the total distortion is the sum of a source distortion due to quantization and a channel distortion dependent on the index assignment. In [105], it is proved that, for a binary symmetric channel, the channel distortion is minimized by an index assignment in the form of a linear labeling; additionally, in [26], a linear labeling that minimizes the source distortion was constructed. This labeling is fully described by a $d \times n$ matrix $\boldsymbol{G}_{d,n}$ which, in essence, transforms an identically distributed random variable into a random variable (the source codebook) which mimics the source distribution.

Let $\boldsymbol{U}_n$ be an $n \times n$ square matrix and $\boldsymbol{G}_{d,n}$ be any combination of d rows of the matrix $\boldsymbol{U}_n$. In order to construct codebook S_n with codewords $\boldsymbol{y}^{(l)} = \boldsymbol{G}_{d,n}\boldsymbol{b}^{(l)}$ where $\boldsymbol{b}^{(l)} \in$ BPSK$_n$, $1 \leq l \leq 2^n$, such that S_n is an asymptotically Gaussian source dictionary which minimizes the source distortion as $n \to \infty$, it is shown in [26] that $\boldsymbol{U}_n$ must be orthogonal with coefficients going uniformly to 0 as $n \to \infty$. The mapping $\boldsymbol{b} \in$ BPSK$_n \to (\boldsymbol{G}_{d,n}\boldsymbol{b} \in S_n)$ is then linear and allows the building of a source dictionary which is asymptotically Gaussian.

In [26], it is shown that matrix $\boldsymbol{U}_n$ being the generator matrix of full-diversity rotated $\mathbb{Z}^n$-lattice constellations satisfies the above conditions. This generator matrix is constructed by the canonical embeddings of the ideal ring of the maximal real cyclotomic fields [24]. Similar properties are achieved when selecting r columns of the matrix $\boldsymbol{U}_n$, and we shall denote the $n \times r$ matrices constructed in this manner as $\boldsymbol{G}'_{n,r}$. The reader is referred to [26] and also [28] for more detail.

A.2.2.2 JSCC of Spatiotemporal Subbands

The JSCC video system we proposed in [72] couples a $t + 2D$ MCTF wavelet decomposition with the JSCC scheme described above. The MCTF decomposition permits spatial and temporal resolution scalability; however, since the resulting spatiotemporal wavelet coefficients are not Gaussian, the JSCC scheme cannot be applied directly. Thus, in order to take into account the non-Gaussianity of the video source, we classify the spatiotemporal coefficients in each subband into two classes and adapt the quantizer to each class.

Specifically, in the temporal lowpass (approximation) frames, subband vectors are classified according to the norm of the uncoded vectors. On the other hand, in the temporal highpass (detail) frames, the classification is based on a stochastic model of the spatiotemporal dependencies between coefficients. Following this model, we consider the conditional probability of the coefficients in a given subband to be Gaussian with variance depending on the set of spatiotemporal neighbors. The spatiotemporal neighborhood consists of coefficients that have already been received by the decoder when the current coefficient has to be decoded; as a consequence the same procedure is applied at the decoder, and no side information is required to indicate the class of each vector.

For each class, we design a VQ codebook by minimizing cost $\gamma = \min_l \mathrm{E}\left[||\boldsymbol{x} - \beta\boldsymbol{G}_{d,n}\boldsymbol{b}^{(l)}||^2\right]$, where $\boldsymbol{b}^{(l)}$ ranges over the set of 2^n possible codewords of BPSK$_n$, and β is a parameter which scales the lattice constellation $\boldsymbol{G}_{d,n}$ to the source

dynamics. In order to find β, as well as the codebook with vectors $\boldsymbol{y}^{(l)} = \beta \boldsymbol{G}_{d,n} \boldsymbol{b}^{(l)}$, an iterative optimization algorithm (similar to that of shape-gain VQ) is used. A similar optimization is applied when using the matrix $\boldsymbol{G}'_{n,r}$ for VQ

Based on the fact that the channel distortion is minimized with a linear labeling, we use an iterative bit-allocation algorithm which takes into account a nonnegativity constraint on the rate and optimally distributes the available bits among the spatiotemporal subbands. This bit-allocation algorithm indicates the size of the $\boldsymbol{G}_{d,n}$ or $\boldsymbol{G}'_{n,r}$ matrix which minimizes the end-to-end distortion, under certain constraints dictated by computational-complexity issues and dependencies among the spatiotemporal coefficients. For more details, see [72].

A.2.3 JSCC using a Rotation Matrix Prior to the Fading channel

In [71], it was seen that transmitting the codewords $\boldsymbol{b} \in \text{BPSK}_n$ on a fading channel without protection resulted in reconstructed sequences of rather poor quality. Thus, we employed RCPC codes, resulting in, unfortunately, a significantly increased bitrate due to the added redundancy. Here, we consider an alternative solution wherein no redundancy is added, and the robustness is achieved due to modulation diversity offered by rotated constellations.

The design of matrices that can rotate a $\mathbb{Z}^n$-lattice so as to produce a full-diversity signal constellation suitable for modulation diversity is a difficult task, and practical constructions have been provided only by Bayer-Fluckiger *el al*. [24] and Belfiore *et al*. [26]. However, the structured VQ codebooks we already use in our JSCC video-coding system ($\boldsymbol{G}_{d,n}$ or $\boldsymbol{G}'_{n,r}$) are derived from rows or columns of the $n \times n$ generator matrix $\boldsymbol{U}_n$ developed in [26]. Since this $\boldsymbol{U}_n$ is capable of rotating a $\mathbb{Z}^n$-lattice to produce a full-diversity constellation, we can use it as the rotation matrix for modulation diversity in addition to its role in quantization-codebook generation.

Thus, to achieve modulation diversity, we can proceed as follows. Assuming quantization with $\boldsymbol{G}_{d,n}$, prior to transmission through the channel, we apply the matrix $\boldsymbol{U}_n$ as a rotation, producing $\boldsymbol{u} = \boldsymbol{U}_n^T \boldsymbol{b}$, where $\boldsymbol{b} \in \text{BPSK}_n$. $\boldsymbol{u}$ is a vector of real values, $\boldsymbol{u} \in \mathbb{R}^n$; additionally, $\boldsymbol{u}$ belongs to the n-dimensional cubic lattice $\mathbb{Z}_{n,L=n}$ with generator matrix $\boldsymbol{U}_n$ and full diversity $L = n$. Lattice $\mathbb{Z}_{n,L=n}$ is a rotation of $\mathbb{Z}^n$ guaranteeing a maximum degree of diversity. In the case of quantization via $\boldsymbol{G}'_{n,r}$, the rotation matrix is $r \times r$ matrix $\boldsymbol{U}_r$, $\boldsymbol{u} = \boldsymbol{U}_r^T \boldsymbol{b}$ with $\boldsymbol{b} \in \text{BPSK}_r$, and $\boldsymbol{u}$ belongs to the r-dimensional cubic lattice $\mathbb{Z}_{r,L=r}$ with generator matrix $\boldsymbol{U}_r$ and full diversity $L = r$.

Decoding is based on a maximum-likelihood (ML) decoding algorithm, the sphere decoder [202]. In this fashion, we obtain the benefit of the increased diversity due to the rotation simultaneously with the benefit of an ML decoding method at the decoder.

The operation of the system is depicted in Fig. A.2. In this figure, $\boldsymbol{x}$ is a vector of spatiotemporal wavelet coefficients that is quantized and indexed by a vector $\boldsymbol{b}$ through the matrix $\boldsymbol{G}_{d,n}$ or $\boldsymbol{G}'_{n,r}$ using the classification and the bit-allocation procedures described above in Sec. A.2.2.2. $\boldsymbol{u}$ is a vector of real values obtained by applying the appropriate rotation matrix, $\boldsymbol{U}_n$ or $\boldsymbol{U}_r$. The channel is a flat fading Rayleigh channel, and we assume that channel-state information is known at the receiver. The assumption of independent fadings is approached in practical systems through the use of an interleaver/deinterleaver. Thus, the received vector after deinterleaving is given by $\hat{\boldsymbol{u}} = \boldsymbol{a} * \boldsymbol{u} + \boldsymbol{z}$, where $\boldsymbol{a}$ is a vector of the random fading coefficients with Rayleigh distribution, and $E[a_i^2] = 1$; $\boldsymbol{z}$ is a noise vector with Gaussian-distributed independent random variables with zero mean

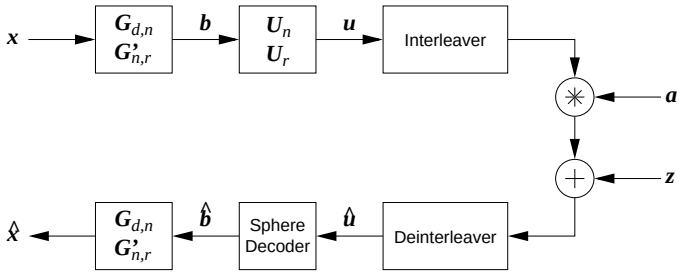


Figure A.2: System model including a rotation matrix prior to the transmission on a flat Rayleigh fading channel.

Figure A.3: Reconstructed frames after transmission on a flat Rayleigh fading channel, both with (right) and without (left) a rotation matrix. Channel SNR is 8.0 dB.

and variance N_0; and $*$ denotes a component-wise product. We extract, using the sphere decoder, the corresponding $\hat{\boldsymbol{b}}$ vector, and, through a simple multiplication with $\boldsymbol{G}_{d,n}$ or $\boldsymbol{G}'_{n,r}$, the vector $\hat{\boldsymbol{x}}$ of reconstructed wavelet coefficients.

As a final note, we observe that we use a rotation matrix with size matched to the dimension of the vector quantizer employed (i.e., $\boldsymbol{U}_n$ is used for VQ with $\boldsymbol{G}_{d,n}$; $\boldsymbol{U}_r$ is used for VQ with $\boldsymbol{G}'_{n,r}$). In general, it is possible to use a rotation matrix with dimension greater than or equal to that of the vector at the output of the quantizer, and the higher the dimensionality chosen for the rotation matrix, the higher will be the order of diversity. However, the computational complexity of the decoder increases significantly with the dimensionality of the rotation matrix; for this reason, we restrict the dimensionality of the rotation matrix to match that of the quantizer in the simulations presented below.

A.2.4 Experimental results

We consider a temporal Haar decomposition applied on GOFs of 16 frames, with 4 temporal and 2 spatial resolution levels. The spatial multiresolution analysis uses biorthogonal $9/7$ filters, while the temporal decomposition includes motion compensation via block matching with full-pixel accuracy. The global rates tested in bits per pixel (bpp) are 0.1 bpp, 0.16 bpp, 0.33 bpp, and 0.48 bpp. Consequently, in accordance with these coding rates, the possible rotation matrices are U_{16}, U_8, U_4 and U_2. As noted above, we match the dimension of the rotation matrix to that of quantizer output in order to reduce complexity at the decoder. Moreover, we restrict the dimension to be less than or equal to 16 thereby permitting use of the sphere decoder of [202], which is limited to dimensions no greater than 32 for complexity reasons.

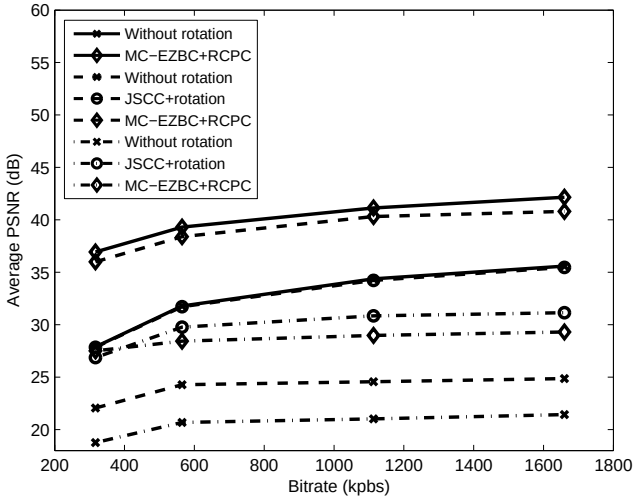


Figure A.4: Performance for "Hall monitor" for channel of SNR = noiseless (solid lines), 11.0 dB (dashed lines), and 8.0 dB (dash-dot lines).

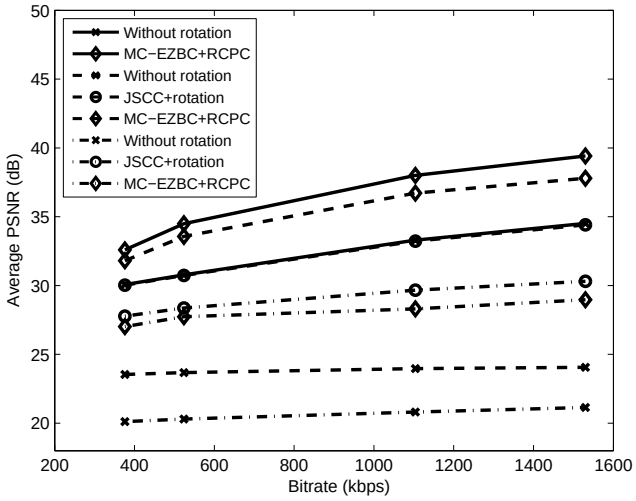


Figure A.5: Performance for "Foreman" for channel of SNR = noiseless (solid lines), 11.0 dB (dashed lines), and 8.0 dB (dash-dot lines).

For our experiments, we consider CIF (352×288) test sequences at 30 fps. Fig. A.3 illustrates a reconstructed frame of the sequence "foreman" after transmission over a flat Rayleigh fading channel for the two cases of coding with a rotation matrix prior to the transmission and without rotation.

We compare our rotation-based JSCC scheme (which we denote "JSCC+rotation") to a second video coder which corresponds to an implementation of MCTF concatenated with traditional error-control coding. We employ the prominent MC-EZBC [39] coder which provides state-of-the-art MCTF coding performance, and, to provide error resilience, we packetize the MC-EZBC bitstream while applying RCPC codes to the resulting packets (we denote the resulting system as "MC-EZBC+RCPC"). Specifically, each packet contains the information corresponding to a single spatial resolution level from a single temporal subband frame. Hence, the packets have unequal length and are protected with a UEP arrangement of RCPC codes. If the decoder fails to decode a received packet, the packet is dropped. The RCPC codes have $R_p = 2/3$, $3/4$, and $7/8$ with memory $m = 6$ and mother code $R = 1/2$ [88]. The UEP arrangement of RCPC puncturing rates is determined by an exhaustive-search procedure that examines all possible combinations of the three R_p rates applied to the packets, under the constraint that the puncturing rate for a given packet cannot be greater than that applied to packets from higher spatiotemporal resolution levels. For each particular combination of RCPC protection, the source rate of each subband is decreased appropriately to maintain a fixed global bitrate in accordance with the original MC-EZBC rate-allocation scheme in which each subband is allocated source rate proportional to the subband variance. The UEP combination yielding the maximum end-to-end PSNR as averaged over 100 noise realizations is then selected. Although this exhaustive-search procedure is clearly impractical, it couples the state-of-the-art MCTF source coder with UEP channel protection optimal under the constraints, and provides a reasonable best-case alternative to which to compare.

Figs. A.4 and A.5 present the average PSNR for the "hall-monitor" and "foreman" sequences obtained by the three coding schemes at different bitrates after transmission on a flat Rayleigh channel for both moderate (11.0 dB) and low (8.0 dB) channel SNR; In all cases, PSNR figures are averaged over 100 noise realizations. In these results, "without rotation" refers to JSCC+rotation without the rotation matrix applied, which is, in essense, the coder of [72].

We observe that the MC-EZBC+RCPC coding scheme yields superior performance under noiseless channel conditions; this is as expected since MC-EZBC itself performs significantly better than does our VQ-based source coder for a noiseless channel. Consequently, as the channel noise increases, we expect that MC-EZBC+RCPC will continue to outperform the proposed scheme, as long as the channel remains "close" to being noiseless. However, as the channel SNR becomes increasingly worse, the Rayleigh fading channel imposes rather difficult conditions, and the performance of MC-EZBC+RCPC falls off rather precipitously. In this case, classical protection in the form of RCPC codes does not work as well as the proposed modulation-diversity approach to JSCC. In fact, the JSCC+rotation coding scheme remains quite close to its noiseless performance and outperforms MC-EZBC+RCPC for most cases of low channel SNR.

A.2.5 Conclusion

In this letter, we presented a robust joint source-channel video-coding scheme for transmission over a flat Rayleigh fading channel. The proposed system combines an MCTF-

based spatiotemporal wavelet decomposition, structured VQ with linear labeling, and modulation diversity in the form of a rotation matrix with maximum diversity such that robustness is achieved without the addition of redundancy. In experimental results, we compared the proposed system to the prominent MCTF-based MC-EZBC coder protected unequally with RCPC codes, representing the more traditional alternative of concatenating source and channel codes. As the channel fading increases, the proposed system retains end-to-end distortion performance close to that of the noiseless channel, while the MC-EZBC coder suffers from a dramatic decrease in performance. As a consequence, the proposed system achieves superior performance for almost all instances of low channel SNR.

Publications

International journal articles

1. M. Trocan, B. Pesquet-Popescu and J.E. Fowler Graph-cut based rate distortion Lagrangian model for subband image compression. submitted to *Hindawi Journal of Applied Signal Processing*, 2007.

2. G. Feideropoulou, M. Trocan, J.E. Fowler, B. Pesquet-Popescu and J.C. Belfiore Rotated constellations for video transmission over Rayleigh fading channels. *IEEE Signal Processing Letters*, 2007.

3. G. Feideropoulou, M. Trocan, J.E. Fowler, B. Pesquet-Popescu and J.C. Belfiore Joint source-channel coding with partially coded index assignment for robust scalable video. *IEEE Signal Processing Letters*, 2006.

International conference articles

1. M. Trocan , C. Tillier, and B. Pesquet-Popescu. A sliding window implementation of the 5-band motion compensated temporal lifting scheme. In *Proc. of the IEEE International Workshop on Nonlinear Signal and Image Processing, NSIP'07*, Bucharest, Romania, September 2007.

2. M. Trocan and B. Pesquet-Popescu. Graph-cut rate distortion optimization for subband image compression. In *Proc. of the IEEE European Signal Processing Conference, EUSIPCO'07*, Poznan, Poland, September 2007.

3. M. Trocan, B. Pesquet-Popescu and J.E. Fowler. Graph-cut rate distortion algorithm for contourlet based image compression. In *Proc. of the IEEE Int. Conf. on Image Processing*, San Antonio, Texas U.S.A., September 2007.

4. M. Trocan and B. Pesquet-Popescu. Video coding with fully separable wavelet and wavelet packet transforms. In *Proc. of SPIE Visual Communications and Image Processing*, San Jose, California U.S.A., January 2007.

5. M. Trocan, C. Tillier, B. Pesquet-Popescu, and M. van der Schaar. A 5-band temporal lifting scheme for video surveillance. In *Proc. of the IEEE International Workshop on Multimedia Signal Processing*, Victoria B.C., Canada, October 2006.

6. B.U. Toreyin, M. Trocan, E. Cetin, and B. Pesquet-Popescu. Linear and nonlinear temporal prediction employing lifting structures for scalable video coding. In *Proc. of the IEEE European Signal Processing Conference*, Florence, Italy, September 2006.

7. B.U. Toreyin, M. Trocan, E. Cetin, and B. Pesquet-Popescu. LMS-based adaptive prediction for scalable video coding . In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Proc.*, Toulouse, France, May 2006.

8. M. Trocan and B. Pesquet-Popescu. Scene-cut processing in motion- compensated temporal filtering. In *Proc. of Advanced Concepts for Intelligent Vision Systems (ACIVS)*, Anwertpen, Belgium, Septembre 2005.

9. M. Trocan, C. Tillier, and B. Pesquet-Popescu. Joint wavelet packets for group of frames in MCTF. In *Proc. of SPIE Visual Communications and Image Processing*, San Diego, California U.S.A., July 2005.

10. L. Gueguen, M. Trocan, H. Maitre, A. Giros, M. Datcu and B. Pesquet-Popescu. A comparison of multispectral satellite sequence compression approaches. In *Proc. of IEEE International Symposium on Signals, Circuits and Systems*, Iasi, Romania, July 2005.

MPEG Standardization Contributions

1. R. Xiong X. Ji, D. Zhang, J. Xu, G. Pau, M. Trocan and V. Bottreau. Vidwav Wavelet Video Coding Specifications. Doc. M12339, MPEG 73th meeting, Juillet 2005.

2. G. Pau, M. Trocan and B. Pesquet-Popescu. Bidirectional Joint Motion Estimation for Vidwav Software. Doc. M12303, MPEG 73th meeting, Juillet 2005.

3. M. Trocan, G. Pau and B. Pesquet-Popescu. Cross-Verification of RWTH Results on SVC CE-4. Doc. M11318, MPEG 70th meeting, Octobre 2004.

List of Figures

List of Tables

Bibliography

[1] Coding of audiovisual objects, part 10: Advanced video coding. *Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), ISO/IEC 14 496-10 (identical to ITU-T Recommendation H.264.*

[2] Coding of audiovisual objects, part 2: Visual. *Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), 14 496-2.*

[3] Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s, part 2: Video. *Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), ISO/IEC 11 172-2.*

[4] Generic coding of moving pictures and associated audio information, part 2: Video. *Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), ISO/IEC 11 172-2.*

[5] Mpeg-4 video verification model version 17.0. *ISO/IEC JTC1/SC29/WG11 N3515,Beijing, China, Jul. 2000.*

[6] Video codec for audiovisual services at p x 64 kbit/s. *Int. Telecommun. Union-Telecommun. (ITU-T), Geneva, Switzerland, Recommendation H.261.*

[7] Video coding for low bit rate communication. *Int. Telecommun. Union-Telecommun. (ITU-T), Geneva, Switzerland, Recommendation H.263.*

[8] Information technology – JPEG 2000 image coding system. Technical report, ISO/IEC 15444-1, 2000.

[9] Wavelet codec reference document and software manual. MPEG document N7334, July 2005.

[10] J.-C. Pesquet A. Benazza-Benyahia and M.H. Gharbia. Adapted Vector-Lifting Schemes for Multiband Textured Image Coding. In *IGARSS*, Toulouse, July 2003.

[11] T. Aboulnasr and K. Mayyas. A robust variable step-size LMS-type algorithm: analysis and simulations. *IEEE Transactions on Signal Processing*, 45(3):631–639, 1997.

[12] MPEG Video AhG. Call for proposals on scalable video coding technology. Doc. N5958, Brisbane MPEG 66th meeting, October 2003.

[13] I. Ahmad, X. Wei, Y. Sun, and Y.-Q. Zhang. Video transcoding: An overview of various techniques and research issues. *IEEE Transactions on Multimedia*, 7(5):793 –803, 2005.

[14] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice-Hall, Upper Saddle River, 1993.

[15] T. André, B. Pesquet-Popescu, M. Gastaud, M. Antonini, and M. Barlaud. Motion estimation using chrominance for wavelet-based video coding. *Picture Coding Symposium, PCS'04*, 2004, San Francisco, CA.

[16] Y. Andreopoulos, A. Munteanu, J. Barbarien, M. van der Schaar, J. Cornelis, and P. Schelkens. In-band motion compensated temporal filtering. *EURASIP Signal Processing: Image Communication (special issue on Subband/Wavelet Interframe Video Coding)*, 19:653–673, 2004.

[17] Y. Andreopoulos, A. Munteanu, G. Van der Auwera, J.P.H. Cornelis, and P. Schelkens. Complete-to-overcomplete discrete wavelet transforms: theory and applications. *IEEE Transactions on Signal Processing*, 53:1398–1412, 2005.

[18] Y. Andreopoulos, M. van der Schaar, A. Munteanu, J. Barbarien, P. Schelkens, and J. Cornelis. Fully scalable 3-D overcomplete wavelet video coding using adaptive motion compensated temporal filtering. *Proc. on IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'03)*, 2003.

[19] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Trans. on Image Proc.*, 1:205–220, 1992.

[20] S. Attallah and M. Najim. On the convergence enhancement of the wavelet transform based LMS. In *IEEE Int. Conf. Acoustics, Speech, Signal Processing*, pages 973–976, Detroit, MI, May 1995.

[21] L. Alparone B. Aiazzi and S. Baronti. Near-lossless compression of 3-D optical data. *IEEE Transctions on Geoscience and Remote Sensing*, 39(11):2547–2558, Nov. 2001.

[22] J. Flusser B. Zitova. Image registration methods: A survey. *Image and Vision Computing*, 21: 977–1000, 2003.

[23] J. Barbarien, Y. Andreopoulos, A. Munteanu, P. Schelkens, and J. Cornelis. Coding of motion vectors produced by wavelet-domain motion estimation. *Proc. Picture Coding Symposium'03*, pages 193–198, 2003.

[24] Eva Bayer-Fluckiger, Frédérique Oggier, and Emanuele Viterbo. New algebraic constructions of rotated $\mathbf{Z}^n$-lattice constellations for the Rayleigh fading channel. *IEEE Transactions on Information Theory*, 50(4):702–714, April 2004.

[25] A.N. Belbachir and P.M. Goebel. The contourlet transform for image compression. *Proc. of Physics in Signal and Image Processing Conference, PSIT*, 2005.

[26] Jean-Claude Belfiore, Xavier Giraud, and Jorge Rodriguez-Guisantes. Optimal linear labeling for the minimization of both source and channel distortion. In *Proceedings of the IEEE International Symposium on Information Theory*, page 404, Sorrento, Italy, June 2000.

[27] Joseph Boutros and Emanuele Viterbo. Signal space diversity: A power- and bandwidth-efficient diversity technique for the Rayleigh fading channel. *IEEE Transactions on Information Theory*, 44(4):1453–1467, July 1998.

[28] Joseph Boutros, Emmanuele Viterbo, Catherine Rastello, and Jean-Claude Belfiore. Good lattice constellations for both Rayleigh fading and Gaussian channels. *IEEE Transactions on Information Theory*, 42(2):502–518, March 1996.

[29] Y. Boykov and G.Funka-Lea. Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision*, 70:109–131, 2006.

[30] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1124 – 1137, 2004.

[31] Y. Boykov and O. Veksler. Graph cuts in vision and graphics: Theories and applications. *Handbook of Mathematical Models in Computer Vision, Springer-Verlag*, pages 79–76, 2006.

[32] Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. *IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[33] Y. Boykov, O. Veksler, and R Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1222 – 1239, 2001.

[34] Y. Boykov and V.Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. *Proc. of IEEE International Conference on Computer Vision*, 1:26–33, 2003.

[35] Maja Bystrom and Thomas Stockhammer. Dependent source and channel rate allocation for video transmission. *IEEE Transactions on Wireless Communications*, 3(1):258–268, January 2004.

[36] M. Cagnazzo, F. Castaldo, T. Andre, M. Antonini, and M. Barlaud. Optimal motion estimation for wavelet video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 17:907–911, 2007.

[37] V. Chappelier, C. Guillemot, and S. Marinkovic. Image coding with iterated contourlet and wavelet transforms. *Proc. of IEEE International Conference on Image Processing*, 5:3157– 3160, 2004.

[38] P. Chen and J.Woods. Bidirectional MC-EZBC with lifting implementation. *IEEE Trans.on CSVT*, 14:1183–1194, 2004.

[39] Peisong Chen and John W. Woods. Bidirectional MC-EZBC with lifting implementation. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(10):1183–1194, October 2004.

[40] S.-C. Chen and M.-L. Shyu. Video scene change detection method using unsupervised segmentation and object tracking. *ICME*, Tokyo, 2001.

[41] Y.J. Chen, S. Oraintara, and K. Amaratunga. M-channel lifting-based design of paraunitary and biorthogonal filter banks with structural regularity. In *Proceedings of the IEEE International Conference on Circuits and Systems*, pages IV 221–IV 224, May 2003.

[42] P.Y. Cheng, J. Li, and C.-C. J. Kuo. Video coding using embedded wavelet packet transform and motion compensation. *Proc. of SPIE Visual Imformation Processing*, 1996.

[43] Gene Cheung and Avideh Zakhor. Bit allocation for joint source/channel coding of scalable video. *IEEE Transactions on Image Processing*, 9(3):340–356, March 2000.

[44] S.J. Choi and J.W. Woods. Motion-compensated 3-D subband coding of video. *IEEE Trans. on Image Proc.*, 8:155–167, 1999.

[45] W. Y. Choi and R. Park. Motion vector coding with conditional transmission. *Proceedings of GLOBECOM'91*, pages 85–90, 1991.

[46] S. Chopra and M. R. Rao. On the multiway cut polyhedron. *Networks*, 21:51–89, 1991.

[47] A. Cohen, I. Daubechies, and J. C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45:485–500, 1992.

[48] A. Cohen and B. Matei. Nonlinear subdivison schemes : Applications to image processing. *Tutorials on Multiresolution in Geometric Modelling*, pages 93–97, 2002.

[49] R.R. Coifman and M.V. Wickerhauser. Entropy based methods for best basis selection. *IEEE Transactions on Information Theory*, 38:719–746, 1992.

[50] R.R. Coifman and V. Wickerhauser. *Wavelets and adapted waveform analysis*. J.J. Benedetto and M. Frazier, 1993.

[51] D. Coltuc, M. Datcu, and K. Seidel. Multichannel Compression for Sequential Image Retrieval. In G.J.A. Nieuwenhuis, R.A. Vaughan, and M. Molenaar, editors, *Operational Remote Sensing for Sustainable Development, 18th EARSeL Symposium 1998*, pages 345–351. A. A. Balkema Rotterdam/Brookfield, 1999, Enschede, Netherlands.

[52] D. Coltuc, D. Luca, K. Seidel, and M. Datcu. 3-Dimensional Signal Analysis of Multichannel Spectrometric Imagery. In E. Parlow, editor, *Progress in Environmental Research and Applications, 15th EARSeL Symposium 1995, Basel, Switzerland*, pages 155–162, Sept 1996.

[53] C.F.N. Cowan and P.M. Grant. *Adaptive Filters*. Prentice-Hall, Englewood Cliffs, New Jersey, 1985.

[54] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23:864–894, 1994.

[55] J. Darbon and M. Sigelle. Image restoration with discrete constrained total variation part I: Fast and exact optimization. *Journal of Mathematical Imaging and Vision*, 26:261–276, 2006.

[56] J. Darbon and M. Sigelle. Image restoration with discrete constrained total variation part II: Levelable functions, convex priors and non-convex cases. *Journal of Mathematical Imaging and Vision*, 26:277–291, 2006.

[57] I. Daubechies. The wavelet transform, time-frequency localization, and signal analysis. *IEEE Transactions on Information Theory*, 36:961–1005, 1990.

[58] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics Journal of Control and Optimization, Philadelphia, Pennsylvania, 1992.

[59] I. Daubechies. Where do wavelets come from? – A personal point of view. *Proceedings of the IEEE*, 84:510–513, 1996.

[60] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4(3):245–267, 1998.

[61] R. L. De Queiroz, G. Bozdagi, and T. Sencar. Fast video segmentation using encoding cost data. Technical report, Xerox Corporation, 1999.

[62] M. N. Do and M. Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE Trans. on Image Proc.*, 14:2091–2106, 2005.

[63] M. N. Do and M. Vetterli. Pyramidal directional filter banks and curvelets. *Proceedings of the IEEE International Conference on Image Processing*, Thessaloniki, Greece, Oct.2001.

[64] D. L. Donoho. Wedgelets: Nearly minimax estimation of edges. Technical report, Statistics Department, Stanford University, 1997.

[65] P.L. Dragotti, V. Velisavljevic, M. Vetterli, and B. Beferull-Lozano. Discrete directional wavelet bases and frames for image compression and denoising. *Proc. of the SPIE Conference on Wavelet Applications in Signal and Image Processing*, San Diego, USA, August 2003.

[66] Q. Du and C.-I Chang. Linear mixture analysis-based compression for hyperspectral image analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 42(4):875–891, April 2004.

[67] Q. Du and J. E. Fowler. Hyperspectral image compression using JPEG2000 and principal component analysis. *IEEE Geoscience and Remote Sensing Letters*, 4:201–205, 2007.

[68] F. Dufaux and F Moscheni. Motion estimation techniques for digital TV: a review and a new contribution. *Proceedings of the IEEE Journal*, 83:858 – 876, 1995.

[69] G. Feideropoulou, M. Trocan, J.E. Fowler, B. Pesquet-Popescu, and J.C. Belfiore. Rotated constellations for video transmission over rayleigh fading channels. *to appear Signal Processing Letters*, 2007.

[70] G. Feideropoulou, M. Trocan, J.E. Fowler, B. Pesquet-Popescu, and J.C. Belfiore. Joint source-channel coding with partially coded index assignment for robust scalable video. *IEEE Signal Processing Letters*, April 2006.

[71] Georgia Feideropoulou, Béatrice Pesquet-Popescu, and Jean-Claude Belfiore. Joint source-channel coding of scalable video on a Rayleigh fading channel. In *Proceedings of the IEEE Workshop on Multimedia Signal Processing*, pages 303–306, Sienna, Italy, September 2004.

[72] Georgia Feideropoulou, Béatrice Pesquet-Popescu, and Jean-Claude Belfiore. Bit allocation algorithm for joint source-channel coding of $t + 2D$ video sequences. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 177–180, Philadelphia, PA, March 2005.

[73] Georgia Feideropoulou, Béatrice Pesquet-Popescu, Jean-Claude Belfiore, and Jorge Rodriguez. Non-linear modeling of wavelet coefficients for a video sequence. In *Proceedings of the International Workshop on Nonlinear Signal and Image Processing*, Grado, Italy, June 2003.

[74] B. Feng, J. Xu, F. Wu, S. Yang, and S. Li. Energy distributed update steps (EDU) in lifting based motion compensated video coding. *Proc. IEEE ICIP'04*, Singapore, Oct. 2004.

[75] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[76] James E. Fowler and Justin T. Rucker. 3D wavelet-based compression of hyperspectral imagery. In Chien-I Chang, editor, *Hyperspectral Data Exploitation: Theory and Applications*, chapter 14. John Wiley & Sons, Inc., Hoboken, NJ, 2007.

[77] Shrinivas Gadkari and Kenneth Rose. Vector quantization with transmission energy allocation for time-varying channels. *IEEE Transactions on Communications*, 47(1):149–157, January 1999.

[78] D. Le Gall and A. Tabatabai. Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'88*, 2:761–764, 1988.

[79] Ö. N. Gerek and A. E. Cetin. Adaptive polyphase subband decomposition structures for image compression. *IEEE Transactions on Image Processing*, 9:1649–1659, October 2000.

[80] Ö. N. Gerek and A. E. Cetin. Lossless image compression using an edge adapted lifting predictor. *In Proc. of IEEE International Conference on Image Processing, ICIP'2005*, 2005, Genova, Italy.

[81] Xavier Giraud, Emmanuel Boutillon, and Jean-Claude Belfiore. Algebraic tools to build modulation schemes for fading channels. *IEEE Transactions on Information Theory*, 43:938–952, May 1997.

[82] B. Girod. Motion-compensating prediction with fractional-pel accuracy. *IEEE Transactions on Communications*, 41:604–612, 1993.

[83] B. Girod and S. Han. Optimum update for motion-compensated lifting. *IEEE Signal Processing Letters*, 12(2):150–153, 2005.

[84] A. Golwelkar and J. W. Woods. Scalable video compression using longer motion compensated temporal filters. *Proc. SPIE VCIP*, 5150:1406–1416, 2003.

[85] M.W. Marcellin G.P. Abousleman and B.R. Hunt. Compression of hyperspectral imagery using the 3D DCT and hybrid DPCM/DCT. *IEEE Transactions on Geoscience and Remot Sensing*, 33(1):26–35, Jan. 1995.

[86] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society*, 51:271–279, 1989.

[87] L. Gueguen, M. Trocan, H. Maître, A. Giros, M. Datcu, and B. Pesquet-Popescu. A comparison of multispectral satellite sequence compression approaches. *ISSCS*, July 2005, Iasi, Romania.

[88] David Haccoun and Guy Bégin. High-rate punctured convolutional codes for Viterbi and sequential decoding. *IEEE Transactions on Communications*, 37(11):1113–1125, November 1989.

[89] F. J. Hampson and J.-C. Pesquet. M-band nonlinear subband decompositions with perfect reconstruction. *IEEE Trans. on Image Proc.*, 7:1547–1560, 1998.

[90] G.R. Higgie and A.C.M. Fong. Efficient encoding and decoding algorithms for variable-length entropy codes. *IEEE Transactions on Communications*, 150:305–316, 2003.

[91] Bertrand Hochwald and Kenneth Zeger. Tradeoff between source and channel coding. *IEEE Transactions on Information Theory*, 43(5):1412–1424, September 1997.

[92] S. Hosur and A. H. Tewfik. Wavelet transform domain adaptive FIR filtering. *IEEE Transactions on Signal Processing*, 45:617–630, 1997.

[93] S. Hsiang and J. Woods. Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling. In *ISCAS*, pages 589–595, Geneva, Switzerland, 2000.

[94] S.T. Hsiang and J.W. Woods. Invertible three-dimensional analysis/synthesis system for video coding with half-pixel accurate motion compensation. In *VCIP 99, SPIE Vol. 3653*, pages 537–546, 1999.

[95] W.-L. Hsu and H. Derin. 3D adaptive wavelet packet for video compression. *IEEE Transactions on Image Procesing*, 1:602–605, 1995.

[96] X. Huo, J. Chen, and D.L. Donoho. JBEAM: Coding lines and curves via digital beamlets. *Data Compression Conference, 2004. Proceedings. DCC 2004*, pages 449–458.

[97] H. Ishikawa and D. Geiger. Occlusions, discontinuities and epipolar lines in stereo. *Proc. of IEEE European Conference on Computer Vision*, pages 232–248, 1998.

[98] A. G. Tescher J. A. Saghri and J. T. Reagan. Practical transform coding of multispectral imagery. *IEEE Signal Processing Magazine*, pages 32–43, January 1995.

[99] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analyses. *SIAM Review*, 36:377–412, 1994.

[100] G. Jeannic, V. Ricordel, and D. Barba. The edge driven oriented wavelet transform: an anisotropic multidirectional representation with oriented lifting scheme. *In Proceedings of SPIE - Visual Communications and Image Processing, VCIP'07*, 6508, 2007.

[101] G. Karlsson and M. Vetterli. Three-dimensional subband coding of video. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 1100–1103, New York, NY, Apr. 1988.

[102] M. Kawahara, Y.-J. Chiu, and T. Berger. High-speed software implementation of Huffman coding. *Proceedings of IEEE Data Compression Conference, DCC '98*, 1998.

[103] M. Kerdranvat. Hierarchical motion estimation and motion information encoding. *in Proceesing of Third International Workshop on HDTV*, Torino, Italy, 1989.

[104] B.-J. Kim, Z. Xiong, and W.A. Pearlman. Very low bit-rate embedded video coding with 3-D set partitioning in hierarchical trees (3D-SPIHT). *IEEE Trans on Circ. and Syst. for Video Tech.*, 8:1365–1374, 2000.

[105] Petter Knagenhjelm and Erik Agrell. The Hadamard transform—A tool for index assignment. *IEEE Transactions on Information Theory*, 42:1139–1151, July 1996.

[106] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishigurom. Motion compensated interframe coding for video conferencing. *Proc. Nat. Telecommun. Conf., New Orleans*, December 1981.

[107] V. Kolmogorov and R Zabin. What energy functions can be minimized via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:147 – 159, 2004.

[108] J. Konrad and M. Ristivojevic. Video segmentation and occlusion detection over multiple frames. *SPIE VCIP*, San Jose, 2003.

[109] T. Kronander. *Some aspects of perception based image coding*. PhD thesis, 1989.

[110] R. Kutil. Anisotropic 3D wavelet packet bases for video coding. *IEEE Transactions on Image Procesing*, 3:73–76, 2003.

[111] R. Kutil. Anisotropic 3D wavelet packets bases for video coding. *Proceedings of the IEEE International Conference on Image Processing (ICIP'03)*, 3:73–76, 2003.

[112] V. Kwatra, A. Schodl, I. Essa, and A. Bobick. Graphcut textures: image and video synthesis using graph-cuts. *Proceedings of ACM Transactions on Graphics, SIGGRAPH'03*, 2003.

[113] R.H. Kwong and E.W. Johnston. A variable step size LMS algorithm. *IEEE Transactions on Signal Processing*, 40:1663–1642, 1992.

[114] E. Le Pennec and S. G. Mallat. Sparse geometric image representation with bandelets. *IEEE Transactions on Image Processing*, 14(4), 2005.

[115] S.-H. Lee and J.-K. Kim. Fast block motion estimation for overlapped motion compensationusing selective pixel matching. *Proceedings of International Conference on Image Processing, ICIP'99*, 1:80–83, 1999.

[116] R. Li, B. Zeng, and M.L. Liou. A new three-step search algorithm for block motion estimation. *IEEE Trans. on Circuits and Systems for Video Technology*, 4:438–480, 1994.

[117] S. Li, J. Xu, Z. Xiong, and Y.-Q. Zhang. 3D embedded subband coding with optimal truncation (3D-ESCOT). *Applied and Computational Harmonic Analysis*, 10:589, May 2001.

[118] W. Li. Overview of fine granularity scalability in MPEG-4 video standard. *IEEE Trans. Circuits Syst. Video Technol.*, 11:301–317, 2001.

[119] Y.-M. Lin and P.-Y. Chen. An efficient implementation of CAVLC for H.264/AVC. *IEEE International Conference on Innovative Computing, Information and Control, ICICIC '06*, 3:601–604, 2006.

[120] Jianhua Lu, Khaled Ben Letaief, and Ming L. Liou. Robust video transmission over correlated mobile fading channels. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(5):737–751, August 1999.

[121] M.-C.Costa, L. Letocart, and F. Roupin. Minimal multicut and maximal integer multiflow: A survey. *In ELSEVIER European Journal of Operational Research*, pages 55–69, 2005.

[122] K.K. Ma and P.I. Hosur. Performance report of motion vector field adaptive search technique (MVFAST). *ISO/IEC JTC1/SC29/WG11 MPEG99/m5851, Noordwijkerhout, NL*, March, 2000.

[123] S. G. Mallat. Multifrequency channel decompositions of images and wavelet models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37:2091–2110, 1989.

[124] S. G. Mallat. Multiresolution approximations and wavelet orthonormal bases of $\mathrm{l}^2(\mathrm{r})$. *Trans. Amer. Math. Soc.*, 315:69–87, 1989.

[125] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693, 1989.

[126] M. Manikandan, P. Vijayakumar, and N. Ramadass. Motion estimation method for video compression - an overview. *Proceedings of IEEE International Conference on Wireless and Optical Communications Networks, IFIP'06*, 2006.

[127] D. Marpe, H.L. Cycon, and W. Li. Complexity-constrained best-basis wavelet packet algorithm for image compression. *IEEE Proceedings on Vision, Image and Signal Processing*, 145: 391–398, 1998.

[128] D. Marpe, H. Schwarz, and T. Wiegand. Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 13:620–636, 2003.

[129] Robert J. McEliece. On the BCJR trellis for linear block codes. *IEEE Transactions on Information Theory*, 42(4):1072–1091, July 1996.

[130] H. Meyr, H. Rosdolsky, and T. Huang. Optimum run length codes. *IEEE Transactions on Communications*, 22:826–835, 1974.

[131] G.P. Nason and B.W. Silverman. The stationary wavelet transform and some statistical applications. *Wavelets and Statistics, Lecture Notes in Statistics*, pages 281–299, 1995.

[132] M. Nelson and J.-L. Gailly. *The Data Compression Book*. M&T Books, New York, 1995.

[133] A. N. Netravali and B. G. Haskell. *Digital Pictures, Representation and Compression*. 1988.

[134] J.-R. Ohm. Three-dimensional subband coding with motion compensation. *IEEE Trans. on Image Proc.*, 3:559–589, 1994.

[135] J.-R. Ohm. Three-dimensional subband coding with motion compensation. *IEEE Transactions on Image Processing*, 3:559–571, 1994.

[136] J.-R. Ohm. Complexity and delay analysis of MCTF interframe wavelet structures. doc. m8520, Klagenfurt MPEG meeting, July 2002.

[137] J.-R. Ohm, M. van der Schaar, and J. Woods. Interframe wavelet coding: Motion picture representation for universal scalability. *EURASIP Signal Processing: Image Communication, Special issue on Digital Cinema*, 2004.

[138] S. Okubo. Requirements for high quality video coding standards. *Signal Process. Image Commun.*, 4:141–151, 1992.

[139] H. W. Park and H. S. Kim. Motion estimation using low-band-shift method for wavelet-based moving-picture coding. *IEEE Transactions on Image Processing*, 9(4):577–587, 2000.

[140] Y. S. Park and H. W. Park. Arbitrary-ratio image resizing using fast DCT of composite length for DCT-based transcoder. *IEEE Transactions on Image Processing*, 15:494–500, 2006.

[141] G. Pau and B. Pesquet-Popescu. Comparison of spatial M-band filter banks for t+2d video coding. *Proc. of SPIE/IEEE VCIP'05*, Beijing, China, 2005.

[142] G. Pau, B. Pesquet-Popescu, and G. Piella. Modified m-band synthesis filter bank for fractional scalability of images. *IEEE Signal Processing Letters*, 13:345–348, 2006.

[143] G. Pau, C. Tillier, and B. Pesquet-Popescu. Optimization of the predict operator in lifting-based motion compensated temporal filtering. In *Proc. of Visual Communications and Image Processing*, San Jose, CA, January 2004.

[144] G. Pau, C. Tillier, and B. Pesquet-Popescu. Optimization of the predict operator in lifting-based motion compensated temporal filtering. In *SPIE VCIP*, San Jose, CA, USA, jan 2004.

[145] G. Pau, C. Tillier, B. Pesquet-Popescu, and H. Heijmans. Iterative predict optimization in MCTF video. MPEG document m9929, July 2003.

[146] G. Pau, M. Trocan, and Béatrice Pesquet-Popescu. Bidirectional joint motion estimation for vidwav software. Doc. M12303, Poznan MPEG 73th meeting, July 2005.

[147] C. Pe Rosiene and T.Q Nguyen. Tensor-product wavelet vs. Mallat decomposition: a comparative analysis. *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'99)*, 3:431–434, 1999.

[148] W.H. Peng and Y.K. Chen. Enhanced mode adaptive fine granularity scalability. *International Journal of Imaging Systems and Technology*, 13:308–321, 2004.

[149] W.H. Peng and Y.K. Chen. Mode adaptive fine granularity scalability. *Proc. IEEE ICIP'01*, pages 993–996, Greece,2001.

[150] B. Pesquet-Popescu and V. Bottreau. Three-dimensional lifting schemes for motion compensated video compression. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, May 2001.

[151] G. Piella, B. Pesquet-Popescu, H. Heijmans, and G. Pau. Combining seminorms in adaptive lifting schemes and applications to image analysis and compression. *Journal of Mathematical Imaging and Vision*, 24, 2006.

[152] F. Porikli and Y. Wang. Automatic video object segmentation using volume growing and hierarchical clustering. Technical Report TR2004-012, MERL-Mitsubishi Electric Research Laboratory, 2004.

[153] Lee C. Potter and Da-Ming Chiang. Minimax nonredundant channel coding. *IEEE Transactions on Communications*, 43(4):804–811, April 1995.

[154] A. Puri, H. M. Hang, and D. L. Schilling. Interframe coding with variable block-size motion compensation. *in Proceedings of GLOBECOM'87*, pages 65–69, 1987.

[155] A. Raj and R. Zabih. A graph cut algorithm for generalized image deconvolution. *Proc. of IEEE International Conference on Computer Vision*, pages 1048–1054, 2005.

[156] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Transactions on Image Procesing*, 2:160–175, 1993.

[157] K. Ramchandran, M. Vetterli, and C. Herley. Wavelets, subband coding, and best bases. *IEEE Proceedings, Special Issue on Wavelets*, 84:541–560, 1996.

[158] K. Ramchandran, Z. Xiong, K. Asai, and M. Vetterli. Adaptive transforms for image coding using spatially varying wavelet packets. *IEEE Transactions on Image Processing*, 5:1197 – 1204, 1996.

[159] F. Rizzo, B. Carpentieri, G. Motta, and J.A. Storer. Low-complexity lossless compression of hyperspectral imagery via linear prediction. *IEEE Signal Processing Letters*, 12:138–141, Feb. 2005.

[160] S. Roy and I.J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. *Proceedings Sixth International Conference of Computer Vision*, pages 492–499, 1998.

[161] A. Said and W. A. Pearlman. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. on Circuits and Systems for Video Techn.*, 6:243–250, 1996.

[162] T. Schell and A. Uhl. New models for generating optimal wavelet-packet-tree-structures. *Proceedings of the 3rd IEEE Benelux Signal Processing Symposium*, pages 225–228, 2002.

[163] T. Schoenemann and D. Cremers. *Near Real-Time Motion Segmentation Using Graph Cuts*. 2006. 455-464 pp.

[164] H. Schwarz, D. Marpe, and T. Wiegand. The scalable H.264/MPEG4-AVC extension: Technology and applications. *European Symposium on Mobile Media Delivery,EuMob'06*, Alghero, Italy, Sept., 2006.

[165] N. Sebe, C. Lamba, and M.S Lew. An overcomplete discrete wavelet transform for video compression. *Proceedings of IEEE International Conference on Multimedia and Expo, ICME '02*, 1:641–644, 2002.

[166] A. Secker and D. Taubman. Highly scalable video compression with scalable motion coding. *IEEE Transactions on Image Processing*, 13:1029–1041, 2004.

[167] J. Sembiring, M. Nakabayashi, K. Soemintapoera, and K. Akizuki. Image compression using zerotrees of wavelet packets in rate-distortion sense. *IEEE 9th Asia-Pacific Conference on Communications*, 2:822–824, 2003.

[168] B. Shahraray. Scene change detection and content-based sampling of video sequences. *SPIE*, 2419, 1995.

[169] P. Greg Sherwood and Kenneth Zeger. Progressive image coding for noisy channels. *IEEE Signal Processing Letters*, 4(7):189–191, July 1997.

[170] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[171] Mikael Skoglund. On channel-constrained vector quantization and index assignment for discrete memoryless channels. *IEEE Transactions on Information Theory*, 45(17):2615–2622, November 1999.

[172] G. J. Sullivan and R. L. Baker. Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks. *in Proceedings of GLOBECOM'91*, pages 85–90, 1991.

[173] W. Sweldens. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In A. F. Lain and M. Unser, editors, *Wavelet Applications in Signal and Image Processing III*, pages 68–79. Proceedings of SPIE, vol. 2569, 1995.

[174] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, 3:186–200, 1996.

[175] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal of Mathematical Analysis*, 29:511–546, 1997.

[176] D. Taubman. High scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7):1158–1170, July 2000.

[177] D. Taubman and A. Zakhor. Multi-rate 3-D subband coding of video. *IEEE Trans. on Image Proc.*, 3:572–588, 1994.

[178] D. S. Taubman and M.W. Marcellin. *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer, Boston,MA, 2002.

[179] J. Y. Tham, S. Ranganath, and A.A Kassim. Highly scalable wavelet-based video codec for very low bit-rate environment. *IEEE Journal on Selected Areas in Communications*, 16:12–27, 1998.

[180] C. Tillier and B. Pesquet-Popescu. 3D, 3-band, 3-tap temporal lifting for scalable video coding. In *Proceedings of the IEEE International Conference on Image Processing*, Barcelona, Spain, Sept. 2003.

[181] C. Tillier, B. Pesquet-Popescu, and M. van der Schaar. Bidirectional predict-update 3-band schemes. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Canada, May 2004.

[182] C. Tillier, B. Pesquet-Popescu, and M. van der Schaar. Improved update operators for lifting-based motion-compensated temporal filtering. *IEEE Signal Processing Letters*, 12(2), 2005.

[183] C. Tillier, B. Pesquet-Popescu, and M. van der Schaar. 3-band temporal structures for scalable video coding. *IEEE Transitions on Image Processing*, 15:2545–2557, 2006.

[184] C. Tillier, B. Pesquet-Popescu, and M. van der Schaar. Weighted average spatio-temporal update operator for subband video coding. *ICIP*, Singapore, Oct. 2004.

[185] B.U. Toreyin, M. Trocan, E. Cetin, and B. Pesquet-Popescu. Linear and nonlinear temporal prediction employing lifting structures for scalable video coding. *In Proceedings of the IEEE European Signal Processing Conference, EUSIPCO'06*, 2006, Florence, Italy.

[186] B.U. Toreyin, M. Trocan, B. Pesquet-Popescu, and E. Cetin. LMS-based adaptive prediction for scalable video coding. *In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'06*, 2006, Toulouse, France.

[187] A.M. Tourapis, O.C. Au, and M.L. Liou. Fast block-matching motion estimation using predictive motion vector field adaptive search technique (PMVFAST). *ISO/IEC JTC1/SC29/WG11 MPEG2000/M5866, Noordwijkerhout, NL*, March, 2000.

[188] M. Trocan and B. Pesquet-Popescu. Video coding with fully separable wavelet and wavelet packet transforms. *Proc. of the VCIP - IST/SPIE Symposium on Electronic Imaging*, San jose, USA, 2007.

[189] M. Trocan and B. Pesquet-Popescu. Scene-cut processing in motion-compensated temporal filtering. *ACIVS, Anwertpen, Belgium*, Sept. 2005.

[190] M. Trocan, C. Tillier, and B. Pesquet-Popescu. A sliding window implementation of the 5-band motion compensated temporal lifting scheme. *In Proceedings of the IEEE International Workshop on Nonlinear Signal and Image Processing, NSIP'07*, 2007, Bucharest, Romania.

[191] M. Trocan, C. Tillier, and B. Pesquet-Popescu. Joint wavelet packets for group of frames in MCTF. *Proc. of the SPIE Conference on Wavelet Applications in Signal and Image Processing*, San Diego, USA, 2005.

[192] M. Trocan, C. Tillier, B. Pesquet-Popescu, and M. van der Schaar. A 5-band temporal lifting scheme for video surveillance. *IEEE 8th Workshop on Multimedia Signal Processing*, pages 278–281, 2006.

[193] B.T. Truong, C. Dorai, and S. Venkatesh. Improved fade and dissolve detection for reliable video segmentation. *Proceedings of the IEEE International Conference on Image Processing*, 3: 961–964, 2000.

[194] Y. Tsaig. *Automatic Segmentation of Moving Objects in Video Sequences*. PhD thesis, Department of Computer Science, School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel.

[195] D. Turaga and M. van der Schaar. Unconstrained temporal scalability with multiple reference and bi-directional motion compensated temporal filtering. doc. m8388, Fairfax MPEG meeting, 2002.

[196] D. Turaga, M. van der Schaar, and B. Pesquet-Popescu. Temporal prediction and differential coding of motion vectors in the MCTF framework. In *Proceedings of the IEEE International Conference on Image Processing*, Barcelona, Spain, Oct. 2003.

[197] M. van der Schaar and Y. Andreopoulos. Rate-distortion-complexity modeling for network and receiver aware adaptation. *IEEE Transactions on Multimedia*, 7:471– 479, 2005.

[198] M. van der Schaar and D. S. Turaga. Unconstrained motion compensated temporal filtering (UMCTF) framework for wavelet video coding. In *Proc. of IEEE ICASSP*, 2003.

[199] O. Veksler. Image segmentation by nested cuts. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 339–344, 2000.

[200] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, and P.L. Dragotti. Directionlets: Anisotropic multi-directional representation with separable filtering. *IEEE Trans. on Image Proc.*, 15: 1916–1933, 2006.

[201] J.D. Villasenor, B. Belzer, and J. Lico. Wavelet filter evaluation for image compression. *IEEE Transactions on Image Processing*, 4:1053–1060, 1995.

[202] Emanuele Viterbo and Joseph Boutros. A universal lattice code decoder for fading channels. *IEEE Transactions on Information Theory*, 45(5):1639–1642, July 1999.

[203] M. Wakin, J. Romberg, Hyeokho Choi, and R. Baraniuk. Geometric tools for image compression. *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, 2:1725 – 1729, 2002.

[204] J. Wang, H. Lu, G. Eude, and Q. Liu. Moving object segmentation using graph cuts. *Proceedings of ICSP'04. 7th International Conference on Signal Processing*, 1:777–780, 2004.

[205] Y. Wang, S. Cui, and J.E. Fowler. 3D video coding with redundant-wavelet multihypothesis. *IEEE Transactions on Circuits and Systems for Video Technology*, 16:166–177, 2006.

[206] S. Weiss, M. Harteneck, and R. W. Stewart. On implementation and design of filter banks for subband adaptive systems. *IEEE Workshop Signal Processing Systems*, pages 172–181, October 1998.

[207] M.V. Wickerhauser and A.K. Peters. *Adapted wavelet analysis from theory to software*. Wellesley Mass., 1994.

[208] V.M. Wickerhauser and R.R. Coifman. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38:713–718, 1992.

[209] Y. Wu and J.Woods. MC-EZBC Video Proposal from RPI. Technical Report MPEG04/M10569/S15, ISO/IEC JTC1/SC29/WG11, 2004.

[210] Y. Wu and J. W. Woods. Recent improvements in MC-EZBC video coder. Technical Report JTC1/SC29/WG11/M10396, MPEG (Moving Pictures Experts Group), Hawaii, USA, 2003.

[211] R. Xiong, X. Ji, D. Zhang, J. Xu, G. Pau, M. Trocan, S. Brangoulo, and V. Bottreau. Vidwav wavelet video coding specifications. Technical Report doc. M12339, ISO/IEC JTC1/SC29/WG11, July, 2005.

[212] R. Xiong, X. Ji, D. Zhang, J. Xu, G. Pau, M. Trocan, S. Brangoulo, and V. Bottreau. Vidwav wavelet video coding specifications. *ISO/IEC/JTC1/SC29/WG11 doc. M12339*, July 2005, Poznan, Poland.

[213] R. Xiong, F. Wu, S. Li, Z. Xiong, and Y.-Q. Zhang. Exploiting temporal correlation with adaptive block-size motion alignment for 3D wavelet coding. In *Proc. of SPIE Visual Communications and Image Processing*, San Jose, CA, 2004.

[214] R. Xiong, F. Wu, J. Xu, S. Li, and Y.-Q. Zhang. Barbell lifting wavelet transform for highly scalable video coding. *Proc. Picture Coding Symposium '04*, San Francisco, CA, USA, 2004.

[215] J. Xu and B. Pesquet-Popescu. Exploration experiments in wavelet video coding. *Video - AHG on further Exploration in Wavelet Video Coding, ISO/IEC JTC1/SC29/WG11, doc. N7098*, Busan, April 2005.

[216] J. Xu, Z. Xiong, S. Li, and Y. Zhang. Three-dimensional embedded subband coding with optimized truncation (3D-ESCOT). *Applied and Computational Harmonic Analysis*, 10:290–315, 2001.

[217] J. Ye and M. van der Schaar. Fully scalable 3-D overcomplete wavelet video coding using adaptive motion compensated temporal filtering. *Proc. SPIE Video Communications and Image Processing (VCIP'03)*, 2003.

[218] Y. Zhan, M. Picard, B. Pesquet-Popescu, and H. Heijmans. Long temporal filters in lifting schemes for scalable video coding. doc. m8680, Klagenfurt MPEG meeting, July 2002.

[219] Y. Zhang, X. Ji, D. Zhao, and W. Gao. Video coding by texture analysis and synthesis using graph cut. *In Proceedings of Pacific-Rim Conference on Multimedia, China*, 2006.

[220] S. Zhu and K.-K. Ma. A new diamond search algorithm for fast block matching motion estimation. *IEEE Transactions on Image Processing*, 9:287–290, 2000.

[221] F. Ziliani. The importance of scalability in video surveillance architectures. *The IEE International Symposium on Imaging for Crime Detection and Prevention*, pages 29–32, 2005.