



# Thèse

**présentée pour obtenir le grade de docteur**

**de TÉLÉCOM ParisTech**

**et SUP'COM-Tunis, Université de Carthage**

**Spécialité : Signal et Images**


# Mounir KAANICHE


## Schémas de lifting vectoriels adaptatifs et applications à la compression d'images stéréoscopiques

**Soutenue à TÉLÉCOM ParisTech le 2 Décembre 2010 devant le jury composé de**

| | |
|---|---|
| **Marc Antonini** | **Président** |
| **James E. Fowler** | **Rapporteur** |
| **Philippe Salembier** | **Rapporteur** |
| **Béatrice Pesquet-Popescu** | **Directrice de thèse** |
| **Amel Benazza-Benyahia** | **Directrice de thèse** |
| **Jean-Christophe Pesquet** | **Directeur de thèse** |

# Appendix A

# Résumé en français

## A.1 Introduction : Contexte de la thèse

Le principe de la stéréovision consiste à utiliser plusieurs images d'une même scène prises sous différents points de vue. Dans le cas où deux caméras (gauche et droite) sont utilisées, une paire d'images stéréo est acquise. Ce type d'images, appelées images stéréoscopiques, est utilisé dans de nombreuses applications telles que la télé-présence en vidéoconférence [Feldmann *et al.*, 2010] et la reconstruction de modèles 3D en télédétection [Tsutsui *et al.*, 2007].

De ce fait, un grand intérêt a été porté aux images stéréoscopiques. Cependant, le flot de données généré est assez élevé. A titre indicatif, le satellite spot5 permet de couvrir une zone importante de 120 km $\times$ 60 km, et une seule image représente environ 12000 $\times$ 12000 pixels [Poli *et al.*, 2004]. Pour cela, l'opération de compression s'avère nécessaire que ce soit pour leur stockage ou leur transmission.

L'objectif fixé dans cette thèse est double.

Dans une première partie, nous nous intéressons au développement et à l'analyse de nouvelles méthodes de codage sans perte progressif des images stéréoscopiques en s'appuyant sur les schémas de lifting. En effet, le codage sans perte progressif combine les contraintes de compression sans perte et la progressivité. La reconstruction progressive est recommandée pour les applications de transmission sur les réseaux, et le codage sans perte est exigé pour diverses applications telles que la télédétection.

Dans une deuxième partie, nous nous focalisons sur les aspects d'optimisation de tous les opérateurs impliqués dans une structure de lifting 2D non séparable afin de proposer des

schémas de codage mieux adaptés au contenu du signal d'entrée.

Notons d'une part que l'étude des images stéréoscopiques à des fins de reconstruction 3D ainsi que la compression des images multi-composantes, notamment les images multi-spectrales, a fait l'objet de nombreuses études au sein de l'unité de recherche en imagerie satellitaire et ses apllications (URISA, SUP'COM-Tunis). D'autre part, le codage des images multivues est un des thèmes de recherche au sein du département de traitement du signal et des images (TSI, Telecom ParisTech). De ce fait, une convention de thèse en cotutelle s'est établie entre ces deux écoles, sous la direction de Mme Béatrice Pesquet Popescu et Mme Amel Benazza-Benyahia, en s'intéressant à la compression des images stéréoscopiques. De plus, une collaboration ancienne et intense reliant ces deux écoles et l'université Paris-Est sur le thème de l'analyse d'images à l'aide d'ondelettes m'a offert l'occasion de mener ces travaux de thèse avec Mr Jean Christophe Pesquet, membre du Laboratoire d'Informatique de l'Institut Gaspard Monge (LabInfo-IGM), responsable de l'équipe Signal et Communications.

## A.2 Etat de l'art

Dans cette section, nous dressons tout d'abord un rapide état de l'art sur les méthodes les plus répandues dans la littérature pour la compression d'images stéréoscopiques. Nous fournissons égalament le principe des schémas de lifting afin de nous aider à mieux situer nos principales contributions développées dans cette thèse.

### A.2.1 Méthodes de codage des images stéréoscopiques

Le codage repose principalement sur la réduction de la redondance. En stéréoscopie, deux types de redondances co-existent : les similarités intra-image, qui se manifestent au niveau des régions uniformes, et les similarités inter-images, qui sont dues au fait que les images stéréoscopiques correspondent à une même scène 3D. Ainsi, un schéma de codage conjoint permettant d'exploiter ces deux types de redondance, en se basant sur le principe de la compensation de la disparité, sera plus efficace qu'un codage séparé de chacune des deux images [Boulgouris, Strintzis, 2002]. L'approche de base du codage conjoint repose sur les étapes suivantes.

Tout d'abord, une étape d'estimation de la disparité est effectuée. Celle-ci peut être réalisée au niveau pixel ou bien au niveau bloc. Cependant, dans le cas des applications de compression, cette étape est assez souvent effectuée au niveau bloc, en utilisant l'algorithme d'appariement de blocs.
Ensuite, l'image droite (considérée comme image cible) est prédite en utilisant l'image gauche (considérée comme image de référence) compensée en disparité. La différence entre l'image droite originale et celle prédite constitue la nouvelle image, appelée image résiduelle.
Enfin, l'image de référence, le champ de disparité et l'image résiduelle générée sont codés. Le champ de disparité est comprimé en utilisant un codage prédictif MICD suivi par un codage entropique. Par contre, le codage de l'image de référence et de l'image résiduelle est souvent effectué en utilisant certaines transformations telles que la transformation en ondelettes (TO) et la transformation en cosinus dicrete (DCT). Après avoir appliqué les différentes décompositions à la paire d'images stéréo, il est nécessaire de transcrire en bits les coefficients d'ondelettes en utilisant des codeurs progressifs afin d'assurer la possibilité de reconstruction progressive au décodeur [Taubman, 2000].
Bien que les méthodes conventionnelles de codage conjoint d'images stéréo préconisées dans la littérature s'articulent autour des étapes mentionnées ci-dessus, elles peuvent être classées en trois catégories. La première catégorie de méthodes vise à améliorer l'étape d'estimation (et compensation) de la disparité afin d'exploiter au mieux les redondances inter-images [Woo, Ortega, 1997; Woo, Ortega, 2000]. La deuxième catégorie propose d'améliorer le codage entropique des coefficients obtenus dans le domaine transformé [Palfner *et al.*, 2002; Ellinas, Sangriotis, 2006]. La troisième catégorie de méthodes s'intéresse au choix des transformations appliquées à l'image de référence et l'image résiduelle [Frajka, Zeger, 2003; Darazi *et al.*, 2009; Maalouf, Larabi, 2010]. Nos travaux s'inscrivent dans cette dernière catégorie.

### A.2.2 Principe des schémas de lifting

Parmi les différentes transformations développées dans la littérature (DCT, TO, etc), un intérêt soutenu a été porté aux TO et plus particulièrement aux schémas de lifting depuis leur développement par Sweldens [Sweldens, 1995; Sweldens, 1996]. Ceux-ci sont des outils bien appropriés à la représentation d'une image sur plusieurs niveaux de résolution tout en

assurant la propriété de reconstruction parfaite.

Afin de présenter le schéma de lifting dans sa forme la plus basique, nous allons considérer le cas d'un signal 1D noté $s_j(n)$ correspondant à une approximation du signal initial $s_0(n)$ au niveau de résolution $j \in \mathbb{N}^*$. Un schéma de lifting peut être décomposé en trois parties comme on peut le voir dans la figure 3.7(a).

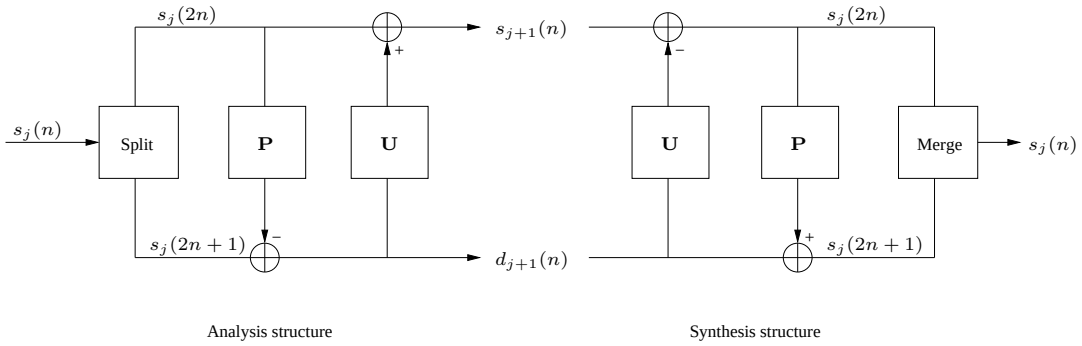



Figure A.1: Schéma générique d'une structure de lifting : (a) structure d'analyse, (b) structure de synthèse.

- **Transformation polyphase :** Tout d'abord, le signal $s_j(n)$ est séparé en deux sous-ensemble disjoints : le premier est formé des échantillons d'indices impairs $s_j(2n+1)$ et le second des échantillons d'indices pairs $s_j(2n)$.

- **Etape de prédiction :** Ensuite, chaque échantillon de l'un des deux sous-ensembles (disons le sous-ensemble d'échantillons d'indices impairs) est prédit à partir des échantillons d'indices pairs voisins grâce à un filtre linéaire transverse. L'erreur de prédiction $\tilde{d}_{j+1}(n)$ constitue les coefficients des détails résiduels à la résolution $(j+1)$ :

$$\tilde{d}_{j+1}(n) = s_j(2n+1) - \mathbf{p}_j^\top \mathbf{s}_j(n) \tag{A.1}$$

  où

  - $\mathbf{p}_j$ est le vecteur des poids du filtre de prédiction,
  - $\mathbf{s}_j(n) = \Big(s_j(2n-2k)\Big)_{k\in\mathcal{P}_j}$ est le vecteur de référence contenant les échantillons pairs utilisés dans l'étape de prédiction,
  - $\mathcal{P}_j$ représente le support du prédicteur.

- **Etape de mise à jour :** Enfin, les échantillons pairs sont lissés en utilisant les coefficients détails calculés précédemment. Le signal résultant $\tilde{s}_{j+1}(n)$, appelé signal

d'approximation, constitue une version grossière du signal original :

$$\tilde{s}_{j+1}(n) = s_j(2n) + \mathbf{u}_j^\top \tilde{\mathbf{d}}_{j+1}(n) \tag{A.2}$$

où

- $\mathbf{u}_j$ est le vecteur des poids du filtre de lissage (appelé aussi filtre de mise à jour),
- $\tilde{\mathbf{d}}_{j+1}(n) = \left(\tilde{d}_{j+1}(n-k)\right)_{k\in\mathcal{U}_j}$ est le vecteur de référence contenant les coefficients de détails utilisés dans l'étape de mise à jour,
- $\mathcal{U}_j$ représente le support de l'opérateur de mise à jour.

Cette transformation 1D est généralement étendu au cas 2D de manière séparable. En effet, le schéma de lifting s'applique aux lignes puis aux colonnes (ou inversement) donnant lieu ainsi à une sous-bande d'approximation et trois sous-bandes de coefficients de détails orientés horizontalement, verticalement et diagonalement.

## A.3 Contributions

### A.3.1 Nouvelles approches basées sur les schémas de lifting vectoriels

Le concept de lifting vectoriel a été développé par Benazzza et *al.* [Benazza-Benyahia *et al.*, 2002] dans le cadre de la compression d'images multispectrales. Notre première contribution consiste à étendre ce concept au cas de la compression d'images stéréoscopiques. L'idée de base est de se servir de l'image de référence pour coder l'image droite en utilisant le champ de disparité. En ce sens, contrairement aux méthodes classiques, la méthode proposée ne génère aucune image résiduelle mais deux représentations multirésolutions compactes de l'image gauche et l'image droite. A ce propos, nous avons considéré deux versions de Schéma de Lifting Vectoriel (*Vector Lifting Schemes*, VLS, en anglais), qui seront détaillées par la suite.

#### VLS-I

Le premier schéma, noté VLS-I, est décrit par la Fig. A.2.
Ainsi, l'image gauche $I_j^{(l)}$, sélectionnée comme image de référence, est codée en mode intra en utilisant une structure de lifting classique P-U [Calderbank *et al.*, 1998]. Par contre, l'image droite $Ij^{(r)}$ est codée en appliquant une structure de lifting mettant en œuvre un

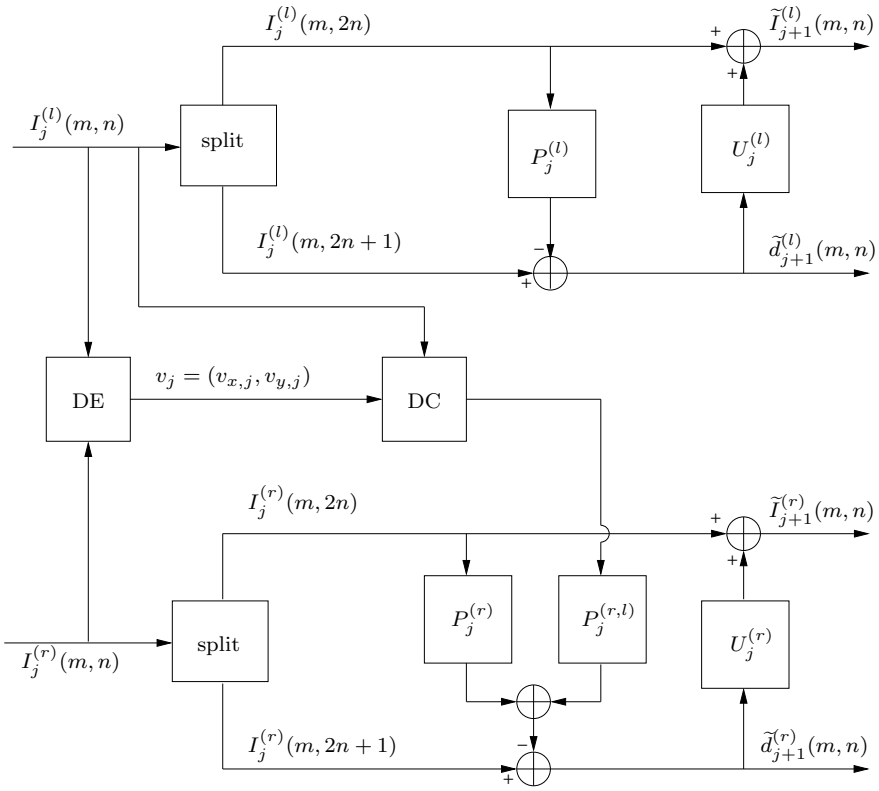



Figure A.2: Principe du VLS-I.

prédicteur hybride qui exploite les redondances inter-images via le champ de disparité $\mathbf{v}_j$. A priori, toute méthode d'estimation de la disparité pourrait convenir. Toutefois, pour éviter une estimation de la disparité à chaque niveau de résolution $j$, nous proposons de la faire une seule fois sur l'image pleine résolution et d'appliquer par la suite une décimation et une division d'un facteur de $2^j$ sur le champ de disparité initial, puisque les dimensions des sous-bandes au niveau de résolution $j$ correspondent à celles de l'image initiale divisées par $2^j$.

Un petit inconvénient du schéma proposé est qu'il produit un effet de "fuite" dans l'étape de mise à jour, dans le sens où l'information venant de la sous-bande d'approximation $I_j^{(l)}$, qui est utilisée dans l'étape de prédiction pour le calcul du signal de détail $\tilde{d}_{j+1}^{(r)}$, est réinjectée dans l'étape de mise à jour pour calculer les coefficients du signal d'approximation $\tilde{I}_{j+1}^{(r)}$ de l'image droite.

### VLS-II

Pour remédier au problème mentionné ci-dessus, un second schéma, noté VLS-II, est proposé. En effet, l'image de référence est toujours codée en mode intra de façon similaire à VLS-I. Par contre, l'image droite est codée en adoptant une nouvelle structure P-U-P comme le montre la Fig. 4.3.

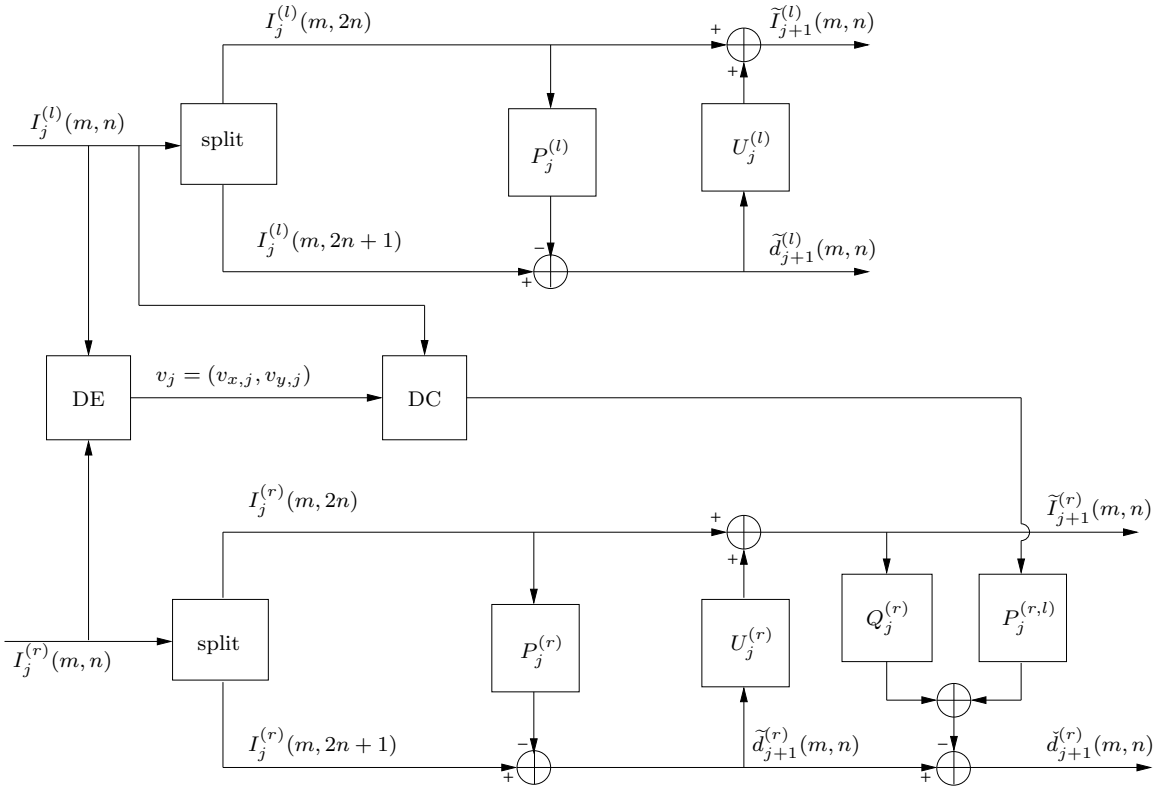



Figure A.3: Principe du VLS-II.

Pour cela, une première étape de prédiction intra-image est appliquée pour calculer un signal de détail intermédiaire, $\check{d}_{j+1}(m,n)$, qui va servir au calcul du signal d'approximation $\tilde{s}_{j+1}(m,n)$. Ensuite, une deuxième étape de prédiction inter-images est effectuée pour générer le signal de détail final $\tilde{d}_{j+1}(m,n)$.

L'avantage de cette structure est que dans le cas idéal où l'image gauche est égale à l'image droite, le signal détail de l'image droite $d_{j+1}^{(r)}$ est nul alors que le signal approximation $I_{j+1}^{(r)}$ est égal à celui de $I_{j+1}^{(l)}$. Pour cette raison, nous avons ajouté l'étape suivante: au dernier niveau de résolution $j=J$, au lieu de coder le signal approximation $I_J^{(r)}$, on procède au

codage de la sous bande résiduelle définie par :

$$e_J^{(r)}(m_x, m_y) = I_J^{(r)}(m_x, m_y) - \lfloor \sum_{k \in \mathcal{P}_J^{(r,l)}} p_{J,k}^{(r,l)} I_J^{(l)}(m_x + v_{x,J}(m_x, m_y), m_y + v_{y,J}(m_x, m_y) - k) \rfloor. \tag{A.3}$$

où $\mathcal{P}_J^{(r,l)}$ et $p_{J,k}^{(r,l)}$ représentent respectivement le support et les poids du vecteur de prédiction inter-sous-bandes approximations et $\lfloor \cdot \rfloor$ désigne l'opérateur partie entière.

**Analyse théorique des schémas proposés**

Pour montrer l'intérêt des structures proposées, nous avons développé une analyse théorique en termes de performances statistiques. Cette analyse a été menée dans le cadre général de processus aléatoires stationnaires. Ensuite, pour illustrer les performances de chaque structure, nous avons considéré des signaux autorégressifs d'ordre 1 mutuellement indépendants.

Notons que l'étude théorique menée et les résultats expérimentaux obtenus (en termes de débit final et courbes débit-distorsion) sont présentés en détail dans [Kaaniche *et al.*, 2007] et [Kaaniche *et al.*, 2009a].

### A.3.2 Intégration d'une carte de disparité dense dans les schémas de codage d'images stéréo

Comme nous l'avons déja mentionné, les méthodes de codage conjoint reposent principalement sur l'étape de l'estimation et la compensation de la disparité afin d'exploiter les redondances inter-images. Cette étape est souvent effectuée en utilisant une approche par bloc, et plus précisément la technique du *block-matching*. Cependant, cette approche souffre de certaines limitations puisqu'elle n'est pas adapté aux zones peu texturées et au voisinage des discontinuités. Ceci peut produire une carte de disparité imprécise et, par conséquent, cela risque d'affecter le processus de compensation de disparité.

Pour cette raison, nous avons proposé dans cette thèse d'utiliser une méthode d'estimation qui génère une carte de disparité dense et lisse [Miled *et al.*, 2006b; Miled *et al.*, 2009b]. L'idée de base de cette approche consiste à formuler le problème d'estimation de la disparité comme un problème de programmation convexe visant à minimiser une fonction objectif convexe sur l'intersection d'ensembles convexes. Ces ensembles sont associés à des contraintes convexes modélisant des informations a priori tenant compte des propriétés du

champ à estimer et des données observées. La contrainte la plus importante consiste à appliquer une borne supérieure sur la variation totale du champ estimé pour des fins de régularisation. Celle-ci permet de fournir un champ lisse tout en respectant les discontinuités présentes dans la scène.

Comme une carte dense est généralement très coûteuse en terme de débit, elle est ensuite segmentée en blocs de taille $16 \times 16$. Puis, pour chaque bloc, on teste son homogénéité en calculant sa variance. Enfin, lorsque la variance est inférieure à un certain seuil, les valeurs de disparité de ce bloc seront remplacées par leur valeur moyenne ou médiane. Dans le cas contraire, on subdivise ce bloc en 4 sous-blocs et on réitère le même traitement sur chaque bloc. Notons que la carte segmentée, ainsi que l'information correspondante à la taille du bloc sont codées, par la suite, en utilisant l'encodeur H.264.

L'intérêt de l'utilisation de cette carte dense segmentée dans les schémas de codage d'images stéréo a été récemment présenté dans [Kaaniche *et al.*, 2009b]. De plus, une évaluation des performances de la méthode d'estimation dense dans le contexte du codage des séquences vidéo stéréoscopique est aussi présentée dans [Daribo *et al.*, 2009].

### A.3.3 Schéma de lifting non séparable et méthodes d'optimisation des filtres

Il est important de noter que les structures des schémas de lifting que nous avions développées sont séparables. En effet, un filtrage 1D est appliqué suivant les lignes puis suivant les colonnes (ou inversement) pour générer les coefficients d'ondelettes. Cependant, un tel traitement séparable rend la décomposition peu efficace dans le cas des images présentant des contours qui ne sont ni horizontaux ni verticaux. De plus, cette contrainte de séparabilité limite le nombre de degrés de liberté dans le choix des opérateurs de prédiction et de mise à jour. Pour cela, nous nous sommes intéressés dans une deuxième partie aux schémas de lifting 2D non séparables, tout en nous focalisant sur les aspects d'optimisation des différents filtres mis en jeu.

#### Structure de lifting 2D non séparable

La structure de lifting 2D considérée est composée de trois étapes de prédiction suivies par une étape de mise à jour, comme le montre la Fig. 6.1. Le principe général de cette transformation 2D non séparable consiste à décomposer une image $x_j(m,n)$ en

quatre composantes polyphase $x_{0,j}(m,n) = x_j(2m,2n)$, $x_{1,j}(m,n) = x_j(2m,2n+1)$, $x_{2,j}(m,n) = x_j(2m+1,2n)$, et $x_{3,j}(m,n) = x_j(2m+1,2n+1)$ pour générer une sous-bande d'approximation $x_{j+1}(m,n)$ et les sous-bandes des détails orientés horizontalement $x_{j+1}^{(HL)}(m,n)$, verticalement $x_{j+1}^{(LH)}(m,n)$ et diagonalement $x_{j+1}^{(HH)}(m,n)$.

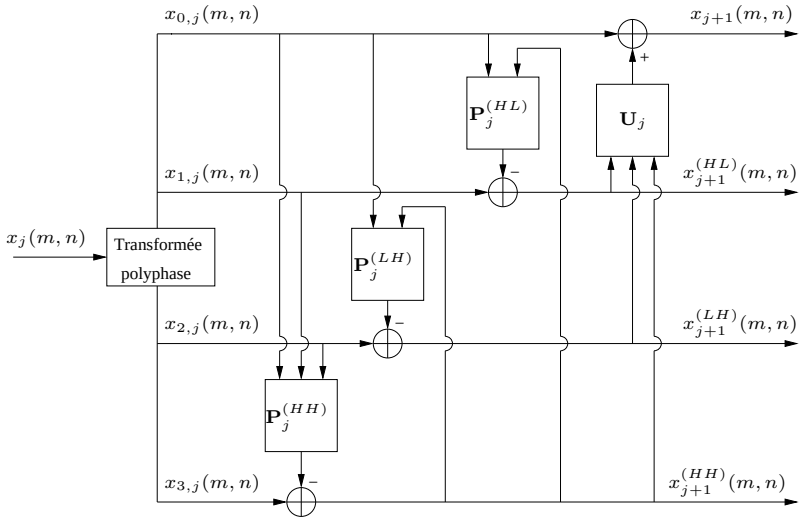



Figure A.4: Structure de lifting 2D non séparable.

Comme les performances de codage des schémas de lifting dépendent généralement des opérateurs de prédiction $\left(\mathbf{p}_j^{(o)}\right)_{o\in\{HL,LH,HH\}}$ et de mise à jour $\mathbf{u}_j$ mis en jeu, une grande importance a été accordée à l'adaptation des opérateurs de lifting afin de réaliser des schémas de lifting mieux *adaptés* au *contenu* du signal d'entrée [Pesquet-Popescu, 1999].

### Optimisation des filtres de prédiction

Certains travaux d'optimisation se sont intéressés à l'optimisation des filtres de prédiction en minimisant l'entropie des signaux détails [Benazza-Benyahia *et al.*, 2007]. Cependant, l'entropie du signal de détail étant une fonction implicite des coefficients du filtre de prédiction à optimiser, sa minimisation ne peut se faire que par des algorithmes itératifs. En particulier, l'algorithme de Nelder-Mead a été utilisé dans [Benazza-Benyahia *et al.*, 2007] pour minimiser l'entropie des signaux détails. Toutefois, ces algorithmes d'optimisation présentent quelques inconvénients. Tout d'abord, ils sont itératifs et peuvent converger vers des minima locaux. De plus, ils sont trop lourds en temps de calcul.

Pour remédier à ces problèmes, une autre alternative reposant sur la minimisation de la variance du signal de détail a été assez souvent retenue [Pesquet-Popescu, 1999; Boulgouris

*et al.*, 2001; Gouze *et al.*, 2004]. En effet, de point de vue de la théorie de l'information [Cover, Thomas, 1991], un signal de faible variance peut être codé plus efficacement qu'un signal de forte variance. Pour cela, nous avons choisi d'optimiser chaque filtre de prédiction $\mathbf{p}_j^{(o)}$, avec $o \in \{HH, HL, LH\}$, en minimisant le critère suivant :

$$\begin{aligned} &\forall\, o \in \{HL, LH, HH\} \quad \text{and} \quad i \in \{1,2,3\}, \\ &\mathcal{J}_1(\mathbf{p}_j^{(o)}) = \sum_{m,n} \Big( x_{j+1}^{(o)}(m,n) \Big)^2 \\ &\qquad = \sum_{m=1}^{M_j} \sum_{n=1}^{N_j} \Big( x_{i,j}(m,n) - (\mathbf{p}_j^{(o)})^\top \tilde{\mathbf{x}}_j^{(o)}(m,n) \Big)^2 \end{aligned} \tag{A.4}$$

où $x_{i,j}(m,n)$ est la $(i+1)^{\text{ème}}$ composante polyphase à prédire, $\tilde{\mathbf{x}}_j^{(o)}(m,n)$ est un vecteur de référence contenant les échantillons utilisés dans l'étape de prédiction de $x_{i,j}(m,n)$, $\mathbf{p}_j^{(o)}$ est le vecteur de prédiction à optimiser, et $[M_j, N_j]$ correspondent aux dimensions de la sous-bande $x_j$ divisées par 2.

**Optimisation du filtre de mise à jour : minimisation des effets d'aliasing**

L'optimisation du filtre de mise à jour est un problème plus complexe que celui du filtre de prédiction et il a été relativement peu abordé dans la littérature [Pesquet-Popescu, 1999; Gouze *et al.*, 2004; Piella *et al.*, 2006; Kaaniche *et al.*, 2010; Kaaniche *et al.*, 2011]. En effet, un opérateur de mise à jour efficace retourne un signal basse fréquence offrant une représentation fidèle du signal original à une résolution inférieure. Généralement, l'image des coefficients est destinée à subir des dégradations plus ou moins fortes lorsque la transformation en ondelettes est suivie par un codage générant des pertes d'information. Pour cela, Gouze et *al.* [Gouze *et al.*, 2004] ont proposé d'optimiser le filtre de mise à jour afin d'assurer une reconstruction du signal la plus fiable possible à partir de la seule sous-bande basse fréquence. Plus précisément, le filtre de mise à jour $\mathbf{u}_j$ est conçu en minimisant l'erreur de reconstruction $\mathcal{J}_2$ entre l'image originale $x_j$ et l'image reconstruite $\hat{x}_j$ tout en supposant que les coefficients des détails sont nuls. Un tel critère est défini par :

$$\begin{aligned} \mathcal{J}_2(\mathbf{u}_j) &= \mathsf{E}[(x_j(m,n) - \hat{x}_j(m,n))^2] \\ &= \frac{1}{4}\Big( \mathsf{E}[(x_{0,j}(m,n) - \hat{x}_{0,j}(m,n))^2] + \mathsf{E}[(x_{1,j}(m,n) - \hat{x}_{1,j}(m,n))^2] \\ &\quad + \mathsf{E}[(x_{2,j}(m,n) - \hat{x}_{2,j}(m,n))^2] + \mathsf{E}[(x_{3,j}(m,n) - \hat{x}_{3,j}(m,n))^2] \Big). \end{aligned} \tag{A.5}$$

Dans ce contexte, nous avons proposé une nouvelle méthode d'optimisation visant à réduire les phénomènes indésirables de repliement [Kaaniche *et al.*, 2010]. Plus précisément, le filtre de mise à jour est conçu de façon que sa sortie soit le plus proche possible de celle d'un filtre passe-bas idéal. Notons qu'un tel critère tient compte des caractéristiques du signal d'entrée et ne se limite pas à une simple différence entre les filtres. Ainsi, le filtre de mise à jour est optimisé en minimisant le critère quadratique suivant :

$$\mathcal{J}_3(\mathbf{u}_j) = \int_{-\pi}^{\pi} \left| F_j(e^{\imath\omega_1}, e^{\imath\omega_2}) - H(e^{\imath\omega_1}, e^{\imath\omega_2}) \right|^2 S_{x_j}(\omega_1, \omega_2) d\omega_1 d\omega_2 \tag{A.6}$$

où $S_{x_j}$ est la densité spectrale de puissance de l'image $x_j(m,n)$, $F_j$ et $H$ désignent respectivement la fonction de transfert du filtre passe-bas de la structure de lifting et du filtre passe-bas idéal.

On remarque que ce critère calcule l'erreur quadratique moyenne entre les sorties des filtres $F_j$ et $H$ *avant* l'étape de décimation. Cependant, afin de réduire la complexité de la méthode d'optimisation, il serait intéressant de minimiser l'erreur calculée *après* l'étape de décimation. Celle-ci est exprimée par :

$$\begin{aligned} \mathcal{J}_4(\mathbf{u}_j) &= \mathsf{E}\Big[\Big(x_{j+1}(m,n) - y_{j+1}(m,n)\Big)^2\Big] \\ &= \mathsf{E}\Big[\Big(x_{0,j}(m,n) + \mathbf{u}_j^\top \mathbf{x}_{j+1}(m,n) - y_{j+1}(m,n)\Big)^2\Big] \end{aligned} \tag{A.7}$$

où

- $\mathbf{u}_j = \Big(u_j^{(o)}(k,l)\Big)^\top_{(k,l)\in\mathcal{U}_j^{(o)}, o\in\{HL,LH,HH\}}$ est le vecteur des poids du filtre de mise à jour,
- $\mathbf{x}_{j+1}(m,n) = \Big(x_{j+1}^{(o)}(m-k,n-l)\Big)^\top_{(k,l)\in\mathcal{P}_{i,j}^{(o)}, o\in\{HL,LH,HH\}}$ est le vecteur de référence contenant les coefficients détails utilisés dans l'étape de mise à jour.
- $y_{j+1}(m,n) = \tilde{y}_j(2m,2n) = (h * x_j)(2m,2n)$, où $h$ désigne la réponse impulsionnelle du filtre passe-bas idéal.

Il est important de noter que nous avons montré dans [Kaaniche *et al.*, 2011] que les minimisations des critères $\mathcal{J}_3$ et $\mathcal{J}_4$ aboutissent à la même solution du filtre de mise à jour vérifiant le système d'équations linéaires suivant :

$$\mathsf{E}[\mathbf{x}_{j+1}(m,n)\mathbf{x}_{j+1}(m,n)^\top]\mathbf{u}_j = \mathsf{E}[y_{j+1}(m,n)\mathbf{x}_{j+1}(m,n)] - \mathsf{E}[x_{0,j}(m,n)\mathbf{x}_{j+1}(m,n)] \tag{A.8}$$

Bien que cette stratégie donne des performances, en terme de qualité de reconstruction, assez similaires à la méthode de l'état de l'art reposant sur la minimisation de l'erreur de reconstruction, notre technique d'optimisation présente l'avantage de la simplicité de sa mise en œuvre, tout en réalisant un gain d'un facteur multiplicatif de 3 près en temps de calcul [Kaaniche *et al.*, 2011].

#### Analyse théorique

Pour montrer l'intérêt de la méthode d'optimisation proposée, nous avons développé une analyse théorique pour un signal d'entrée autorégressif d'ordre 2 de coefficients de corrélation $\rho_{1,j}$ et $\rho_{2,j}$ dans la direction horizontale et verticale respectivement. L'analyse théorique menée a mis en évidence l'apport de l'optimisation du filtre de mise à jour en terme de réduction des effets d'aliasing. De plus, nous avons déterminé les expressions explicites des coefficients des filtres de prédiction et de mise à jour en fonction des coefficients de corrélation du modèle statistique considéré. Ces expressions théoriques sont intéressantes pour les raisons suivantes. D'une part, elles nous permettent de simplifier l'implémentation de la méthode d'optimisation proposée. D'autre part, le coût de transmission des coefficients des différents filtres sera considérablement réduit puisqu'il suffit d'envoyer les facteurs de corrélation $\rho_{1,j}$ et $\rho_{2,j}$, qui nous permettent de les calculer dans le cas idéal.

#### Optimisation par des critères parcimonieux

En poursuivant la piste d'optimisation des différents opérateurs de lifting, nous nous sommes intéressés dans la dernière partie de la thèse au développement de critères parcimonieux en remplaçant le critère usuel $\ell_2$ par des critères $\ell_1$.

#### Minimisation d'un critère $\ell_1$

Tout d'abord, nous avons présenté les raisons qui motivent la minimisation de critères $\ell_1$. Pour cela, au lieu de minimiser la variance des signaux de détails qui correspond à un critère quadratique, nous avons proposé d'optimiser chaque filtre de prédiction $\mathbf{p}_j^{(o)}$, avec $o \in \{HL, LH, HH\}$, en minimisant la norme $\ell_1$ du signal de détails $x_{j+1}^{(o)}$. Un tel critère

est exprimé par :

$$\begin{aligned} &\forall\, o \in \{HL, LH, HH\} \quad \text{and} \quad i \in \{1,2,3\}, \\ &\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(o)}) = \sum_{m,n} \left| x_{j+1}^{(o)}(m,n) \right| \\ &\qquad = \sum_{m=1}^{M_j} \sum_{n=1}^{N_j} \left| x_{i,j}(m,n) - (\mathbf{p}_j^{(o)})^\top \tilde{\mathbf{x}}_j^{(o)}(m,n) \right| \end{aligned} \tag{A.9}$$

Cependant, le problème majeur dans cette approche d'optimisation réside dans le fait que le critère (7.6) est non-différentiable. Pour cela, une grande classe d'algorithmes itératifs proximaux a été proposée pour résoudre ce type de problème de minimisation [Combettes, Pesquet, 2010]. Dans notre contexte, nous avons eu recours à l'algorithme de Douglas-Rachford qui a prouvé son efficacité dans la résolution de problèmes inverses de grande taille [Eckstein, Bertsekas, 1992].

### Minimisation de l'erreur de prédiction globale pondérée

Il est important de remarquer de la Fig. 6.1 que le signal de détails diagonaux $x_{j+1}^{(HH)}$ est utilisé dans la deuxième et la troisième étape de prédiction pour calculer les signaux de détails horizontaux $x_{j+1}^{(HL)}$ et verticaux $x_{j+1}^{(LH)}$. Par conséquent, l'optimisation du filtre $\mathbf{p}_j^{(HH)}$ résultant de la minimisation de la norme $\ell_1$ de $x_{j+1}^{(HH)}$ est sous-optimale. Pour cela, nous avons proposé par la suite d'optimiser le filtre $\mathbf{p}_j^{(HH)}$ en minimisant l'erreur de prédiction globale. Cette erreur est exprimée sous la forme de somme *pondérée* des normes $\ell_1$ des sous-bandes détails :

$$\mathcal{J}_{w\ell_1}(\mathbf{p}_j^{(HH)}) = \sum_{o \in \{HL,LH,HH\}} \sum_{m,n} \sqrt{w_j^{(o)}} \left| x_{j+1}^{(o)}(m,n) \right| \tag{A.10}$$

où les $w_j^{(o)}$, $o \in \{HL, LH, HH\}$, sont les poids des sous-bandes détails $x_{j+1}^{(o)}$.
Contrairement au critère précédent, Eq. (7.6), composé d'une norme $\ell_1$, le nouveau critère est constitué de trois normes $\ell_1$. Pour minimiser un tel critère, Eq. (7.17), nous avons utilisé l'algorithme de Douglas-Rachford formulé dans un espace *produit* [Combettes, Pesquet, 2010]. Une fois le filtre $\mathbf{p}_j^{(HH)}$ optimisé et défini, les deux autres filtres de prédiction $\mathbf{p}_j^{(HL)}$ et $\mathbf{p}_j^{(LH)}$ sont ensuite optimisés séparément en minimisant $\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(HL)})$ et $\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(LH)})$. Ceci s'explique par le fait que l'optimisation de chacun de ces deux filtres ne dépend pas de l'autre.

**Optimisation conjointe**

Comme l'optimisation du filtre $\mathbf{p}_j^{(HH)}$ est effectuée en minimisant la somme pondérée des normes $\ell_1$ des 3 sous-bandes détails, on peut constater que la solution optimale de $\mathbf{p}_j^{(HH)}$ va dépendre des filtres $\mathbf{p}_j^{(HL)}$ et $\mathbf{p}_j^{(LH)}$. D'autre part, il est clair que l'optimisation des filtres $\mathbf{p}_j^{(HL)}$ et $\mathbf{p}_j^{(LH)}$ dépend de $\mathbf{p}_j^{(HH)}$ puisque le signal $x_{j+1}^{(HH)}$ résultant de l'optimisation de $\mathbf{p}_j^{(HH)}$ est utilisé comme signal de référence dans les étapes d'optimisation de $\mathbf{p}_j^{(HL)}$ et $\mathbf{p}_j^{(LH)}$. Ainsi, on conclut que l'optimisation de $\mathbf{p}_j^{(HH)}$ dépend de $\mathbf{p}_j^{(HL)}$ et $\mathbf{p}_j^{(LH)}$, et inversement.

Pour cela, nous avons proposé une méthode d'optimisation *conjointe* qui consiste à optimiser de façon itérative les différents filtres. L'algorithme itératif est décrit de la manière suivante :

① Initialiser le numéro de l'itération $it$ à 0.

• Optimiser séparément les 3 filtres de prédiction $\mathbf{p}_j^{(HH)}$, $\mathbf{p}_j^{(LH)}$ et $\mathbf{p}_j^{(HL)}$ en minimisant $\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(HH)})$, $\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(LH)})$ et $\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(HL)})$, respectivement. Les solutions optimales obtenues seront désignées par : $\mathbf{p}_j^{(HH,it)}$, $\mathbf{p}_j^{(LH,it)}$ and $\mathbf{p}_j^{(HL,it)}$.

**• Optimiser le filtre de mise à jour en minimisant $\mathcal{J}_4(\mathbf{u}_j)$.**

**• Calculer les poids $w_j^{(o,it)}$ de chaque sous-bande de détails ainsi que l'erreur de prédiction globale pondérée (c'est à dire la somme pondérée des normes $\ell_1$ des sous-bandes de détails).**

② pour $it = 1, 2, 3, ...$

• Fixer $\mathbf{p}_j^{(LH)} = \mathbf{p}_j^{(LH,it-1)}$, $\mathbf{p}_j^{(HL)} = \mathbf{p}_j^{(HL,it-1)}$, et optimiser $\mathbf{P}_j^{(HH)}$ en minimisant $\mathcal{J}_{w\ell_1}(\mathbf{p}_j^{(HH)})$. Soit $\mathbf{p}_j^{(HH,it)}$ le nouveau premier filtre de prédiction optimal.

• Fixer $\mathbf{p}_j^{(HH)} = \mathbf{p}_j^{(HH,it)}$, et optimiser $\mathbf{P}_j^{(LH)}$ en minimisant $\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(LH)})$. Soit $\mathbf{p}_j^{(LH,it)}$ le nouveau deuxième filtre de prédiction optimal.

• Fixer $\mathbf{p}_j^{(HH)} = \mathbf{p}_j^{(HH,it)}$, et optimiser $\mathbf{P}_j^{(HL)}$ en minimisant $\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(HL)})$. Soit $\mathbf{p}_j^{(HL,it)}$ le nouveau troisième filtre de prédiction optimal.

**• Optimiser le filtre de mise à jour en minimisant $\mathcal{J}_4(\mathbf{u}_j)$.**

**• Calculer les poids $w_j^{(o,it)}$ de chaque sous-bande de détails ainsi que l'erreur de prédiction globale pondérée.**

③ Sélectionner les filtres optimaux à partir de l'itération donnant l'erreur de prédiction globale pondérée la plus faible.

Il est important de souligner ici que l'étape 3 de l'algorithme a été introduite puisque l'erreur de prédiction globale pondérée ne diminue pas nécessairement d'une itération $it$ à $it+1$. Ceci s'explique par le fait que les poids $w_j^{(o)}$ sont recalculés à chaque itération après l'optimisation de tous les filtres.

## A.4 Conclusion et perspectives

L'intérêt croissant des images stéréoscopiques dans beaucoup de champs d'application s'est traduit par la constitution de banques d'images de tailles énormes. Pour cela, l'objectif majeur de ce travail de thèse était de proposer et d'analyser de nouveaux schémas de codage d'images stéréoscopiques. Ces schémas permettent d'assurer la reconstruction progressive avec la possibilité de restitution exacte de la paire d'images.
Les principales contributions développées dans cette thèse peuvent être résumées comme suit.
Dans une première partie, nous avons proposé de nouveaux schémas de codage conjoint de la paire d'images stéréo reposant sur le concept du lifting vectoriel. En ce sens, contrairement aux méthodes classiques, la méthode proposée ne génère aucune image résiduelle mais deux représentations multirésolutions compactes de l'image gauche et l'image droite. De plus, notre schéma est adapté au contenu (intra et inter) de la paire stéréo. En effet, les opérateurs de prédiction sont optimisés à chaque étage du lifting en minimisant la variance des signaux de détails. Dans ce contexte, nous avons proposé deux schémas de liftng vectoriels (VLS-I et VLS-II). Une analyse théorique en termes de performances statistiques a été menée pour mettre en évidence l'intérêt des structures proposées.
Comme perspective à cette partie, les performances des schémas VLS peuvent être améliorées en tenant compte des problèmes d'occlusion. Plus précisément, il serait intéressant d'utiliser un schéma de codage hybride où les pixels occultés (resp. non occultés) seront codés en utilisant une étape de prédiction intra (resp. inter) images. De plus, l'extension de ces structures de lifting au cas du codage d'images multivues peut être envisagée .

Comme nous l'avions mentionné, les méthodes de codage conjoint de la paire d'images stéréo reposent principalement sur l'étape de l'estimation et la compensation de la disparité afin d'exploiter les redondances inter-images. Pour cela, nous avons proposé dans cette thèse d'intégrer une méthode récente d'estimation de disparité dans des applications

de codage d'images stéréoscopiques. Cette technique d'estimation, qui se base sur une formulation ensembliste du problème de mise en correspondance, utilise différentes contraintes convexes caractérisant toute propriété a priori sur le champ de disparité à estimer. Ainsi, les contraintes utilisées permettent de produire un champ lisse tout en respectant les discontinuités (objets) présentes dans la scène. Comme une carte de disparité dense est très coûteuse en terme de débit, nous avons proposé d'y appliquer un algorithme de segmentation, de sorte que les zones homogènes seront représentées par des blocs de grande taille et les zones de faible texture vont être représentées par des blocs de petite taille.

Les structures des schémas de lifting que nous avions utilisées sont séparables (c'est à dire elles reposent sur l'utilisation de filtres mono-dimensionnels (1D) pour calculer les coefficients d'ondelettes). Cependant, l'inconvénient des schémas conventionnels séparables est qu'ils ne sont pas très efficaces pour exploiter le caractère bi-bimensionnel (2D) des images. Pour cela, nous nous sommes intéressés dans une deuxième partie aux schémas de lifting 2D non séparables tout en nous focalisant sur les aspects d'optimisation des différents filtres mis en jeu. La structure de lifting 2D considérée est composée de trois étapes de prédiction suivies par une étape de mise à jour. Dans ce contexte, nous avons proposé une nouvelle méthode, visant à éliminer les phénomènes indésirables de repliement, pour la conception du filtre de mise à jour. Cette méthode consiste à optimiser ce filtre de façon à ce qu'il soit le plus proche possible d'un filtre passe-bas idéal. Nous avons développé également une analyse théorique pour le cas d'un signal autorégressif d'ordre 2 afin de donner des formes explicites des coefficients de tous les filtres. Les expressions théoriques obtenues peuvent être intéressantes en pratique puisqu'elles permettent de simplifier l'implémentation de la méthode d'optimisation proposée. Concernant les filtres de prédiction, ils sont assez souvent optimisés en minimisant la variance (critère $\ell_2$) des signaux de détails. Dans ce contexte, nous avons développé des critères parcimonieux, en remplaçant le critère usuel $\ell_2$ par des critères $\ell_1$. Tout d'abord, nous avons montré l'intérêt d'utiliser un critère $\ell_1$ au lieu d'un critère $\ell_2$. Ensuite, en remarquant que l'optimisation d'un filtre de prédiction dépend de celle des autres filtres, nous avons proposé d'optimiser les filtres de prédiction en minimisant la somme *pondérée* des normes $\ell_1$ des sous-bandes de détails. Enfin, un algorithme itératif permettant l'optimisation conjointe des différents filtres a été mis en place en alternant entre l'optimisation des filtres et le calcul des poids.

Les résultats obtenus nous incitent à envisager les perspectives suivantes.

D'abord, l'extension de ces nouvelles métodes d'optimisation à d'autres structures de lifting semble être une piste de recherche intéressante. En effet, nous nous sommes intéressés jusqu'à présent aux structures de lifting à un seul étage (c'est à dire une structure 2D P-P-P-U comme le montre la Fig. 6.1). Récemment, il a été montré que la transformation 9/7 retenue par la norme JPEG2000 peut se mettre sous la forme 2D non séparable par une mise en cascade de deux étages de la structure étudiée [Iwahashi, Kiya, 2010]. Pour cela, il serait intéressant d'étendre nos algorithmes d'optimisation à des structures de lifting ayant plus qu'un seul étage de lifting telles que la transformation 9/7 et le schéma VLS-II (P-U-P). De plus, les algorithmes d'optimisation abordés dans cette thèse minimisent des critères liés au débit du signal quantifié. Cependant, il serait plus intéressant de minimiser ces critères sous des contraintes de distorsion.

## Abstract

This thesis addresses the problem of stereo image coding for storage and transmission purposes. In particular, we are interested in lossy-to-lossless coding schemes allowing progressive reconstruction.

In the first part, we propose novel coding methods based on Vector Lifting Schemes. Unlike conventional approaches which encode a reference image, a residual one and a disparity map, the proposed methods do not generate a residual image but two compact multiresolution representations of the left and the right images simultaneously. We also show the benefits that can be drawn from integrating a *smooth* and dense disparity field within such joint stereo image coding schemes.

In the second part, we propose optimization techniques which can be used in the design of these lifting schemes. This allows us to build content adaptive methods. More precisely, we focus on the optimization of *all* the operators (i.e the update and prediction filters) involved in a lifting structure. To this end, we propose and analyze a new criterion for the design of the update filter. Concerning the prediction filters, we investigate techniques for optimizing sparsity criteria.

**Acknowledgements**

First of all, I am most grateful to my three supervisors Prof. Béatrice Pesquet-Popescu, Prof. Amel Benazza-Benyahia and Prof. Jean-Christophe Pesquet for their support, their patience in helping me and their encouragement during my PhD. I feel very lucky to have worked with them.

This thesis would not have been completed without the precious feedback of my thesis committee members Dr. Marc Antonini, Prof. Philippe Salembier and Prof. James E. Fowler. I would like to thank them for their constructive comments on my work.

I would also like to express my thanks to my past and current colleagues at the Signal and Image processing Department of Telecom ParisTech (TSI, Télécom ParisTech), Unité de Recherches en Imagerie Satellitaire et ses Applications (URISA, SUP'COM-Tunis) and Laboratoire d'Informatique de l'Institut Gaspard Monge (LabInfo-IGM) of the Université de Paris-Est Marne-la-Vallée.

Finally, a special thanks goes to my dear wife Emna, for her endless support and encouragement, for understanding and patiently enduring long hours of work I spent on this research. Again, without the support of my parents and my sisters, I would never have been able to overcome the difficulties of my life. I greatly acknowledge them for their tremendous sacrifices and continued support during all the years of my study.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

## 1.1 Thesis context

A main objective of continuous developments in the design of imaging systems has been the addition of more realistic perceptual sensations. Indeed, white and black still images were early captured at the beginning of the nineteenth century. Later, the quality of these images was enhanced by the addition of color. Then, the advent of movies initiated a new era for visualization systems and their application areas. Current developments in image processing tend to add depth information to allow a better immersion in virtual worlds and augmented reality.

In general, humans perceive 3D thanks to the coordination of their eyes and brain. More precisely, since the eyes are separated by a few centimeters, they perceive the same scene from two slightly different angles. These images are then merged by the brain to perceive depth. This is the basic idea behind stereoscopic vision.

The principle of binocular imaging systems consists of generating two views, called left image and right image, by recording two slightly different view angles of the same scene. By presenting the appropriate image of a stereo pair to the left/right eye, the viewer perceives the scene in three dimensions (3D). The recent advances in acquisition and display technologies have allowed the widespread use of stereovision in various application fields. For example, tele-presence in videoconferences [Waldowski, 1991; Feldmann *et al.*, 2010] and tele-medicine [Herpers *et al.*, 1999] are two potential applications of stereovision. Another interesting application field is entertainment. Indeed, adding the third dimension to movies and computer games allows to add more interactivity and is more pleasant to the

viewers/players. Stereoscopic vision has also been used in remote sensing. For instance, today's advances in satellite remote sensing technology provide the capability to collect Stereo Image (SI) pairs for several applications such as cartography and urban planning. Satellite stereo images (such as those generated by IKONOS [1] and SPOT5 [2] sensors) are especially helpful to generate a digital elevation model, which is a 3D representation of the topography of a given area [Tsutsui *et al.*, 2007].

The increasing interest in SIs has led to the constitution of image databases that require huge amounts of storage capacity. For example, the on board stereoscopic instrument of SPOT5 covers areas of 120 km $\times$ 60 km and a single view can be represented by 12000 $\times$ 12000 pixels [Poli *et al.*, 2004]. In addition to SPOT5 and IKONOS sensors, it is worth mentioning the Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER), which is a multispectral sensor with stereo capability [Marangaz *et al.*, 2005]. The acquired SIs are used to generate a high-resolution digital elevation model and the storage of a single stereo pair of any scene requires about 30 Megabits. The Multi-angle Imaging SpectroRadiometer (MISR) also uses nine cameras to generate multiview data sets [G. W. Bothwell, Miller, 2002] at a data rate of 3.3 Mbps. These images are used in the study of Earth climate. Furthermore, combined with stereoscopic techniques, they enable the construction of 3D models and provide very accurate estimations of cloud heights.

As shown by the previous application examples, the involved data amounts are prohibitive and constitute a major limitation for developing stereoscopic applications. To overcome this problem, the use of compression techniques is mandatory for image storage as well as for image transmission.

Note that the use of stereo images for 3D reconstruction purposes, as well as the coding of multicomponents images (in particular multispectral images), have been the subjects of many studies developed in the Unité de Recherches en Imagerie Satellitaire et ses Applications (URISA, SUP'COM-Tunis). On the other hand, compressing multiview images is one of the most important research topics in the Signal and Image processing Department of Telecom ParisTech (TSI, Télécom ParisTech). Thus, the joint supervision of this thesis, devoted to stereo images coding purposes, has been established between SUP'COM-Tunis and Télécom ParisTech. Moreover, an old and intense collaboration between these insti-

[1] http://www.satimagingcorp.com/satellite-sensors/ikonos.html

[2] http://www.satimagingcorp.com/satellite-sensors/spot-5.html

tutes and Laboratoire d'Informatique de l'Institut Gaspard Monge (LabInfo-IGM) of the Université de Paris-Est Marne-la-Vallée in the field of wavelet analysis and optimization has given me the opportunity to develop part of this work with LabInfo-IGM.

## 1.2 Objectives and contributions

The objectives of this thesis are twofold. The first one is to design and analyze new lifting based coding methods for storing/transmitting *stereo* images. The second one is to focus on the *optimization* of the different lifting operators, i.e update and prediction filters, to build *content*-adaptive schemes.
More precisely, we are mainly interested in lossy-to-lossless compression schemes allowing progressive reconstruction of the stereo images. Indeed, on the one hand, progressive reconstruction has become a highly desirable functionality for telebrowsing applications. It consists of encoding the image in several layers: the base layer corresponds to a highly compressed version of the image on which additional refinement layers are built to enhance progressively the image quality. At the decoder side, once the users are satisfied with the quality of the reconstructed image, they can decide to abort the transmission at a particular layer. On the other hand, lossless reconstruction is preferred in some applications, such as medical imaging or remote sensing imaging since an even minor distortion in the reconstructed images could affect the interpretation of the scene.

A detailed list of the main contributions of our research work follows:

- First, we propose a novel joint coding scheme for stereo pairs, based on the concept of Vector Lifting Schemes (VLS). Indeed, unlike conventional approaches which consist of encoding a reference image and a residual one as well as a disparity map, our proposed method employs a joint multiscale decomposition which offers the advantage of generating simultaneously two compact multiresolution representations of the left and the right images. Two versions of this new scheme, VLS-I and VLS-II, are presented and discussed. Furthermore, we conduct a theoretical analysis of the proposed schemes in terms of prediction efficiency.

- While a fixed size block-matching technique is used to estimate the disparity map in the first part, we show in the second one the benefits of integrating a dense dis-

parity estimation method in joint stereo image coding schemes. Unlike conventional methods that employ a block-based disparity estimation technique, our method use a dense and *smooth* disparity map. Since the coding cost of a dense disparity field is too expensive compared to a block-based representation, we propose to resort to an efficient quadtree decomposition and entropy coding that achieves a tradeoff between the accuracy of the disparity representation and the coding cost.

- Next, we propose to adapt the lifting operators to the contents of the input images. In the previously mentioned contributions, the stereo image coding process has been performed by using separable multiscale decompositions. However, due to their separate processing along the image lines and columns, separable lifting schemes may not appear very efficient to cope with the two-dimensional characteristics of edges that are neither horizontal nor vertical. For this reason, we will focus on the optimization of *all* the operators involved in a two-dimensional non separable lifting scheme structure. While the prediction filters are optimized by minimizing the variance of the detail coefficients, one of the main contributions of this part is to design the update filter by using a criterion that allows us to reduce the aliasing effects. Moreover, we show that the proposed optimization method leads to the same optimal update filter when the optimization is performed either before or after the decimation step. Another contribution consists of deriving theoretical expressions of the optimal filter coefficients based on a simple first order 2D autoregressive model of the input image. The use of such a statistical model requiring only two correlation factors highly simplifies the implementation process of our optimization method.

- Finally, we investigate techniques for optimizing sparsity criteria, which can be used for the design of all the filters defined in the previous structure. Indeed, unlike the previous context of work where the prediction filters have been *separately* optimized by minimizing an $\ell_2$ criterion, we focus on the use of an $\ell_1$ criterion instead of an $\ell_2$ one. Since the output of a prediction filter can be used as an input for another prediction filter, we then propose to optimize such a filter by minimizing a *weighted* $\ell_1$ criterion related to the global prediction error. We will see that the optimization of the diagonal prediction filter depends on the optimization of the remaining ones and vice-versa. Related to this fact, we also propose to *jointly* optimize the prediction

filters by using an algorithm that alternates between the optimization of the filters and the computation of the weights.

## 1.3 Thesis outline

This thesis is organized as follows.

In Chapter 2, we first introduce the basic concepts of camera models, and then we establish the geometrical relations between the two views. We also review the most important techniques aiming at estimating the disparity field. This step is a key issue for exploiting the cross-view redundancies and therefore achieving more efficient coding schemes.

In Chapter 3, we give an overview of stereo image coding schemes. To this end, we first remind the most important concepts behind image compression techniques. A particular attention will be paid to lifting based coding methods. We then review previous works on stereo image coding.

In Chapter 4, we propose a novel stereo image coding structure based on a joint multiscale decomposition of the left and the right images. Two versions of the proposed method are provided and discussed. We also conduct a statistical analysis of the proposed schemes in terms of prediction efficiency.

In Chapter 5, we propose to replace the block-based disparity estimation technique by a recent dense disparity estimation method. We first describe the considered dense disparity algorithm. We then address the problem of a suitable representation of the resulting dense disparity map for coding purposes.

In Chapter 6, we first provide a survey of adaptive lifting schemes. We then describe the considered non separable lifting scheme structure which will be optimized. We finally propose a new optimization criterion for the design of the update filter. A theoretical analysis of the proposed method in terms of the used adaptation criterion is also conducted.

As a continuation of Chapter 6, in Chapter 7, we propose sparse optimization criteria that can be used for the design of all the filters involved in the previous lifting structure. We first discuss the motivation of using an $\ell_1$ criterion in lifting optimal design problems. Then, we describe the iterative algorithm used to minimize such a criterion. Next, we propose a *weighted* $\ell_1$ criterion which aims at minimizing the global prediction error. A novel algorithm which *jointly* optimizes the prediction filters is also presented.

Finally, some relevant conclusions and possible extensions of this research are addressed in

Chapter 8.

## 1.4 Publications

**International Journal papers**

• Mounir Kaaniche, Amel Benazza-Benyahia, Béatrice Pesquet-Popescu and Jean-Christophe Pesquet, Non separable lifting scheme with adaptive update step for still and stereo image coding, *accepted for publication in Elsevier Signal Processing: Special issue on Advances in Multirate Filter Bank Structures and Multiscale Representations*, 2011.
• Mounir Kaaniche, Amel Benazza-Benyahia, Béatrice Pesquet-Popescu and Jean-Christophe Pesquet, Vector Lifting Schemes for Stereo Image Coding, *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2463-2475, November 2009.

**International Conference papers**

• Mounir Kaaniche, Jean-Christophe Pesquet, Amel Benazza-Benyahia and Béatrice Pesquet-Popescu, Two-dimensional non separable adaptive lifting scheme for still and stereo image coding, *In IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, pp. 1298-1301, Dallas, USA, 14-19 March 2010.
• Mounir Kaaniche, Wided Miled, Béatrice Pesquet-Popescu, Amel Benazza-Benyahia and Jean-Christophe Pesquet, Dense disparity map representations for stereo image coding, *In IEEE International Conference on Image Processing, ICIP*, Cairo, Egypt, November 2009.
• Ismael Daribo, Mounir Kaaniche, Wided Miled, Marco Cagnazzo and Béatrice Pesquet-Popescu, Dense disparity estimation in multiview video coding, *In International Workshop on Multimedia Signal Processing, MMSP*, Rio de Janeiro, Brazil, October 2009.
• Mounir Kaaniche, Amel Benazza-Benyahia, Jean-Christophe Pesquet and Béatrice Pesquet-Popescu, Lifting schemes for joint coding of stereoscopic pairs of satellite images, *In the European Signal and Image Processing Conference, EUSIPCO*, pp. 981-944, Poznan, Poland, 3-7 September 2007.

**Seminars**

• Mounir Kaaniche, Amel Benazza-Benyahia, Béatrice Pesquet-Popescu and Jean-Christophe Pesquet, Vector lifting schemes for stereo image coding, *Séminaire Compression d'images Multi-composantes*, Journée CNES des jeunes chercheurs, 8$^{e}$ édition, Institut Aéronautique

et Spatial, Toulouse, France, October 2008.

**Award**

MMSP 2009: *Top* 10% *Paper Award* was granted for the paper entitled "Dense disparity estimation in multiview video coding".

# Chapter 2

# Main concepts in Stereo Vision

## 2.1 Introduction

This chapter outlines the main aspects relevant for understanding the stereoscopic imaging process of a 3D scene through a camera system. We will first introduce the basic concepts of camera models, and then we will establish the geometrical relation between the two views. Next, we will present the stereo matching problem as well as the most commonly used constraints that allow to simplify this process. Finally, an overview of the most frequently used disparity estimation methods is provided.

## 2.2 Acquisition

Before discussing the stereo imaging acquisition process, we first introduce the *pinhole camera model* which is considered as the most simplest camera model. Indeed, experiences have shown that such a simple model can accurately approximate the geometry and optics of most modern cameras [Faugeras, 1994]. Thus, it defines the geometric relationship between a 3D object point and its 2D corresponding projection onto the image plane. This geometric mapping from 3D to 2D is called a perspective projection.

### 2.2.1 Pinhole camera model

Acquisition of an image using the pinhole camera model is illustrated in Fig. 2.1.
This geometric representation consists of a retinal plane, a focal one and the image plane. The point $C$ is called the *optical center* or *center of projection*. The distance between the

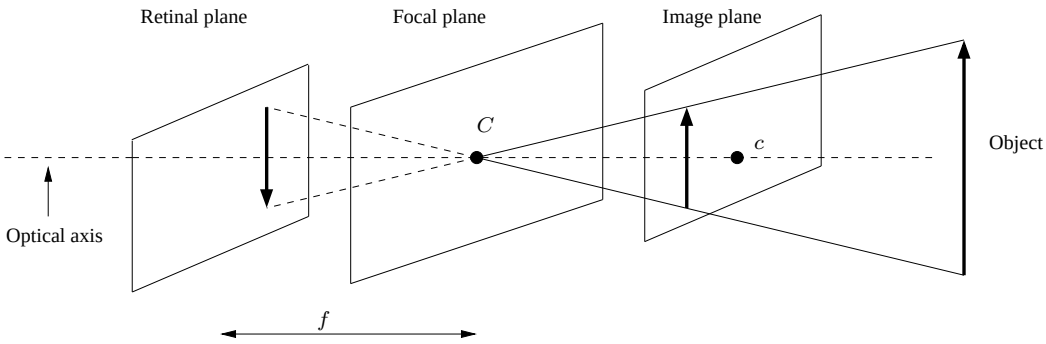



Figure 2.1: Pinhole camera model.

focal plane and the retinal one is called the *focal length* of the optical system and denoted by $f$. The line going through the optical center and perpendicular to the retinal plane is called the *optical axis*, and it intersects the image plane at a point $c$ called the *principal point*.

We should note that, for each object in the 3D world, the considered camera model forms an inverted image of that object on the retinal plane located behind the focal plane. However, it appears preferable to place the image plane at $f$ distance in front of the focal plane in order to avoid a non-inverted image of the object.

Now, let us explore the equations for the perspective projection describing the relation between a point $M$ in the 3D world and its projection $m$ onto the image plane (Fig. 2.2). This relationship is composed of the three following transformations:

- A perspective projection which defines the relation between the 3D point $M$ expressed in the camera coordinate system and its projection $m$ expressed in the same coordinate system.
- A transformation from camera to image that consists in converting the coordinates of $m$ from the camera coordinate system to a new coordinate system attached to the image.
- A transformation from world to camera which expresses the relation between the 3D world coordinate system and the 3D camera coordinate system.

### Perspective projection using homogeneous coordinates

Let $\mathcal{R}_c$ be the camera coordinate system originated at the optical center $C$ and whose $Z$-axis coincides with the optical axis of the camera. For the sake of simplicity, the horizontal

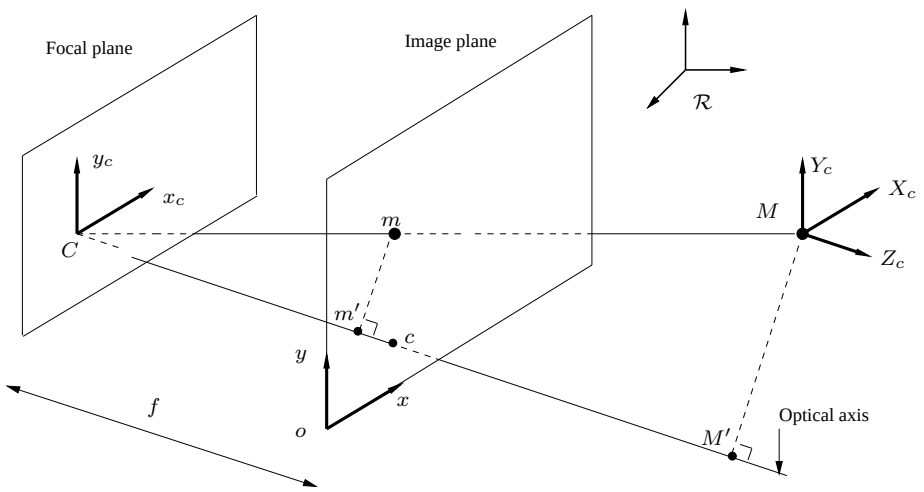



Figure 2.2: Perspective projection.

and vertical axes are assumed to be parallel to those of the image plane.

Consider the point $M$ in the 3D world with coordinates $(X_c, Y_c, Z_c)$ in $\mathcal{R}_c$ and its projection $m$ with coordinates $(x_c, y_c)$ in $\mathcal{R}_c$. Let $m'$ and $M'$ be the projections of $m$ and $M$ onto the optical axis, respectively. Using similarities of triangles $(C, m, m')$ and $(C, M, M')$, we can deduce that:

$$\begin{aligned} x_c &= f\frac{X_c}{Z_c} \\ y_c &= f\frac{Y_c}{Z_c}. \end{aligned} \tag{2.1}$$

It is important to note that these equations are not linear with respect to the Euclidean coordinates $X_c$, $Y_c$ and $Z_c$. For this reason, homogeneous coordinates will be introduced in order to obtain linear equations. These coordinates are defined as follows: if $(X, Y, Z)$ are the *Euclidean coordinates* of a point in $\mathbb{R}^3$, its *homogeneous coordinates* are defined in the projective space $\mathbb{P}^3 \subset \mathbb{R}^4\backslash\{0\}$ by $(\lambda X, \lambda Y, \lambda Z, \lambda)$ for each nonzero real value of $\lambda$.

More precisely, the projective space $\mathbb{P}^3$ is a subset of $\mathbb{R}^4\backslash\{0\}$ defined by the relation of equivalence:

$$(X, Y, Z, W) \simeq (X', Y', Z', W') \quad \Longleftrightarrow \quad \exists\, \lambda \neq 0 \;\mid\; (X, Y, Z, W) = \lambda \cdot (X', Y', Z', W') \tag{2.2}$$

Let us now denote by $\widetilde{\mathbf{M}} = (X, Y, Z, 1) \simeq (\lambda X, \lambda Y, \lambda Z, \lambda) \in \mathbb{P}^3$, where $\lambda \neq 0$, the homogeneous coordinates of the point $M$ whose Euclidean coordinates are $(X, Y, Z)$ in $\mathbb{R}^3$. In a similar way, we denote by $\widetilde{\mathbf{m}} = (x, y, 1) \simeq (\lambda x, \lambda y, \lambda) \in \mathbb{P}^2$, where $\lambda \neq 0$, the homogeneous

coordinates of the point $m$ whose Euclidean coordinates are $(x, y)$ in the image plane. Using the homogeneous representation of the points, Eq. (2.1) can be rewritten as follows:

$$\begin{pmatrix} s.x_c \\ s.y_c \\ s \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \tag{2.3}$$

where $s$ is a nonzero scale factor.

**Transformation from camera to image**

Most of the the current imaging system define the origin $O$ of the image coordinate system at the top left pixel of the image. This coordinate system will be denoted by $\mathcal{R}_o$. However, the previous image coordinates $(x_c, y_c)$ are expressed in $\mathcal{R}_c$. Thus, a transformation between these two coordinate systems is required. If we denote by $(x, y)$ the corresponding pixel in the digitized image (i.e expressed in $\mathcal{R}_o$), this position is related to the camera coordinate system by the following transformation:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} k_x & 0 & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_c \\ y_c \\ 1 \end{pmatrix} \tag{2.4}$$

where

- $k_x$ and $k_y$ are the horizontal and vertical scale factors (pixels/mm).
- $x_0$ and $y_0$ are the pixel coordinates of the principal point $c$ in $\mathcal{R}_o$.

These parameters are called the camera intrinsic parameters since they do not depend on the position and orientation of the camera [Faugeras, 1994; Pedersini *et al.*, 1999]. Therefore, the perspective projection equation (Eq. (2.3)) becomes:

$$\begin{pmatrix} s.x \\ s.y \\ s \end{pmatrix} = \begin{pmatrix} k_x & 0 & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} \tag{2.5}$$

By defining $\alpha_x = k_x f$ and $\alpha_y = k_y f$, this equation is often rewritten as:

$$\begin{pmatrix} s.x \\ s.y \\ s \end{pmatrix} = \begin{pmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}. \tag{2.6}$$

**Transformation from world to camera**

Let $\mathcal{R}$ be the 3D world coordinate system and $(X, Y, Z)$ the coordinates of the point $M$ in $\mathcal{R}$. These coordinates are related to those expressed in $\mathcal{R}_c$ by a rotation $\mathbf{R}$ followed by a translation $\mathbf{t}$. Thus, we obtain:

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \mathbf{t} \tag{2.7}$$

Using the homogeneous coordinates, Eq. (2.7) can also be expressed as:

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.8}$$

The matrix $\mathbf{R}$ of size $3 \times 3$ and the vector $\mathbf{t}$ of size $3 \times 1$ describe the orientation and position of the camera with respect to the world coordinate system. They are called the *extrinsic* parameters of the camera [Faugeras, 1994; Pedersini *et al.*, 1999].

Finally, combining the three transformations described above yields the general form of the perspective projection matrix $\mathbf{P}$ of the camera:

$$\mathbf{P} = \mathbf{A}\mathbf{P_0}\mathbf{K} \tag{2.9}$$

where

$$\mathbf{A} = \begin{pmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{P_0} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{K} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \tag{2.10}$$

This expression can also be rewritten as follows:

$$\mathbf{P} = \mathbf{A}[\mathbf{R} \quad \mathbf{t}] \tag{2.11}$$

Consequently, the relationship between the homogeneous coordinates of a world point $\widetilde{\mathbf{M}}$ and its projection $\widetilde{\mathbf{m}}$ onto the image plane is described by:

$$\widetilde{\mathbf{m}} = \mathbf{P}\widetilde{\mathbf{M}} \tag{2.12}$$

### 2.2.2 Stereoscopic imaging system

The principle of a binocular stereoscopic imaging systems consists in generating two views, called left image and right image, by recording two slightly different view angles of the same scene (Fig. 2.3). The left and right cameras are respectively represented by their optical centers $C^{(l)}$ and $C^{(r)}$, and their perspective projection matrix $\mathbf{P}^{(l)}$ and $\mathbf{P}^{(r)}$.

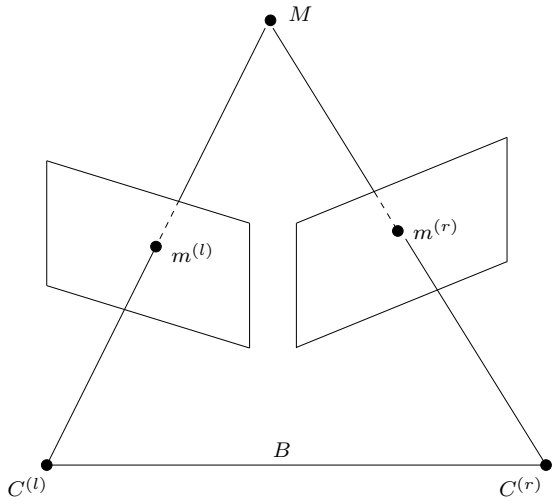



Figure 2.3: Stereoscopic imaging system.

Assuming that the object point $M$ is projected onto two points $m^{(l)}$ and $m^{(r)}$ in the left and right images creating the so-called *homologous points*. Using Eq. (2.12), this projection can be expressed as:

$$\begin{aligned} \widetilde{\mathbf{m}}^{(\mathrm{l})} &= \mathbf{P}^{(l)}\widetilde{\mathbf{M}} \\ \widetilde{\mathbf{m}}^{(\mathrm{r})} &= \mathbf{P}^{(r)}\widetilde{\mathbf{M}}. \end{aligned} \tag{2.13}$$

### 2.2.3 Epipolar geometry

The epipolar geometry describes the relation between the two generated images. The different entities involved in this geometry are illustrated in Fig. 2.4. The plane passing through the object point $M$ and the centers of projection $C^{(l)}$ and $C^{(r)}$ is called the *epipolar*

*plane* Π. The intersection of the line joining the optical centers (called *baseline*) with the image plane results in two points $e^{(l)}$ and $e^{(r)}$ called the *epipoles*. The intersection of the left (resp. right) image plane with the line joining the point $M$ and the left (resp. right) optical center $C^{(l)}$ (resp. $C^{(r)}$) corresponds to $m^{(l)}$ (resp. $m^{(r)}$).

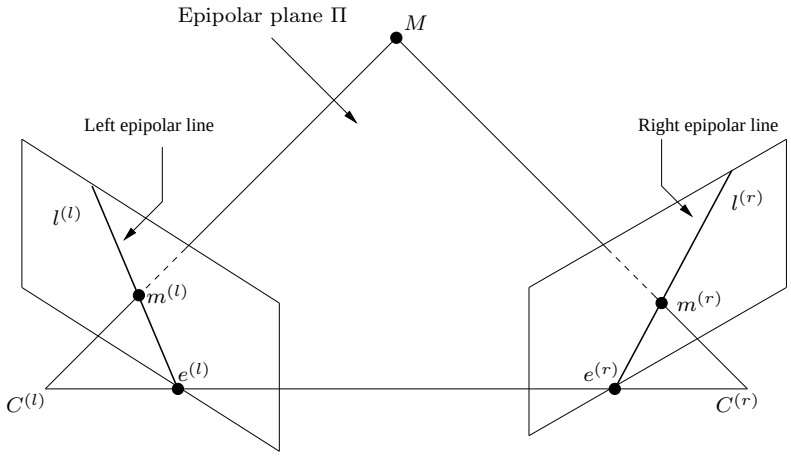



Figure 2.4: Epipolar geometry between a pair of images.

Suppose now that we only know $m^{(r)}$ and we have to search for its corresponding point $m^{(l)}$. The plane Π is determined by the baseline and the ray defined by $m^{(r)}$. From above, we know that the ray passing through $m^{(l)}$ lies in Π. Hence, the point $m^{(l)}$ lies on the line of intersection $l^{(l)}$ of Π with the left image plane. This line is called the *epipolar line* $l^{(r)}$ associated to $m^{(r)}$.

The knowledge of the epipolar geometry leads to an important observation. Given an image point $m^{(r)}$ lying on the right epipolar line $l^{(r)}$, its corresponding point $m^{(l)}$ must lie on the conjugate epipolar line $l^{(l)}$. This corresponds to the epipolar constraint which can be expressed mathematically as:

$$(\widetilde{\mathbf{m}}^{(\mathrm{l})})^{\top}\mathbf{F}\widetilde{\mathbf{m}}^{(\mathrm{r})} = 0 \tag{2.14}$$

where $\mathbf{F}$ is the *Fundamental matrix*. This matrix, of size $3 \times 3$, depends on the intrinsic and extrinsic parameters of the cameras [Zhang, Xu, 1997]. An overview of techniques to find $\mathbf{F}$ can be found in [Zhang, 1998].

Thus, such a constraint allows to reduce the search of homologous points to a 1-D search problem rather than a 2-D one. In practice, this problem can be further simplified by using an appropriate processing known as *epipolar rectification*.

### 2.2.4 Epipolar rectification

There are two primary camera configurations used in stereoscopic vision. The first configuration, which has been considered at the beginning of this section, is called *converging camera configuration*. In this configuration, cameras are rotated towards each other by a small angle and thus the epipolar lines appear inclined. The second configuration is called *parallel camera configuration* (Fig. 2.5), and it is composed of two cameras with parallel optical axes.

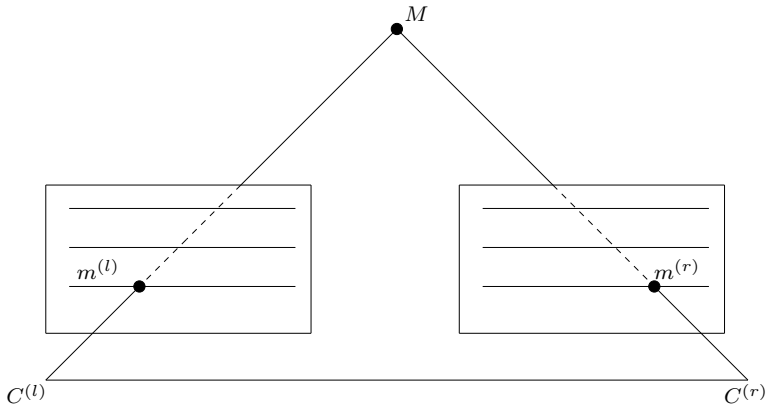



Figure 2.5: Epipolar rectification.

It is important to note that the parallel camera configuration has several advantages. Indeed, it simplifies the search for homologous points, which will be situated on parallel horizontal lines. Furthermore, in this particular case, the epipolar lines are horizontal and coincide with the image scan lines. Hence, two corresponding pixels are located at the same line in the two images as shown in Fig. 2.5. In the opposite case (i.e converging camera configuration), the epipolar lines can be rendered horizontal by using an appropriate processing. This procedure is called rectification. A survey of different rectification methods proposed in the literature can be found in [Hartley, Gupta, 1993; Fusiello *et al.*, 2000; Papadimitriou, Dennis, 1996]

## 2.3 Stereo matching process

### 2.3.1 Disparity information

Given an image point $m^{(r)} = (m_x^{(r)}, m_y^{(r)})$ in the right image, the stereo matching process aims at searching its homologous point $m^{(l)} = (m_x^{(l)}, m_y^{(l)})$ in the left image. The displace-

ment between the coordinates of these points is called *disparity*.

So, the stereo matching problem can be formulated mathematically by the search for a function $v$ that associates a disparity vector to each pixel $(m_x^{(r)}, m_y^{(r)})$ in the right image:

$$\begin{aligned} v :&\mathbb{R}^2 \longrightarrow \mathbb{R}^2 \\ &(m_x^{(r)}, m_y^{(r)}) \longmapsto v(m_x^{(r)}, m_y^{(r)}) = (m_x^{(l)} - m_x^{(r)}, m_y^{(l)} - m_y^{(r)}). \end{aligned} \tag{2.15}$$

When the left and right images are rectified, the displacement between the homologous points is purely horizontal. In this case, the above equation becomes:

$$\begin{aligned} v_x :&\mathbb{R}^2 \longrightarrow \mathbb{R} \\ &(m_x^{(r)}, m_y^{(r)}) \longmapsto v_x(m_x^{(r)}, m_y^{(r)}) = m_x^{(l)} - m_x^{(r)}. \end{aligned} \tag{2.16}$$

### 2.3.2 Depth information

An important application of stereoscopic display technologies consists in adding the *depth information* of the scene to the viewer. This kind of information can be computed based on the triangulation method (see Fig. 2.6) [Hartley, P. Sturn, 1997].

Recall that a 3D object point $M$ is defined by its coordinates $(X, Y, Z)$ where the $Z$ component represents its depth and corresponds to the distance between the object and the cameras. Using the similarity of the two triangles whose hypotenuses are respectively

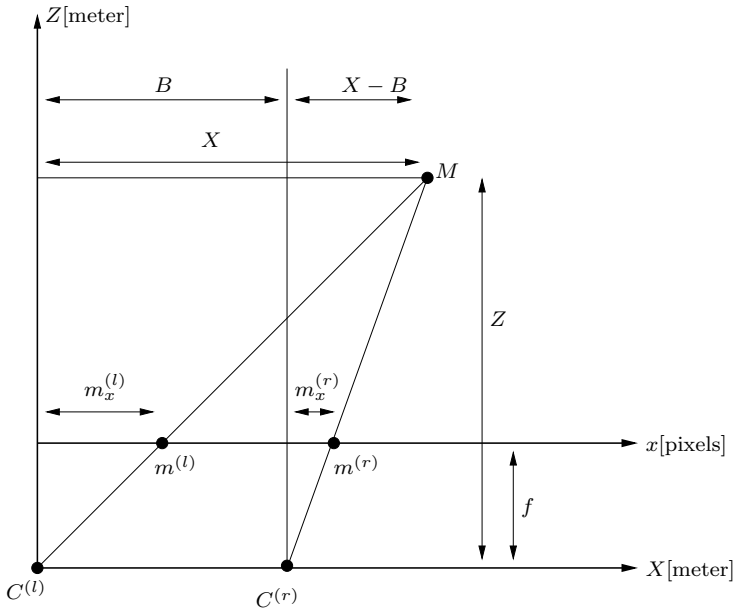



Figure 2.6: Relation between the disparity and depth information.

$(C^{(l)}m^{(l)})$ and $(C^{(l)}M)$, we obtain:

$$\frac{m_x^{(l)}}{f} = \frac{X}{Z}. \tag{2.17}$$

Similarly, we can also check that:

$$\frac{m_x^{(r)}}{f} = \frac{X - B}{Z} \tag{2.18}$$

where $B$ is the baseline and $f$ denotes the camera focal length.

Finally, using the definition of the disparity information given by Eq. (2.16), we deduce that:

$$v_x(m_x^{(r)}, m_y^{(r)}) = m_x^{(l)} - m_x^{(r)} = \frac{B \cdot f}{Z} \tag{2.19}$$

Thus, this equation allows us to calculate the depth information of the real scene. Fig. 2.7 shows the true disparity and depth maps computed from the "room3D" stereo pair based on the camera parameters. Indeed, it can be observed that the depth information is inversely proportional to the disparity.

### 2.3.3 Difficulties in stereo matching process

It is important to note that the stereo matching problem has been considered as a difficult problem in stereo vision for several reasons like depth discontinuities, illumination variations and lack of texture [Dhond, Aggarwal, 1989].

- Occlusion problem [Dhond, Aggarwal, 1989]: In general, stereo images contain nearly similar contents since they correspond to the same scene. However, there are some areas in one image that are absent in the other image, and they are referred to as *occluded areas*. The occlusion effect is illustrated in Fig. 2.8 where the point $m^{(r)}$ is only visible in the right image.

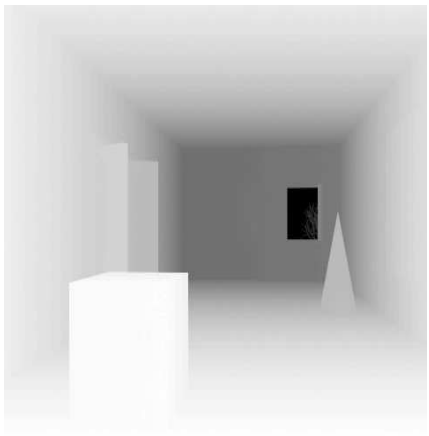
  This occlusion effect in stereo images is due to the different viewpoints of the cameras and the presence of discontinuities in the scene. Therefore, the disparity is undefined in occlusion areas because such areas cannot be found in the other image.

- Illumination variations: In real stereoscopic imaging system, the characteristics of the cameras may be slightly different. Consequently, some illumination changes may appear between the captured images. Fig. 2.9 shows an example of a SPOT5 stereo

(a): Left image

(b): Right image

(c): Disparity map



(b): Depth map

Figure 2.7: Disparity and depth maps computed from the "room3D" stereo image based on the camera parameters.

image which has significant illumination variations. This luminance difference is confirmed by displaying the histograms of both images.

This illumination variation will cause a serious problem in the correspondence process. In order to overcome this problem, a pre-processing step, like the histogram equalization, is often applied to the original stereo images.

- Textureless regions: Obviously, a pixel is less discriminating in areas with repetitive


(a): Left image (b): Right image

Figure 2.8: Occlusion effects: the point $m^{(r)}$ is visible in the right image and absent in the left image.

structure or texture. Indeed, it will be difficult to distinguish pixels in the same area having similar intensity. Thus, such homogeneously textured regions result in ambiguities in the stereo matching process, simply because of the presence of multiple possible matches.

In conclusion, stereo matching is an ill-posed problem with inherent ambiguities.

### 2.3.4 Stereo matching constraints

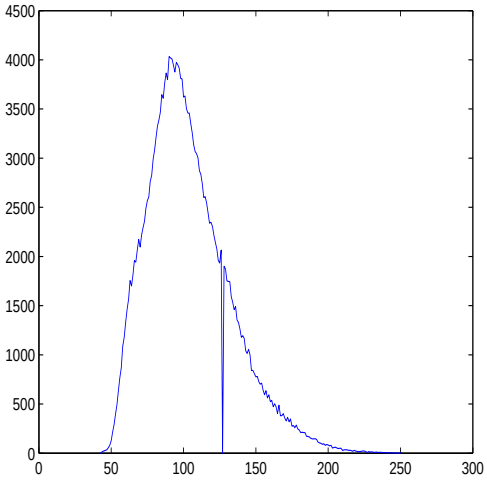
In order to overcome the ambiguities mentioned above, some matching constraints can be imposed. The most commonly used constraints are the following ones [Miled *et al.*, 2006a; Boufama, Jin, 2002]:

- Epipolar constraint: Given an image point in one image, the corresponding point must lie on an epipolar line in the other image. The main advantage of this constraint is that it reduces the matching problem from a 2D search problem to a 1D one. Furthermore, when the stereo images are rectified, the 1D search problem is further simplified since the epipolar line will coincide with the image scan line.

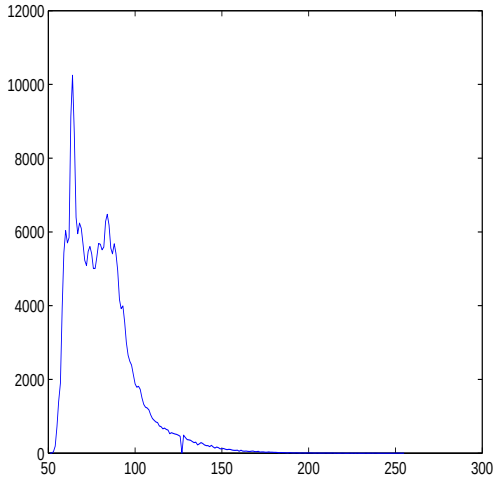
- Uniqueness constraint: It imposes that a given pixel in one image can match to no more than one pixel in the other image. However, this constraint fails in the presence of transparent objects.

(a): Left image

(b): Right image




(c): Histogram of the left image




(b): Histogram of the right image

Figure 2.9: Illumination variations between the left and the right images of the SPOT5 pair.

- Ordering constraint: It ensures that the order of two pixels in a line of one image is preserved in its homologous line of the other image. However, this constraint fails at the *forbidden zone* which is defined by the rays $(C^{(r)}M)$ and $(C^{(l)}M)$.

- Smoothing constraint: It imposes that disparity varies smoothly across homogeneous areas (inside objects).

## 2.4 Overview of stereo matching approaches

With an ultimate aim of producing a disparity map that can be used in the different applications mentioned earlier in Chapter 1, stereo matching process has been extensively studied in computer vision. A survey of the state-of-the-art methods can be found in [Scharstein, Szeliski, 2002].

Traditionally, stereo matching algorithms are basically classified into two categories: local methods and global ones.

### 2.4.1 Local methods

These methods can be mainly grouped into two categories: feature- and area-based approaches. Algorithms in the first category aim at pairing high-level features such as edges [Otha, Kanade, 1985], segments [Medioni, Nevatia, 1985], etc. They yield accurate results but their main drawback is the sparsity of the disparity map. Furthermore, they require a prior feature extraction step.

In contrast with feature-based methods, area-based ones have the advantage of directly generating a dense disparity field, but they are more sensitive to locally ambiguous regions like occlusion regions or regions with uniform texture. This class of methods, where pixels or regions are matched to find the corresponding points, is known as correlation-based methods.

Although many local methods have been developed, we only present the correlation-based methods, which summarize the techniques used by most algorithms. For instance, it is worth pointing out that this technique has been widely employed in many applications such as motion estimation [Alkanhal *et al.*, 1999], image registration [Zitova, Flusser, 2003], etc. The principle of the correlation-based method is illustrated in Fig. 2.10. Given a pixel $m^{(r)}$ in the right image, the problem is to find the corresponding pixel in the left image. For this purpose, a window of pixels around $m^{(r)}$ is firstly selected. Then, this window is matched to a window in the left image by shifting the current window over a search area $S$. At each shift, a cost function between the two windows is computed. To this respect, many different similarity measures have been used in the literature. The most commonly used are the Sum of Square Differences (SSD), the Sum of Absolute Differences (SAD) and the Normalized Cross-Correlation (NCC). Finally, the shift which gives the minimum cost function is considered as the best match.

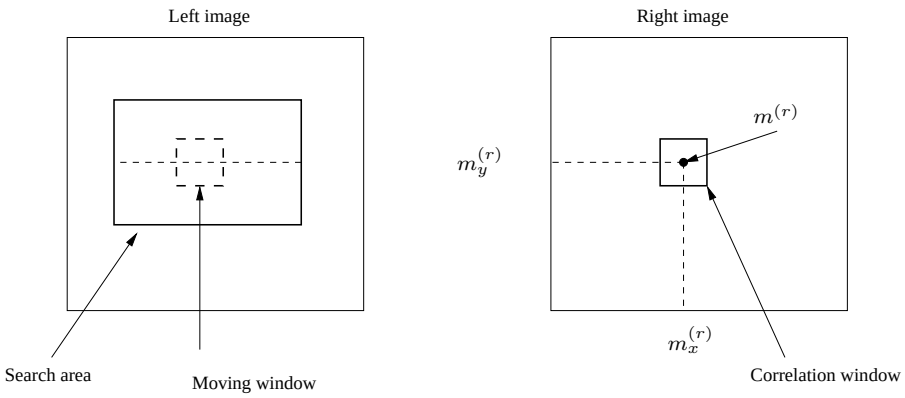



Figure 2.10: Correlation-based stereo matching method

A major difficulty of the correlation-based method is the selection of the window size and shape. Indeed, the basic method uses a rectangular window with a fixed size. However, the limitation of this technique is that it fails at object edges and boundaries. To overcome this problem, more sophisticated approaches have been proposed by varying the size and shape of the window according to the intensity variation [Kanade, Okutomi, 1994; Woo, Ortega, 2000; Veksler, 2001; Xing *et al.*, 2009].

### 2.4.2 Global methods

The second category that aims at reducing sensitivity to local regions in the image are global methods. They typically find the disparity map that minimizes a global energy function over the entire image. Generally, such a function consists of two terms and takes the following form:

$$E(v) = E_1(v) + \alpha E_2(v) \tag{2.20}$$

where $E_1(v)$ is a data term that measures the distance between corresponding pixels, $E_2(v)$ is a regularization term that enforces the smoothness of the disparity field and $\alpha$ is a positive constant weighting the two terms.

Finding the disparity map is achieved by minimizing the global cost function. For this purpose, several energy minimization algorithms have been proposed to solve Eq. (2.20). The most common approaches are dynamic programming [Veksler, 2005; Kim *et al.*, 2005; Sun, 2002], graph cuts [Boykov *et al.*, 2001; Kolmogorov, Zabih, 2001] and variational methods [Slesareva *et al.*, 2005; Miled *et al.*, 2006b; Miled *et al.*, 2009b; Kosov *et al.*, 2009].

Dynamic programming (DP) technique uses the ordering and smoothness constraints to

optimize correspondences in each scan line. It consists of two steps: a forward step for building the path cost matrix for each possible candidate pixel and a backward step, where the matching pixel with the minimum path cost is selected. While this method has the advantage of enforcing the smoothness constraint of the disparity along the epipolar line, it suffers from the inconsistency problem between scanlines. To solve this problem, a two-pass dynamic programming technique has been developed in [Kim *et al.*, 2005].

Graph cut based algorithms have also been proposed to cope with this problem. The idea is to cast the stereo matching problem as a pixel labelling problem to find the minimal cut through a constructed graph.

Much attention was also paid to variational approaches to compute a consistent disparity map while preserving the depth discontinuities. Indeed, these techniques have already been attracted much interest in the computer vision community where they were first used for the purpose of estimating a dense optical flow from a sequence of images. Variational methods minimize the energy function (eq. (2.20)) by solving the associated non-linear Euler-Lagrange equation [Alvarez *et al.*, 2002].

While Dynamic programming and graph cuts methods have been performed in a discrete manner, variational techniques work in a continuous space. Therefore, they have the advantage of producing a disparity field with ideally infinite precision. However, it should be noted that all these global methods are often implemented using numerical schemes which may be computationally intensive. For example, processing a single stereo image pairs can take a few minutes on todays CPUs. In order to become well suited for real-time applications, accurate real-time disparity estimation with variational methods has been recently proposed [Kosov *et al.*, 2009]. Also, more efficient implementation on multicore architecture can be achieved by exploiting the structure of the used algorithm where different instructions may be computed in parallel as it has been addressed in [Miled *et al.*, 2009a]. Moreover, the choice of the regularization weight $\alpha$ may be a difficult task.

Finally, it is worth noting that among these global approaches, it has been shown in [Miled, 2008; Miled *et al.*, 2009a] that variational-based disparity estimation methods are among the most powerful techniques meanwhile preserving the depth discontinuities.

## 2.5 Conclusion

In this chapter, we have reviewed the main issues on stereoscopic imaging system. More precisely, we have focused on the acquisition process of stereo image while emphasizing the geometrical relations between the generated views. We have also reminded the most important techniques aiming at estimating the disparity map. This step plays a crucial role in many applications like 3D reconstruction, view synthesis and stereo image coding. The latter example of applications will be the objective of the next chapter.

# Chapter 3

# Stereo image coding: state-of-the-art

## 3.1 Introduction

Due to the widespread usage of stereo image in many application fields like those addressed in Section 1.1, the demand for storing or transmitting these images over channels has increased. Thus, compressing them would improve the performance of such channels or storage devices.

This chapter gives a survey on conventional stereo images coding schemes. In the first part, we start by presenting the most important concepts behind image compression techniques which will be useful for the remainder of this dissertation. In the second part, we provide an overview of previous works on stereo image coding.

## 3.2 Compression tools

Generally, compression techniques aim at reducing the number of bits needed to represent an image. Several methods of data reduction are available, the choice of which strongly depends on the underlying application requirement [Bovik, 2000]. Traditionally, compression techniques are basically classified into two categories: lossy techniques and lossless ones.

A typical coding scheme is shown in Fig. 3.1. It incorporates three fundamentals steps namely *transformation/modelling*, *quantization* and *entropy coding*.

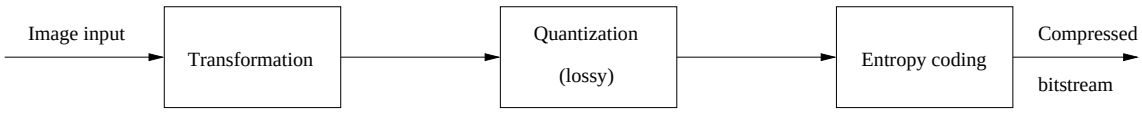



Figure 3.1: Generic compression system.

### 3.2.1 Transformation

In most still images, the neighboring pixels are strongly correlated and therefore contain redundant information. Thus, reducing the amount of data can be efficiently achieved by exploiting the spatial redundancies in the images. For this purpose, two basic compression techniques can be envisaged: predictive coding schemes and transform coding schemes.

#### Predictive coding techniques

One of the simplest way to compress an image is the predictive coding approach which has been adopted in the lossless JPEG standard [Wallace, 1991]. The most common approach to predictive coding is *Differential Pulse Code Modulation* (DPCM) [Cutler, 1950].

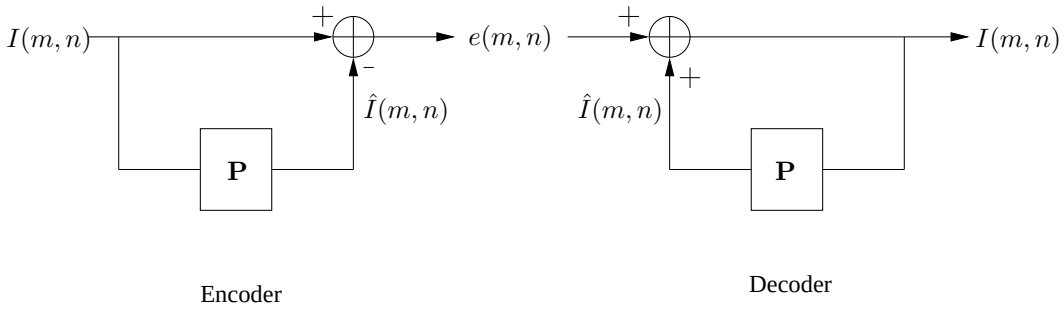



Figure 3.2: DPCM technique in a lossless coding context.

The basic idea of this technique, which is illustrated in Fig. 3.2, consists of two steps. First, the current pixel to be encoded is predicted from its neighboring based only on the previous pixels (in the adopted scanning order):

$$\hat{I}(m,n) = \mathbf{P}\Big(I(m-k,n-l)\Big)_{(k,l)\in\mathcal{P}} \tag{3.1}$$

where $\mathcal{P}$ is the support of the predictor $\mathbf{P}$ of $I(m,n)$.

Then, instead of coding the current pixel $I(m,n)$, the difference between this pixel and its prediction, called *prediction error*, is encoded:

$$e(m,n) = I(m,n) - \hat{I}(m,n). \tag{3.2}$$

It is worth pointing out that a rounding operator should generally be applied on the predicted value if a lossless technique is required.
Thus, if the pixel values are very similar, the error to be encoded has a small magnitude and the number of bits per pixel needed to transmit is thus reduced. Generally, such a method provides a single-resolution version of the image and therefore is not appropriate for progressive reconstruction. Furthermore, it is very sensitive to transmission error [Marvell *et al.*, 1999]. To increase the robustness against noise and improve the compression performance, transform coding schemes have been proposed.

#### Transform coding techniques

An alternative approach for image compression is transform coding where the image is converted from the spatial domain to another domain by applying an invertible transform. An appropriate choice of transform should guarantee a good decorrelation of the coefficients while simultaneously producing a new representation in which most of the original image energy is concentrated on a few coefficients.
For this purpose, many transforms have been proposed, among them the most popular being the Karhunen-Loeve Transform (KLT) [Hua, Liu, 1998], the Discrete Cosine Transform (DCT) [Rao, Yip, 1990] and the Discrete Wavelet Transform (DWT) [Mallat, 1998].

#### Block transforms: KLT and DCT

Generally speaking, a block-based transform consists of first partitioning the image into nonoverlapping adjacent blocks of size $N \times N$ (usually, $N = 8$ or $N = 16$). Then, a transform matrix $\mathbf{T}$ is applied to each block $\mathbf{X}$ yielding an output block of transformed coefficients $\mathbf{Y}$. Well-known examples of block-based transforms are the KLT and the DCT. The KLT is the best transform from a theoretical viewpoint in the sense that it achieves optimal decorrelation and energy compaction. However, it has two major problems. First, KLT is data dependent since its transformation matrix depends on the autocovariance matrix of the input image. Second, it is computationally intensive. Indeed, its complexity is given by $O(N^2)$.
Due to these problems, various approximations to the KLT, that have fast implementation algorithms, have been proposed. The most popular one is DCT which computes the output

block $\mathbf{Y}$ as follows:

$$\mathbf{Y} = \mathbf{T}\mathbf{X}\mathbf{T}^{\top} \tag{3.3}$$

where the generic element $T(m,n)$, for $(m,n) \in \{0,\dots,N-1\}^2$, of the transform matrix $\mathbf{T}$ is defined as:

$$T(m,n) = c_m \cos\left(\frac{(2n+1)m\pi}{2N}\right) \quad \text{with} \quad c_m = \begin{cases} \sqrt{\frac{1}{N}} & \text{if} \quad m = 0 \\ \sqrt{\frac{2}{N}} & \text{if} \quad m > 0. \end{cases} \tag{3.4}$$

This transform has been widely used in image and video compression standards, such as the JPEG standard, the H.264/AVC standard and the different MPEG standards [Pennebaker, Mitchell, 1993]. However, DCT-based block coding also has some shortcomings. Its main drawback is the fact that it causes blocking artifacts at very low bitrate coding (below 0.25 bpp).

To solve this problem, Lapped Orthogonal Transforms (LOT) have been developed [Malvar, 1992]. Rather than partitioning the input image into nonoverlapping blocks, LOT smooth out the block boundaries by using overlapping blocks. Besides, an important feature of such transforms is the existence of filter bank implementations.

### Discrete Wavelet Transform

In order to overcome the aforementioned drawbacks of DCT, wavelets have been attracting much attention in the most recent image compression algorithms, including the JPEG2000 standard [Taubman, Marcellin, 2001]. Wavelets have also been extensively used in various signal processing applications due to their intrinsic properties: multi-scale representation and good space-frequency localization. In fact, unlike the DCT where the image is transformed from the spatial domain into the frequency domain, the wavelet transform provides a good representation of the image in both space and scale domains.

In practice, the information that is produced and analyzed is generally discrete. This is why we will focus on the Discrete Wavelet Transform (DWT) in what follows. More precisely, we will address the wavelet theory in the case of a one dimensional discrete signal in $\ell^2(\mathbb{Z})$ (1D-DWT). Then, we will show that a separable two dimensional WT (2D-DWT) can be easily deduced from the 1D-DWT to transform images.

Mallat has shown that continuous WT can be approximated by a DWT using a discrete filter bank implementation [Mallat, 1998]. The link between the continuous and the discrete wavelet transform has also been studied by Rioul [Rioul, 1993]. Let $(s_0(n))_{n\in\mathbb{Z}}$ be

the original discrete signal in $\ell^2(\mathbb{Z})$ to be decomposed. Generally, $M$-band filter banks are structures composed of $M$ digital filters that allow a signal to be decomposed into $M$ frequency subbands [Vaidyanathan, 1993; Pau, Pesquet-Popescu, 2005; Chaux *et al.*, 2005]. Fig. 3.3 shows an example of a two-band filter bank.

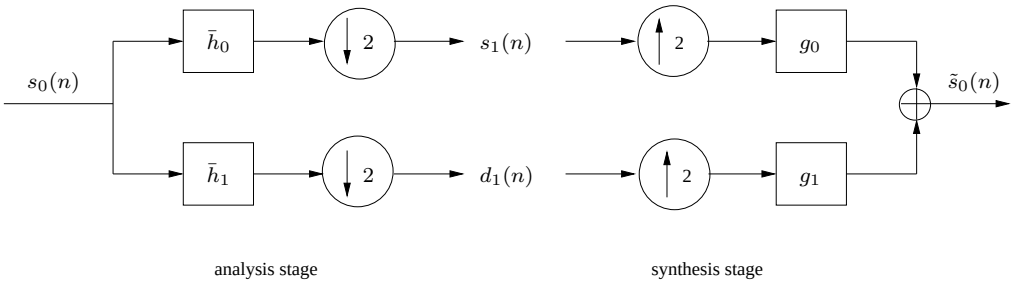



Figure 3.3: 2-band filter bank: analysis and synthesis.

As we can see in Fig. 3.3, the elements of $s_0$ are first fed into two filters, whose impulse responses are denoted by $\bar{h}_0(n) = h_0(-n)$ and $\bar{h}_1(n) = h_1(-n)$, called *analysis filters*. The filter with impulse response $h_0$ (resp. $h_1$) is a low-pass (resp. high-pass) filter. Then, the output of each filter is downsampled by 2 to get a low-pass version $s_1$ (or a coarse resolution version) and respectively a high-pass version $d_1$ of the input signal $s_0$. The low-pass version is called *approximation signal*, while the high-pass one is called *detail signal*.
In practice, it is common to repeat the decomposition process on the resulting approximation signal as illustrated in Fig. 3.4.

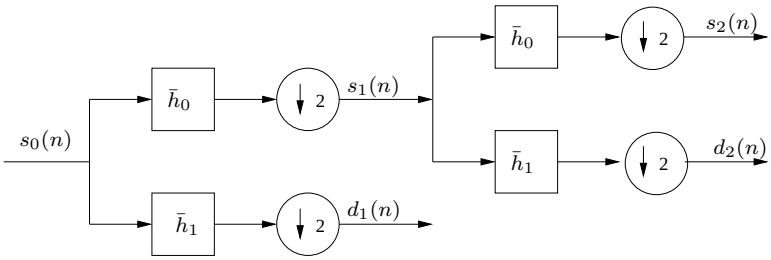



Figure 3.4: One dimensional wavelet decomposition over 2 resolution levels.

Each successive application of the filter bank is called a *level* of decomposition as it provides an approximation of a given resolution level. Thus, we get a representation of the input signal at different levels, referred to as a *multiresolution analysis* [Mallat, 1989].

The relation between the coefficients at two consecutive resolution levels $j$ and $j+1$ is:

$$s_{j+1}(n) = \sum_{k=-\infty}^{+\infty} s_j(k) h_0(k-2n) = (s_j \star \bar{h}_0)(2n) \tag{3.5}$$

$$d_{j+1}(n) = \sum_{k=-\infty}^{+\infty} s_j(k) h_1(k-2n) = (s_j \star \bar{h}_1)(2n) \tag{3.6}$$

where $\star$ denotes the convolution operator and $\bar{h}_0$, $\bar{h}_1$ are the time-reversed versions of the impulse responses $h_0$ and $h_1$ of the analysis filters.

Obviously, it is necessary to recover the original signal from its wavelets coefficients. For this purpose, the outputs $s_{j+1}$ and $d_{j+1}$ are first upsampled by 2, then filtered with the filters with impulse responses $g_0$ and $g_1$ called *synthesis filters*, and finally combined to form a reconstructed signal $\tilde{s}_j$:

$$\tilde{s}_j(n) = \sum_{k=-\infty}^{+\infty} s_{j+1}(k) g_0(n-2k) + \sum_{k=-\infty}^{+\infty} d_{j+1}(k) g_1(n-2k). \tag{3.7}$$

If the reconstructed signal is equal to the original one, the filter bank is said to be a *perfect reconstruction* or a *reversible* filter bank. This property is guaranteed if the analysis and synthesis filters satisfy the following conditions:

$$\begin{cases} \forall\ f \in [0,1], & \hat{h}_0^*(f+\frac{1}{2})\hat{g}_0(f) + \hat{h}_1^*(f+\frac{1}{2})\hat{g}_1(f) = 0 \\ & \hat{h}_0^*(f)\hat{g}_0(f) + \hat{h}_1^*(f)\hat{g}_1(f) = 2 \end{cases} \tag{3.8}$$

where $\hat{h}_0$, $\hat{h}_1$, $\hat{g}_0$ and $\hat{g}_1$ respectively denote the frequency responses of the analysis and synthesis filters.

In addition to the perfect reconstruction property, orthogonality and phase linearity are also two desirable characteristics in the wavelet transform. The orthogonality property is satisfied when the resulting families $\{h_0(k-2n), h_1(k-2n)\}_{n\in\mathbb{Z}}$ is an orthogonal basis of $\ell^2(\mathbb{Z})$. An orthogonal system must have filters of the same length which must be even. The main advantage of orthogonality is to preserve the energy of the coefficients in both the spatial and wavelet domains. Phase linearity is guaranteed by using symmetric filters. Linear phase system filters allow simple symmetric boundary extension techniques that minimize border artifacts. Furthermore, the phase linearity property can be exploited, leading to faster and more efficient filter bank implementations. However, if we except the Haar wavelet transform, a filter bank with real coefficients cannot simultaneously have orthogonality and linear phase [Mallat, 1998].

In the image compression literature, symmetry of filter impulse responses is often considered as a desirable property since it allows to obtain linear phase [Belzer *et al.*, 1995]. As a result, new filters have been considered by relaxing the orthogonality property while preserving the symmetry one. These filters are referred to as *biorthogonal filter banks* and therefore define biorthogonal wavelets. An example is the family of wavelets constructed by Cohen, Daubechis and Feauveau [Cohen *et al.*, 1992; Daubechies, 1992].

To perform DWT for images, Mallat has extended the 1D-DWT to the 2D case by applying separately the one dimensional transform to each dimension. That is, the 2D-DWT is implemented as a 1D row transform followed by 1D column transform, or vice-versa. This is shown in Fig. 3.5 for an input 2D signal $x_j(m,n)$.

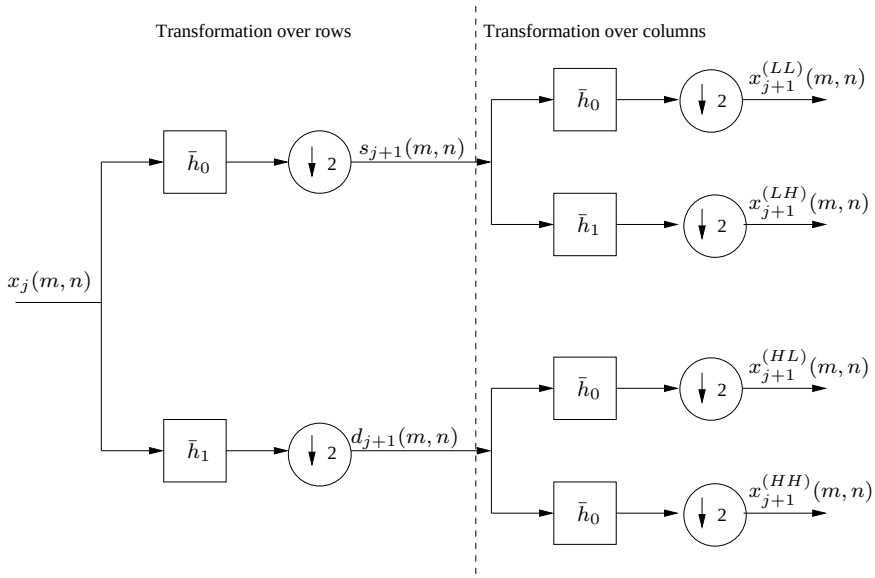



Figure 3.5: Separable filter bank.

Every row of the input image $x_j(m,n)$ is first individually 1D transformed yielding an approximation subband $s_{j+1}(m,n)$ and a detail subband $d_{j+1}(m,n)$. Then, each subband is 1D transformed in a column wise fashion. Thus, four subbands per resolution level are generated: an approximation subband $x_{j+1}^{(LL)}$ and three detail subbands $x_{j+1}^{(HL)}$, $x_{j+1}^{(LH)}$ and $x_{j+1}^{(HH)}$ corresponding respectively to the horizontal, vertical and diagonal orientations. Similarly to the 1D case, an image can be decomposed into different resolution levels by iterating the same process on the approximation subband at each level. Fig. 3.6 illustrates the generated subbands for two resolution levels.

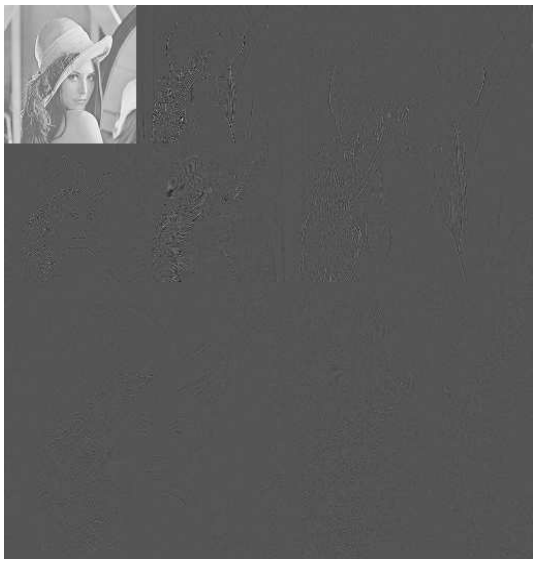


Figure 3.6: Two dimensional subbands after 2 stages of decomposition.

### Lifting schemes

An alternative approach to the classical filter bank for computing DWT is the *Lifting Scheme* (LS) [Sweldens, 1995; Sweldens, 1996]. This approach has been introduced by Sweldens in order to construct biorthogonal wavelets by a simple, reversible and fast process. A generic LS applied to a 1D signal consists of three modules referred to as *split*, *predict* and *update*. Fig. 3.7 shows the block diagram of the LS-based structure.

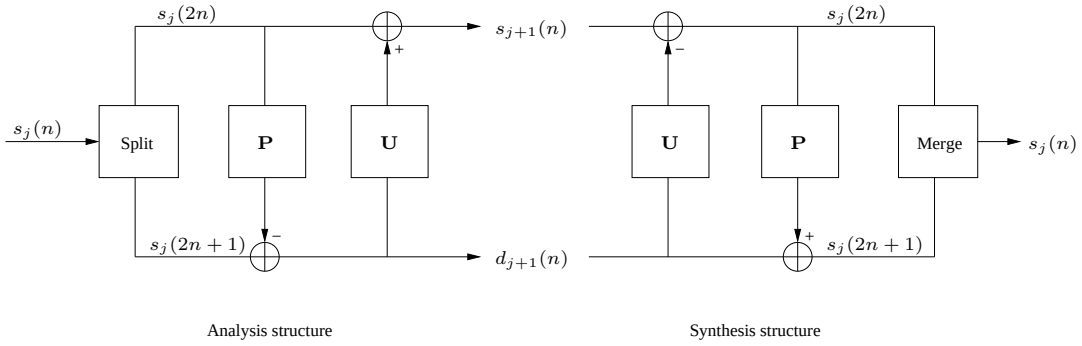



Figure 3.7: A generic one dimensional lifting structure.

• **Split**: In the first step, the input 1D signal $s_j(n)$ is partitioned into two disjoint data sets formed by the even $s_j(2n)$ and odd samples $s_j(2n+1)$, respectively. Such a splitting

is sometimes called *Lazy wavelet transform* or *polyphase decomposition*.

• **Predict**: Then, each sample of one of the two resulting subsets (say the odd one) is predicted from the neighboring even samples and a prediction error or detail signal is computed:

$$d_{j+1}(n) = s_j(2n+1) - \mathbf{p}^\top \mathbf{s}_j(n) \tag{3.9}$$

where $\mathbf{p}$ is the prediction vector, $\mathbf{s}_j(n) = \Big(s_j(2n-2k)\Big)_{k\in\mathcal{P}}$ is the reference vector containing the even samples used in the prediction step, and $\mathcal{P}$ represents the support of the predictor of $s_j(2n+1)$.

When the signal is highly correlated, such a prediction will usually be efficient and thus the detail signal will contain much less information than the original signal. This represents the key issue of the decorrelation step.

• **Update**: Finally, this step generates a coarser approximation of the original signal thanks to a smoothing of the even samples using the detail coefficients:

$$s_{j+1}(n) = s_j(2n) + \mathbf{u}^\top \mathbf{d}_{j+1}(n) \tag{3.10}$$

where $\mathbf{u}$ is the update vector, $\mathbf{d}_{j+1}(n) = \Big(d_{j+1}(n-k)\Big)_{k\in\mathcal{U}}$ is the reference vector containing the detail coefficients used in the update step, and $\mathcal{U}$ represents the support of the update operator.

Similarly to the filter bank based method, a separable 2D-DWT can be simply implemented by separately applying the 1D-LS to the lines and the columns. Again, a multiresolution representation is obtained by recursively repeating these steps to the resulting approximation coefficients.

One of the great advantages of the LS is its reversibility regardless of the choice of the prediction and update operators. Indeed, the inverse transform can always be obtained immediately from the forward transform. The inversion rules are obvious: revert the order of the operations, invert the signs + and -, and replace the splitting step by a merging step. Thus, inverting the three step procedure results in:

• **Undo update**:

$$s_j(2n) = s_{j+1}(n) - \mathbf{u}^\top \mathbf{d}_{j+1}(n) \tag{3.11}$$

• **Undo predict**:

$$s_j(2n+1) = d_{j+1}(n) + \mathbf{p}^\top \mathbf{s}_j(n) \tag{3.12}$$

• **Merge**:

$$\{s_j(n),\ n \in \mathbb{Z}\} = \{s_j(2n),\ n \in \mathbb{Z}\} \cup \{s_j(2n+1),\ n \in \mathbb{Z}\} \tag{3.13}$$

It is worth pointing out that, in lossless coding context, a rounding operator should be applied on the predict and update steps for both analysis and synthesis stages.

LS is known as the second generation wavelets because it has many advantages with respect to the classical method for the construction of wavelets based on the Fourier transform. The most important of them are:

• Simplicity: it is easier to understand and implement.

• The inverse transform is obvious to find and has the same complexity as the forward transform.

• Computational cost reduction: LS reduces to one-half the computational cost of the general Mallat's algorithm.

• It is shown in [Daubechies, Sweldens, 1998] that every biorthogonal filter bank with finite impulse response filters can be decomposed into a finite number of lifting steps, followed by possible multiplication constants.

Due to these advantages, LS has proved to be an efficient tool for still image coding. In what follows, we give few examples of LS which have been reported in [Calderbank *et al.*, 1998]. The most known ones are the family of $(L, \tilde{L})$ integer wavelet transforms where $L$ and $\tilde{L}$ represent the number of vanishing moments of the analysis and synthesis high pass filters, respectively. It can also be noticed that $L$ and $\tilde{L}$ correspond to the number of coefficients of the predict and update filters, respectively.

• (2,2) integer wavelet transform:

$$\begin{cases} d_{j+1}(n) = s_j(2n+1) - \lfloor \frac{1}{2}(s_j(2n) + s_j(2n+2)) + \frac{1}{2} \rfloor \\ s_{j+1}(n) = s_j(2n) + \lfloor \frac{1}{4}(d_{j+1}(n-1) + d_{j+1}(n)) + \frac{1}{2} \rfloor \end{cases} \tag{3.14}$$

Note that this transform is also known as the integer 5/3 transform and it has been retained for the lossless compression mode of the JPEG2000 standard.

• (4,2) integer wavelet transform:

$$\begin{cases} d_{j+1}(n) = s_j(2n+1) - \lfloor \frac{9}{16}(s_j(2n) + s_j(2n+2)) - \frac{1}{16}(s_j(2n-2) + s_j(2n+4)) + \frac{1}{2} \rfloor \\ s_{j+1}(n) = s_j(2n) + \lfloor \frac{1}{4}(d_{j+1}(n-1) + d_{j+1}(n)) + \frac{1}{2} \rfloor \end{cases} \tag{3.15}$$

• (2,4) integer wavelet transform:

$$\begin{cases} d_{j+1}(n) = s_j(2n+1) - \lfloor \frac{1}{2}(s_j(2n) + s_j(2n+2)) + \frac{1}{2} \rfloor \\ s_{j+1}(n) = s_j(2n) + \lfloor \frac{19}{64}(d_{j+1}(n-1) + d_{j+1}(n)) - \frac{3}{64}(d_{j+1}(n-2) + d_{j+1}(n+1)) + \frac{1}{2} \rfloor \end{cases} \tag{3.16}$$

• (4,4) integer wavelet transform:

$$\begin{cases} d_{j+1}(n) = s_j(2n+1) - \lfloor \frac{9}{16}(s_j(2n) + s_j(2n+2)) - \frac{1}{16}(s_j(2n-2) + s_j(2n+4)) + \frac{1}{2} \rfloor \\ s_{j+1}(n) = s_j(2n) + \lfloor \frac{9}{32}(d_{j+1}(n-1) + d_{j+1}(n)) - \frac{1}{32}(d_{j+1}(n-2) + d_{j+1}(n+1)) + \frac{1}{2} \rfloor \end{cases} \tag{3.17}$$

• (6,2) integer wavelet transform:

$$\begin{cases} d_{j+1}(n) = s_j(2n+1) - \lfloor \frac{75}{128}(s_j(2n) + s_j(2n+2)) - \frac{25}{256}(s_j(2n-2) + s_j(2n+4)) \\ \qquad\qquad + \frac{3}{256}(s_j(2n-4) + s_j(2n+6)) + \frac{1}{2} \rfloor \\ s_{j+1}(n) = s_j(2n) + \lfloor \frac{1}{4}(d_{j+1}(n-1) + d_{j+1}(n)) + \frac{1}{2} \rfloor. \end{cases} \tag{3.18}$$

• Another popular transform is the 9/7 one which has been selected for the lossy compression mode of the JPEG2000 standard. It is defined as follows:

$$\begin{cases} \check{d}_{j+1}(n) = s_j(2n+1) + \alpha\Big(s_j(2n) + s_j(2n+1)\Big) \\ \check{s}_{j+1}(n) = s_j(2n) + \beta\Big(\check{d}_{j+1}(n-1) + \check{d}_{j+1}(n)\Big) \\ \tilde{d}_{j+1}(n) = \check{d}_{j+1}(n) + \gamma\Big(\check{s}_{j+1}(n) + \check{s}_{j+1}(n+1)\Big) \\ \tilde{s}_{j+1}(n) = \check{s}_{j+1}(n) + \delta\Big(\tilde{d}_{j+1}(n-1) + \tilde{d}_{j+1}(n)\Big) \\ d_{j+1}(n) = \frac{1}{\zeta}\tilde{d}_{j+1}(n) \\ s_{j+1}(n) = \zeta\tilde{s}_{j+1}(n) \end{cases} \tag{3.19}$$

where

$\alpha = -1.586134342$

$\beta = -0.05298011854$

$\gamma = 0.8829110762$

$\delta = 0.4435068522$

$\zeta = 1.149604398.$

It is worth pointing out that contrary to the previous mentioned examples, which correspond to a Predict-Update (P-U) structure, the 9/7 transform is composed of two lifting stages yielding a P-U-P-U structure.

### Extensions of LS

Motivated by the aforementioned advantages of LS, many research works have been devoted to the extension of LS and construction of new wavelet transforms.

A popular example is known as *Quincunx Lifting Scheme* (QLS) [Gouze *et al.*, 2004; Benazza-Benyahia *et al.*, 2007]. Such a structure is a two dimensional non separable transform performed on a quincunx grid, as shown in Fig. 3.8.

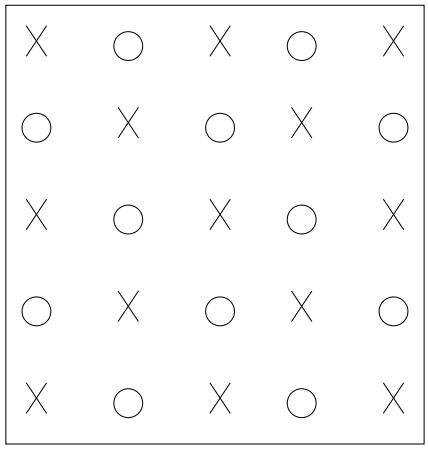


Figure 3.8: Quincunx sampling grid.

The pixels at site '×' are firstly predicted from the neighboring pixels at site '○' and the resulting prediction error corresponds to the detail coefficients. Then, the pixels at site '○' are updated by using the neighboring detail coefficients and the resulting smooth version corresponds to the approximation signal. It is important to note that QLS were found to be efficient for coding satellite images acquired on a quincunx sampling grid.

Similar structures to LS have also been used in nonlinear subband decomposition [Hampson, Pesquet, 1996]. Fig. 3.9 shows the diagram of a nonlinear two-band filter bank.

Note that **H** and **G** can be linear or nonlinear operators. An extension of this structure to $M$-band nonlinear decompositions with maximal decimation and perfect reconstruction properties was developed in [Hampson, Pesquet, 1998].

Solé and Salembier have also proposed another extension of LS referred to as *Generalized Lifting Scheme* (GLS) [Solé, Salembier, 2004].

As it can be seen in Fig. 3.10, the GLS is very similar to the classical LS (Fig. 3.7) except that the operator sums after the prediction and update steps are embedded in a more general framework. For instance, in classical LS, the prediction operator is viewed as a

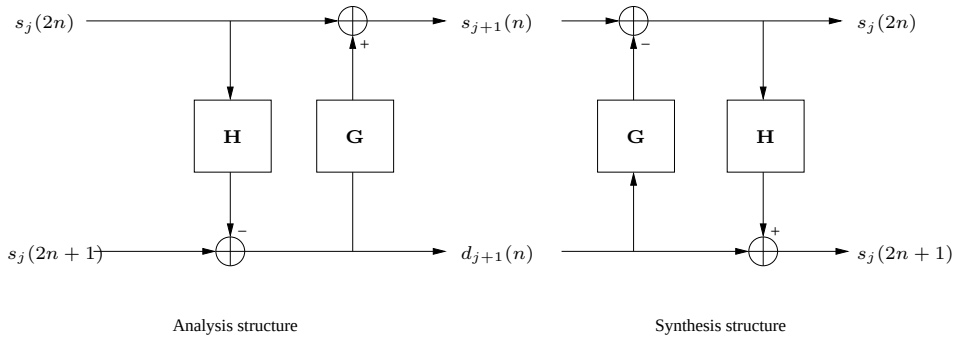



Figure 3.9: Nonlinear two-band filter bank.

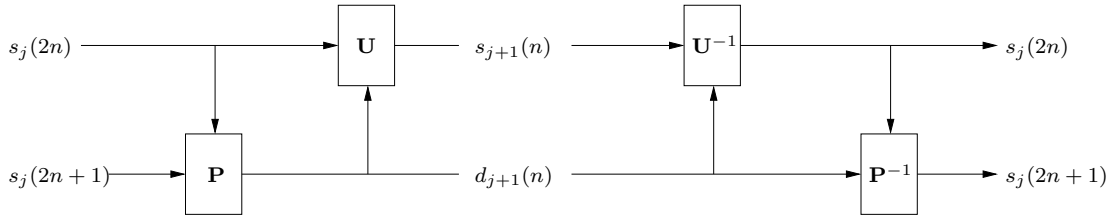



Figure 3.10: Generalized Lifting Scheme.

filter that generates a predicted value which is used to modify the current pixel $s_j(2n+1)$ through a subtraction. However, in the GLS, the prediction is viewed as a function that maps $s_j(2n+1)$ to $d_{j+1}(n)$ taking into account the values of neighboring pixels $s_j(2n)$. One of the main advantages of GLS is its ability to include possibly non linear and/or adaptive operations. More details about the GLS and its reversibility conditions can be found in [Solé, Salembier, 2004].

Moreover, directional transforms with lifting schemes, such as oriented wavelet transforms [Chappelier, Guillemot, 2006] and grouplet [Mallat, 2009], have also been developed for image coding and geometric image processing.

In addition to still image coding applications, many efforts have been deployed in the last years to extend these lifting schemes to other coding applications like video compression [Pesquet-Popescu, Bottreau, 2001], multi-view [Anantrasitichai *et al.*, 2005] and 3D-mesh data compression [Payan *et al.*, 2005]. For example, in the video compression context, motion compensated temporal filtering (MCTF) schemes, based on the LS concept, have attracted a great attention due to their high compression performance comparable with state-of-the-art methods based on the predictive technique [Pau *et al.*, 2004; Cagnazzo *et al.*, 2007]. More precisely, motion estimation (ME) is first performed between the input

frames of the video sequence, and the motion vector field (denoted by $\mathbf{u} = (u_x, u_y)$ in Fig. 3.11) is used for motion compensated operations in both the predict and update steps.

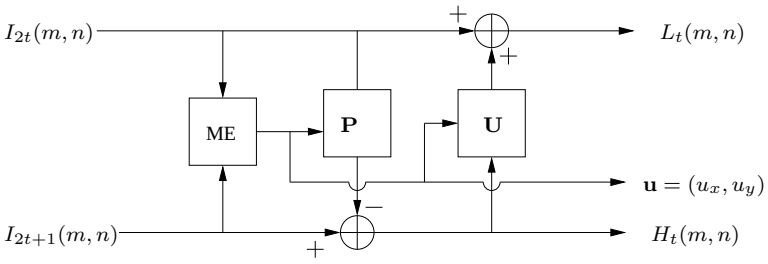



Figure 3.11: Temporal motion compensated lifting scheme.

As can be seen in Fig. 3.11, the odd indexed input frames $I_{2t+1}$ are predicted from the even indexed frames $I_{2t}$, and the prediction error corresponds to the high-pass subband frames $H_t$. Then, the low-pass subband frames $L_t$ are generated by smoothing $I_{2t}$ using the computed detail frames. It is important to note that the prediction and update steps also involve the motion vectors used to match corresponding pixels.

Let $\mathbf{u}^{t_1 \to t_2} = (u_x^{t_1 \to t_2}, u_y^{t_1 \to t_2})$ denote the vector field used to predict the current frame $I_{t_2}$ from the reference one $I_{t_1}$. Therefore, the analysis equations of temporal motion compensated lifting scheme are given by:

$$H_t(m,n) = I_{2t+1}(m,n) - \mathbf{p}^\top \mathbf{I}_{2t}(m,n) \tag{3.20}$$

$$L_t(m,n) = I_{2t}(m,n) + \mathbf{u}^\top \mathbf{H}_t(m,n) \tag{3.21}$$

where $\mathbf{I}_{2t}(m,n) = \left(I_{2(t-k)}(m - u_x^{2(t-k)\to 2t+1}, n - u_y^{2(t-k)\to 2t+1})\right)_{k\in\mathcal{P}}$ (resp. $\mathbf{H}_t(m,n) = \left(H_{t-k}(m+u_x^{2(t-k)+1\to 2t}, n+u_y^{2(t-k)+1\to 2t})\right)_{k\in\mathcal{U}}$) is the reference vector containing the evenly indexed frames (resp. high-pass subband frames) used in the prediction (resp. update) step, and $\mathcal{P}$ (resp. $\mathcal{U}$) represents the support of the temporal prediction (resp. update) operator. However, motion compensated temporal filtering raises the problem of multiple connected/unconnected pixels. Multiple connected pixels correspond to those used to predict more than one pixel in the current frame whereas unconnected pixels are those in the reference frame not used for temporal prediction. To effectively deal with this problem, [Xiong *et al.*, 2004] for example propose a new and general lifting structure, called *Barbell lifting scheme*, in which instead of a single pixel value, a function of a set of pixel values is used as the input to the lifting step.

In parallel with these works devoted to extending LS to a more general framework, other works have also been focused on the design of adaptive lifting schemes. Indeed, in a coding framework, the compactness of any LS-based multiresolution representation depends on the choice of the prediction and update operators.

### Adaptive lifting schemes

Adaptive lifting schemes aim at designing lifting filters to attain a certain objective. To the best of our knowledge, most existing works have mainly focused on the optimization of the prediction operators. For instance, Gerek and *al.* [Gerek, Çetin, 2006] proposed a 2D edge-adaptive lifting scheme by considering three direction angles of prediction (0°, 45° and 135°) and by selecting the orientation which leads to the minimum prediction error. In [Boulgouris, Strintzis, 1999], three separable prediction filters with different numbers of vanishing moments are employed, and then the best prediction is chosen according to the local features. In [Claypoole *et al.*, 1997], a set of linear predictors of different lengths are defined based on a nonlinear function related to edge detector. In [Claypoole *et al.*, 2003], this work is extended by using a two dimensional window and analyzes the transform reversibility, stability and frequency characteristics. Recently, Ding *et al.* [Ding *et al.*, 2007] have built an adaptive directional lifting structure with perfect reconstruction: the prediction is performed in local windows in the direction of high pixel correlation. A good directional resolution is achieved by employing fractional pixel precision level. A similar approach was also adopted in [Chang, Girod, 2007]. An improvement was proposed in [Liu, Ngan, 2008] thanks to a directional adaptive interpolation. In [Bekkouche *et al.*, 2008], an adapted generalized LS, in which the predictor is built upon two filters, is introduced for lossless coding purpose. More precisely, Bekkouche *et al.* studied two kinds of adaptation based on least squares estimation, according to different assumptions, which are either a global or a local second order stationarity of the image. In [Solé, Salembier, 2007], a generalized prediction step is optimized by minimizing at the same time the detail signal energy and entropy given the pixel value probability conditioned to its neighbors pixels values. By assuming that the signal probability density function (pdf) is known, the benefit of this method has firstly been demonstrated for lossless image coding in [Solé, Salembier, 2007]. Then, an extension of this work to sparse image representation and lossy coding contexts has been presented in [Rolon, Salembier, 2007]. Consequently, an estimation of

the pdf should be available at the coder and the decoder side [Rolon *et al.*, 2008; Rolon *et al.*, 2009]. Note that the main drawback of this method as well as directional wavelet transform ones [Ding *et al.*, 2007; Chang, Girod, 2007; Liu, Ngan, 2008; Mallat, 2009; Chappelier, Guillemot, 2006] is that they require to transmit losslessly a side information to the decoder which may affect the whole compression performance especially at low bitrates. Furthermore, such adaptive methods lead to an increase of the computational load required for the selection of the best direction of prediction.

Besides, it can be noticed that there are few works which have discussed the problem of the update filter. Among these works, in [Piella *et al.*, 2002; Abhayaratne *et al.*, 2003], the update operator of a separable LS is adaptively computed thanks to a non linear decision rule using the local gradient information. An extension of this method to 2D non separable schemes that have a spatially adaptive low-pass filter is proposed in [Abhayaratne, 2004; Heijmans *et al.*, 2005]. Another alternative is to adapt the update so that the reconstruction error is minimized when the detail coefficients are canceled [Pesquet-Popescu, 1999; Gouze *et al.*, 2004]. Recently, an adaptation procedure was proposed in [Quellec *et al.*, 2010]. It consists of designing an update and prediction Neville filters whose corresponding primal and dual wavelets have the desired number of vanishing moments. The remaining degrees of freedom are used to optimize a criterion depending on the underlying application. In their paper, the authors were interested in finding the wavelet basis that maximizes the retrieval performance for content-based image retrieval applications.

#### Subband weighting procedure

A key point in the use of efficient bit allocation in wavelet based coders, such as rate-distortion algorithms, consists in assuming an energy preservation transform. More precisely, these algorithms employ quantizers which minimize the distortion in the wavelet domain [Usevitch, 1996]. Thus, such algorithms work well when an orthogonal transform is performed.

However, as mentioned before, the main drawback of biorthogonal wavelets is that they do not satisfy the energy preservation property. Therefore, standard bit allocation techniques become suboptimal for biorthogonal transforms [Usevitch, 1996; Parrilli *et al.*, 2008]. As a result, an efficient bit allocation procedure can be achieved by weighting the average distortion $D_j^{(o)}$ in the wavelet subband, of orientation $o$ at the $j^{\text{th}}$ resolution level, in order

to obtain a good approximation of the distortion in the spatial domain $D_s$. It has been shown in [Usevitch, 1996] that $D_s$ can be expressed as:

$$D_s = \sum_{j=1}^{J} \sum_{o \in \{HL,LH,HH,LL\}} \frac{1}{4^j} w_j^{(o)} D_j^{(o)} \tag{3.22}$$

where $w_j^{(o)}$ is the weight of each wavelet subband at the $j^{\text{th}}$ resolution level and in the orientation $o$. They depend on the filters used for reconstruction.

Note that the weights $w_j^{(o)}$ are equal to 1 if an orthogonal transform is used. Generally, these weights are computed based on the wavelet filters used for the reconstruction process as it is addressed in [Usevitch, 1996; Parrilli *et al.*, 2008]. Theoretically, it is shown in [Parrilli *et al.*, 2008] that such computation is a difficult task especially when using adaptive lifting scheme with more than one decomposition level. To further simplify the weight computation procedure, an alternative method based on the following assumption can also be used. Indeed, as shown in [Usevitch, 1996], if the error signal in a subband (i.e the quantization noise) is white and uncorrelated to the other subband errors, the reconstruction distortion in the spatial domain is related to the distortion in the wavelet domain by Eq. (3.22). Therefore, for each subband $x_j^{(o)}$, a white Gaussian noise of variance $(\sigma_j^{(o)})^2$ is firstly added while keeping the remaining subbands noiseless. Then, the resulting distortion in the spatial domain $\hat{D}_s$ is estimated. Finally, the corresponding subband weight can be estimated as follows:

$$w_j^{(o)} = \frac{\hat{D}_s \times 4^j}{(\sigma_j^{(o)})^2}. \tag{3.23}$$

Once the weights are computed, each wavelet subband resulting from the applied wavelet transform is scaled by a factor $\alpha_j^{(o)}$. Indeed, if the wavelet coefficients $x_j = \{x_j^{(o)}\}$ are multiplied by $\alpha_j^{(o)}$ before quantization, then the quantized values are divided by the same constants $\alpha_j^{(o)}$ before performing the inverse wavelet transform as shown in Fig. 3.12.

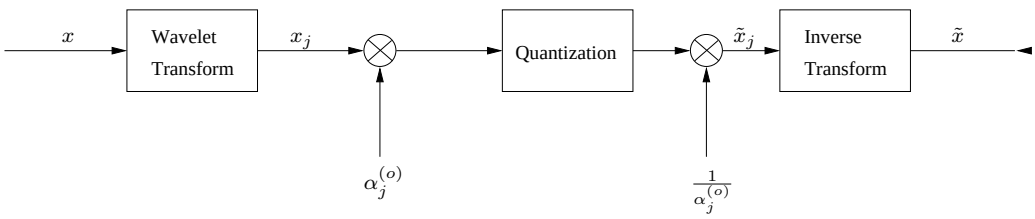



Figure 3.12: Weighted wavelet coefficients.

Therefore, the distortion in the spatial domain (Eq. (3.22)) becomes:

$$D_s = \sum_{j=1}^{J} \sum_{o \in \{HL,LH,HH,LL\}} \frac{1}{4^j} \frac{w_j^{(o)}}{(\alpha_j^{(o)})^2} D_j^{(o)}. \tag{3.24}$$

Since the objective of the subband weighting procedure is to obtain a distortion in the spatial domain which is very close to the distortion in the wavelet domain, the coefficients $\alpha_j^{(o)}$ should be chosen equal to $\sqrt{w_j^{(o)}}$ where $w_j^{(o)}$ is given by Eq. (3.23).

### 3.2.2 Quantization

The goal of the quantization is to reduce the precision of the transformed coefficients by reducing a continuous set of values (or a very large set of possible discrete values) to a relatively small discrete and finite set. Such an operator is lossy (i.e not reversible) and it is not employed in lossless compression schemes. We can distinguish two kinds of quantization: the *scalar quantization* and the *vector quantization*.

A scalar quantization maps each input sample (i.e transform coefficient) to one symbol. A simple example of scalar quantization of an input sample $x_j(m,n)$ into a quantized value $y_j(m,n)$, is the common *uniform quantization*:

$$y_j(m,n) = \lfloor \frac{x_j(m,n)}{Q_{step}} \rfloor Q_{step} \tag{3.25}$$

where $Q_{step}$ is the quantization step size and $\lfloor \cdot \rfloor$ denotes the rounding operator.

Scalar quantization treats each coefficient of $x_j$ independently. If these coefficients are highly dependent, further improvements can be achieved by using vector quantization that assigns one symbol to each vector formed by several samples. It should be noted that scalar quantization requires fewer computations and is thus more often used in practice. Generally, optimal quantizers are designed by minimizing the distortion introduced by the quantization step. This is the case of the JPEG2000 standard, which selects groups of quantization indexes offering the best rate-distortion trade-offs.

### 3.2.3 Entropy coding: examples in wavelet-based codecs

After applying the spatial transform and the quantization step, the generated coefficients must be encoded. However, the coding scheme should enable the quality scalability for progressive reconstruction purposes. This is achieved by sending the wavelet coefficients in decreasing order of their importance. In other words, the most significant ones are

first encoded at a reduced accuracy. So, a first approximation image is produced, which is further gradually refined by decoding the least significant coefficients. An overview of the state-of-the-art embedded wavelet-based codecs can be found in [Fowler, 2003; Fowler, Pesquet-Popescu, 2007]

The pioneering one is the Embedded Zerotree Wavelet coder (EZW) which is an iterative algorithm based on the following three principles [Shapiro, 1993]. Firstly, it exploits the cross-scale similarities of the wavelet coefficients. Secondly, the wavelet coefficients are partially sorted by magnitude. Finally, *successive* approximations of ordered coefficient amplitude are performed to enable a scalable reconstruction in accuracy. More precisely, the coefficients whose magnitudes are above a given threshold are considered as significant coefficients.

As illustrated in Fig. 3.13, "parent" coefficients in a subband can be related to "children" coefficients in the same relative location in a subband at the next higher resolution. Thus, a *zerotree* is formed when a coefficient and all of its descendants are insignificant with respect to a given threshold $T$. Hence, significant coefficients are transmitted first and

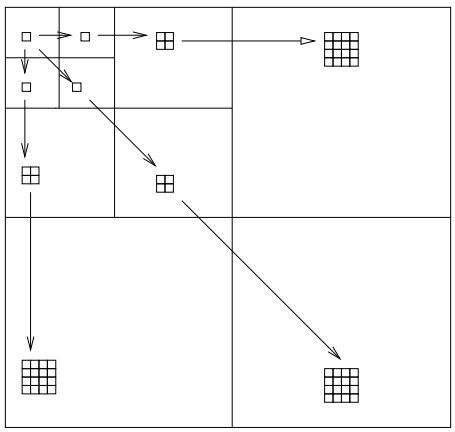


Figure 3.13: Zerotrees in a 2D dyadic transform.

the rest of the coefficients are not coded at this time. As mentioned, EZW is an iterative algorithm, organized around two passes namely the dominant and the subordinate passes. It also employs two lists: the dominant list contains the coordinates of insignificant coefficients whereas the subordinate list contains the magnitude (amplitude) of the significant coefficients. In the dominant pass, coefficients from the dominant list are tested for significance and they are classified and encoded as a zerotree (ZR symbol), an isolated zero (IZ symbol), positive significant (POS symbol), or negative significant (NEG symbol). To

encode these symbols, an adaptive arithmetic coder is used. For significant coefficients, the sign is determined and the magnitude value is appended to the subordinate list. This list is then used in the subordinate pass where the coefficients are sorted in decreasing order and then are refined with an additional bit. The two passes are iterated until the threshold $T$ becomes lower than the minimum wavelet coefficient or until that some target bit rate is reached.

The Set Partitioning In Hierarchical Trees (SPIHT) algorithm aims at improving the zerotree concept by replacing the raster scan with sorted lists [Said, Pearlman, 1996]. The SPIHT has became very popular since it was able to achieve equal or better performances than the EZW algorithm without arithmetic encoder.

Another popular coding algorithm is the Embedded Block Coding with Optimal Truncation (EBCOT) [Taubman, 2000], which has been adopted in the JPEG2000 image compression standard [Taubman, Marcellin, 2001]. The basic idea of this algorithm is to employ extensively conditioned, multiple-context adaptive arithmetic coding. To code the wavelet coefficients of an image, JPEG2000 firstly partitions each transform subband into small 2D rectangular blocks called *codeblocks*, which are typically of size $32 \times 32$ or $64 \times 64$ pixels. Then, the JEPG2000 encoder independently generates an embedded bitstream for each codeblock. To assemble the individual bitstreams into a single, final bitstream, each codeblock bitstream is truncated in some fashion, and the truncated bitstreams are concatenated together to form the final bitstream. The method for codeblock bitstream truncation is typically a Lagrangian rate-distortion optimal technique, referred to as *Post-Compression Rate-Distortion (PCRD) optimization*. This technique is performed simultaneously across all of the codeblocks from the image, producing an optimal truncation point for each codeblock. Indeed, the PCRD optimization distributes the total rate for the image spatially across the codeblocks in an optimal rate-distortion fashion such that codeblocks with higher energy, which tend to more heavily influence the distortion measure, receive greater rate. Additionally, the truncated codeblock bitstreams are interleaved in an optimal order such that the final bitstream is close to being rate-distortion optimal at many truncation points.

More recently, a new Embedded Zero Block Coding algorithm (EZBC) has been proposed [Hsiang, Woods, 2000]. Its concept is inspired by the success of two popular embedded image-coding techniques: zero tree-block coding such as SPIHT, and context modeling of the subband/wavelet coefficients such as EBCOT [Taubman, 2000]. Indeed, 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 considering all the pixels, only a small number of elements in the lists 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 [Taubman, 2000], 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 full bitplane, with an implied higher computational cost.

The main advantage of these embedded codecs is that the encoder can terminate the encoding at any point thereby allowing a target bitrate to be met exactly. Similarly, the decoder can also stop decoding at any point resulting in the image that would have been produced at the rate corresponding to the truncated bitstream. Among these various codecs, it is shown in [Hsiang, Woods, 2000] that EZBC and JPEG2000 codecs are the most effective in terms of compression and reconstruction quality. In this thesis, we have chosen the JPEG2000 codec which yields excellent results performance in terms of compression efficiency and reconstruction quality.

## 3.3 Stereo image coding: state-of-the-art

After the MVC standardization [Chen *et al.*, 2009], most of the ongoing activity on stereoscopic still image coding carried out independently of any standardization activity [Lau *et al.*, 2007]. Indeed, different approaches have been reported concerning still images.

### 3.3.1 Basic approach for joint coding of stereo images

A straightforward approach consists of separately coding each image by using existing still image coders. However, this method is not so efficient since the images are often highly correlated. Therefore, more efficient coding schemes have been designed to take into account the inter-images redundancies [Lukacs, 1986; Perkins, 1992]. The basic-source coding approach is a combination of *inter-view prediction*, to exploit the cross-view redun-

dancies, and *transform coding of the resulting signals* to exploit the spatial redundancies (see Fig. 3.14).

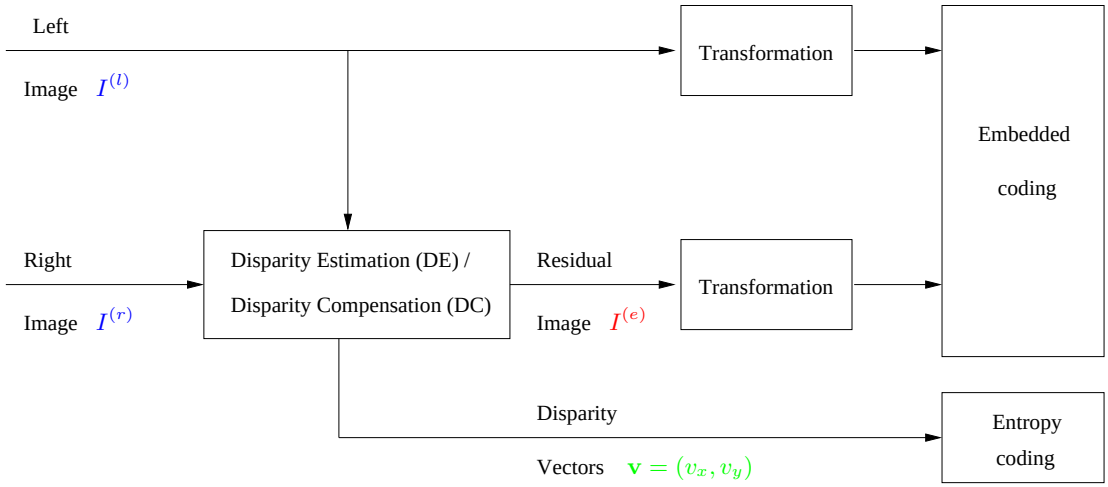



Figure 3.14: Block diagram of a generic stereo image coding scheme.

As shown in Fig. 3.14, the generic stereo image coding scheme involves three steps. In the first step, one image (the left one $I^{(l)}$) is firstly selected as a reference image, and the other image (the right one $I^{(r)}$) is selected as a target image. After that, the disparity map $\mathbf{v} = (v_x, v_y)$ between the right and the left images is estimated. In the second step, the target image $I^{(r)}$ is predicted from the reference one $I^{(l)}$ along the disparity field, and the difference between the original target image and the predicted one, called *residual image* $I^{(e)}$ or *Disparity Compensated Difference* (DCD), is generated:

$$I^{(e)}(m,n) = I^{(r)}(m,n) - I^{(l)}(m + v_x(m,n), n + v_y(m,n)) \tag{3.26}$$

Finally, the reference image, the residual one and the disparity map are encoded. Obviously, recovering the stereo image pair from these data is an easy task: the reference and the residual images are first decoded. Then, the right image is decoded using the disparity vectors.

This approach is known as Disparity Compensation (DC) due to its similarity with Motion Compensation (MC) techniques which are popular for video coding [Wang, Chen, 1999].

### 3.3.2 Characteristics of the disparity map and the DCD

Because of the analogy between MC and DC, it could seem intuitive to apply the existing video coding techniques to stereo images. However, it must be emphasized that the

disparity map and the DCD have different characteristics than their counterparts in video coding. Generally, motion vectors do not have any "dominant" direction. Unlikely, the disparity vectors tend to have an horizontal direction, they are all positively oriented and their range of values is larger than that of motion vectors. The occlusion areas are due to *all* the objects in the scene. It is very different from motion estimation, where only moving objects contribute to occlusions. Besides, typical moving images contain background objects that are static and a small part of the image corresponds to moving areas, this property being exploited at high compression rates whereas, in stereo images, all the objects are displaced from a view to another. Consequently, DCD may have higher average energy than a displaced frame difference and thus may require a higher bitrate. Besides, Moellenhoff and Maier [Moellenhoff, Maier, 1998a] have shown that DCD usually contains very narrow vertical edges surrounded by large regions of very low intensity. For instance, some studies of the characteristics of the DCD have been conducted in [Moellenhoff, Maier, 1998a; Perez *et al.*, 2001]. In the sequel, we give an overview of the most popular stereo image coders.

### 3.3.3 Overview of data compression schemes for stereo images

As mentioned in the beginning of this section, the basic approach for joint coding of stereo images requires the encoding of the disparity field, the reference image and the residual one. Indeed, the disparity map is generally losslessly encoded using DPCM with an entropy coder as it is used with motion vectors in many video coding standards [Yang, Ngi, 2004]; whereas the reference and the residual images are encoded in different transform domains. The most widely used one is the Discrete Cosine Transform [Woo, Ortega, 1997; Ahlvers *et al.*, 2003; Aydinoglu, Hayes, 1995]. However, Moellenhoff and Maier have shown in [Moellenhoff, Maier, 1998a] that DCT yields to a moderate energy packing of the DCD and proposed some improvements including an updated quantization matrix and data ordering in [Moellenhoff, Maier, 1998b]. An important problem for DCT-based coding scheme is that it causes unpleasant visual artifacts at the block boundaries of the decoded image at very low bitrate. This is the reason why the wavelet transform has been envisaged in the recent works.

Indeed, in [Q. Jiang, Hayes, 1999], both the estimation and the disparity compensation are performed in the wavelet domain. A Subspace Projection Technique (SPT) is used

for encoding the wavelet coefficients. More precisely, this method starts by applying a wavelet transform to the original SI pair. We denote by $\{a_J^{(r)}, (d_j^{(r,o)})_{1\leq j\leq J},\ o \in \{1,2,3\}\}$ (resp. $\{a_J^{(l)}, (d_j^{(l,o)})_{1\leq j\leq J},\ o \in \{1,2,3\}\}$) the resulting approximation and detail subbands for the right (resp. left) image. A block-based DE is performed between the corresponding subbands $(a_J^{(r)}, a_J^{(l)})$ and $(d_J^{(r,o)}, d_J^{(r,o)})$. Then, a DC of each block of the image subbands is carried out, leading to the predicted subbands $\{\hat{a}_J^{(r)}, \hat{d}_j^{(r,o)},\quad o \in \{1,2,3\}\}$. Finally, the computation of the DCD is obtained by projecting each block of the approximation subband of the target image $a_j^{(r)}$ onto the subspace $S = \text{span}\{\hat{a}_J^{(r)}, \hat{d}_J^{(r,o)},\ o \in \{1,2,3\}\}$, yielding the projection

$$\check{a}_J^{(r)} = \alpha_0 \hat{a}_J^{(r)} + \sum_{o=1}^{3} \alpha_o \hat{d}_J^{(r,o)} \tag{3.27}$$

where $(\alpha_0, \alpha_1, \alpha_2, \alpha_3)$ are computed by a least squares approach. Consequently, the approximation subband of the residual image $a_J^{(e)}$ and the three detail subbands are given by:

$$a_J^{(e)} = a_J^{(r)} - \check{a}_J^{(r)} \tag{3.28}$$

$$d_j^{(e,o)} = d_j^{(r,o)} - \hat{d}_j^{(r,o)}\ o \in \{1,2,3\}. \tag{3.29}$$

The drawback of this technique is that it is computationally expensive as different bases vectors need to be constructed for representing each block of wavelet coefficient.

Furthermore, several attempts have been carried out to improve the coding efficiency of wavelet-based coders. In [Boulgouris, Strintzis, 2002], an efficient exploitation of the zerotree algorithm [Shapiro, 1993] is performed to shorten the embedded bitstreams of the wavelet coefficients of both the reference image and the DCD. In [Palfner *et al.*, 2002], the authors propose to introduce some modifications on SPIHT coder, which is basically developed for monocular images, in order to be better adapted to stereo images coding. Ellinas et *al.* [Ellinas, Sangriotis, 2006] have replaced the conventional wavelet coder by a morphological wavelet-based coder, which can achieve a gain of about 1 dB.

Moreover, a few works aiming at designing adaptive schemes have also been conducted. Indeed, based on the results reported in [Moellenhoff, Maier, 1998a] which indicate that pixels in residual images are less correlated than those in natural images, Frajka and Zeger [Frajka, Zeger, 2003] have analyzed the local correlation of pixels across block-boundaries in these residual images. They show that this local correlation drops significantly at block-boundaries. For this reason, they have proposed to design a hybrid coding scheme where

DCT is employed for the best matching blocks and the Haar wavelet transform for the occluded ones. To classify occluded blocks, the authors first evaluate the energy of all blocks in the residual image. Then, any block having an energy above a certain threshold is classified as an occluded block. An adaptive lifting scheme is also presented in [Darazi *et al.*, 2009]. Indeed, a luminance correction is firstly applied to the original images. Then, the prediction filtering is locally adapted based on local horizontal and vertical gradient information of the reference image. While this approach can achieve good results in terms of compression ratio, it is not efficient in a lossy coding context (especially at low bit rate) since it is very sensitive to the quality of the reference image. In a more recent work [Maalouf, Larabi, 2010], the authors propose to generate the disparity map and the DCD by applying a bandelet transform [Pennec, Mallat, 2005] to the left and the right images. They apply first this kind of transform to the original images in order to find the different blocks sharing the same geometric flow direction. Then, the disparity is estimated by minimizing a cost function defined on the geometric property of each block. Finally, the disparity map as well as the bandelet coefficients of the reference and the residual images are encoded. The main limitation of this method is that it requires the transmission of a side information corresponding to the size of each block which will affect its performance at very low bitrate.

In addition to these works, we should note also that Annex I of Part II of the JPEG2000 standard is dedicated to multi-component image coding [Taubman, Marcellin, 2001]. It is a two stage compression scheme. For instance, in our case, each view of the stereoscopic image can be seen as a single component. At the first stage, a decorrelation of the SI pair is performed. Note that this decorrelation step must use a reversible transform in order to exactly recover the original SI pair in the context of lossless coding. As a result, a pair $(\tilde{I}, I^{(e)})$ is produced by using a variation of the Haar transform. More specifically, the equations which define $\tilde{I}$ and $I^{(e)}$ are the following ones:

$$\begin{cases} I^{(e)}(m,n) & = I^{(r)}(m,n) - I^{(l)}(m+v_x, n+v_y) \\ \tilde{I}(m+v_x, n+v_y) & = \lfloor (I^{(r)}(m,n) + I^{(l)}(m+v_x, n+v_y))/2 \rfloor \quad \text{if } (m+v_x, n+v_y) \in \mathcal{S} \\ \tilde{I}(m,n) & = I^{(l)}(m,n) \qquad\qquad\qquad\qquad\qquad\qquad\quad \text{if } (m,n) \notin \mathcal{S} \end{cases} \tag{3.30}$$

where $\mathcal{S}$ is the set of connected pixels which represent the pixels in the reference image used in the disparity compensation process [Pesquet-Popescu, Bottreau, 2001]. Then, a

wavelet transform is applied separately to $I^{(e)}$ and $\tilde{I}$.

To conclude this part, it is worth pointing out that the common idea behind all the existing stereo images compression methods consists of encoding the same signals which are the disparity map, the reference image and the residual one. However, the main differences between them can be classified in three classes categories. The first category of methods attempt to propose an efficient disparity estimation approach to better exploit the inter-view redundancies [Woo, Ortega, 1997; Woo, Ortega, 2000]. The second one tries to improve the entropy coding of the transformed coefficients [Palfner *et al.*, 2002; Ellinas, Sangriotis, 2006]. The third category aims at improving the decomposition (i.e the transform) applied to the images before the entropy encoding process [Frajka, Zeger, 2003; Darazi *et al.*, 2009; Maalouf, Larabi, 2010]. It is in this context that we will focus on our contribution in the next chapter.

## 3.4 Conclusion

In this chapter, we have given some background on image compression tools. More precisely, the concept of wavelet transforms, and particularly lifting schemes, has been introduced. We have also reminded the main issues and the previous works on stereo image coding. In the next chapter, we will present a novel joint coding scheme that allows a gradual and finally exact decoding of the stereo image pairs.

# Chapter 4

# Vector Lifting Schemes for stereo image coding

## 4.1 Introduction

In this chapter, we propose a novel approach, based on the Vector Lifting Schemes (VLS), which offers the advantage of generating two compact multiresolution representations of the left and the right images.

The remainder of this chapter is organized as follows. In Section 4.2, we present the proposed coding structure of which two versions (VLS-I and VLS-II) are given. In Section 4.3, we conduct a theoretical analysis of the proposed schemes in terms of prediction efficiency. Finally, in Section 4.4, experimental results are given and some conclusions are drawn in Section 4.5.

## 4.2 Proposed Vector Lifting Schemes

### 4.2.1 Motivation

Fig. 4.1 illustrates our proposed approach based on a joint multiscale decomposition of the left and the right images. It consists of coding the reference image $I^{(l)}$ in intra mode (purely spatial), whereas the other image $I^{(r)}$ is coded by exploiting cross-image redundancies via the available disparity map. The decomposition strategy is inspired from Vector Lifting Schemes (VLS) [Benazza-Benyahia *et al.*, 2002] and it has been presented in [Kaaniche *et al.*, 2007; Kaaniche *et al.*, 2009a]. Our approach is quite different from the conventional

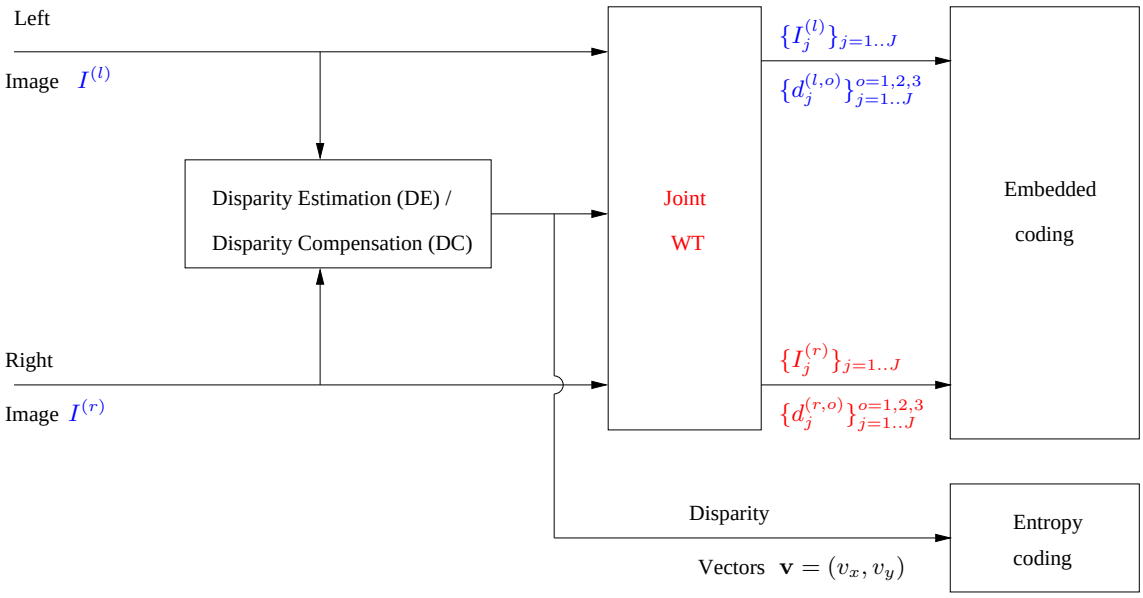



Figure 4.1: Block diagram of the novel stereo image coding scheme.

one. Indeed, our main contribution is that the proposed coding scheme does not generate any residual image, but directly two compact multiresolution representations of the left and right images. Two versions of the VLS will be described in the following.

### 4.2.2 Generic VLS decompositions

For the sake of simplicity, a separable decomposition is considered in this chapter. Therefore, it is enough to address the decomposition of a given line $m$. The corresponding analysis structure is shown in Fig. 4.2.

More precisely, at each resolution level $j$, the even and odd samples of the approximation (scaling) coefficients $I_j^{(l)}(m,2n)$, $I_j^{(r)}(m,2n)$, $I_j^{(l)}(m,2n+1)$ and $I_j^{(r)}(m,2n+1)$ of $I_j^{(l)}$ and $I_j^{(r)}$ respectively are the input coefficients of the lifting scheme. Furthermore, we denote by $\mathbf{v}_j = (v_{x,j}, v_{y,j})$ the available disparity vectors which are obtained by sampling and dividing by $2^j$ the initial (full resolution) disparity vectors $\mathbf{v} = (v_x, v_y)$ since the dimensions of the initial images are divided by $2^j$ at the $j$-th resolution level:

$$\mathbf{v}_j(m,n) = \frac{1}{2^j}\mathbf{v}(2^j m, 2^j n). \tag{4.1}$$

It is important to note that Eq. (4.1) may yield non-integer values of the disparity vectors:

$$\begin{cases} v_{x,j} = \lfloor v_{x,j} \rfloor + \delta_1 \\ v_{y,j} = \lfloor v_{y,j} \rfloor + \delta_2 \end{cases} \tag{4.2}$$

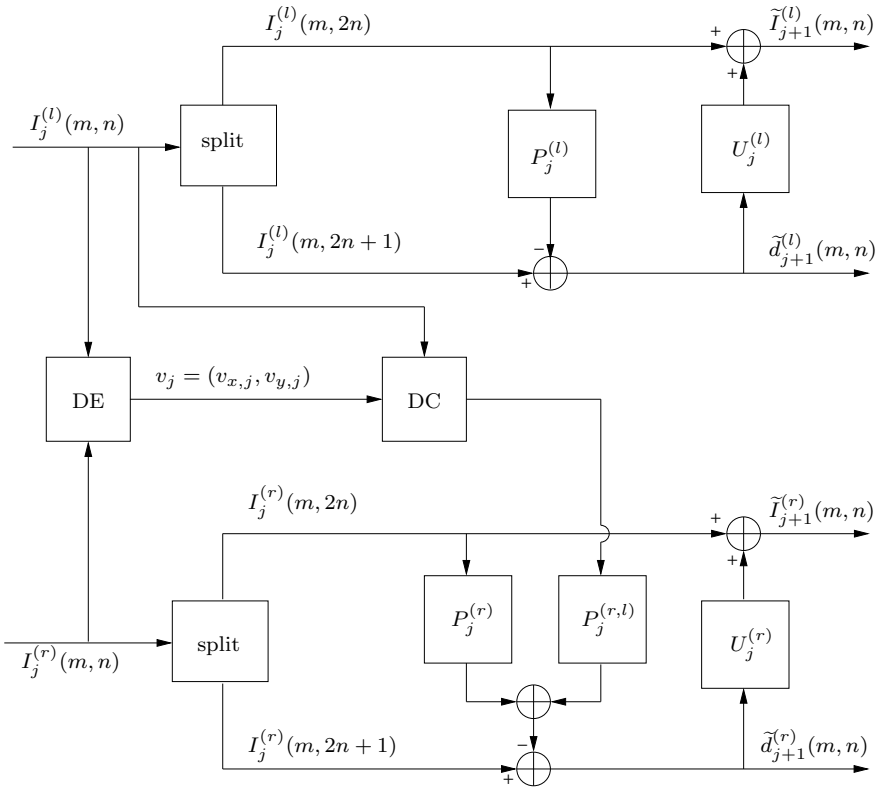



Figure 4.2: Principle of the VLS-I decomposition.

where $\lfloor \cdot \rfloor$ designates the lower rounding operation and $(\delta_1, \delta_2) \in [0,1)^2$. Therefore, if the components of $\mathbf{v}_j$ are integer-valued, for any given pixel $(m,n)$ in the right image corresponds a pixel in the disparity-compensated left image $I_j^{(l)}(m + v_{x,j}(m,n), n + v_{y,j}(m,n))$. Otherwise, the corresponding disparity-compensated intensity results from the usual bilinear interpolation $\alpha_1 A_1 + \alpha_2 A_2 + \alpha_3 A_3 + \alpha_4 A_4$, where

$$\begin{cases} A_1 = I_j^{(l)}(m + \lfloor v_{x,j} \rfloor, n + \lfloor v_{y,j} \rfloor), & \alpha_1 = (1-\delta_1)(1-\delta_2) \\ A_2 = I_j^{(l)}(m + \lfloor v_{x,j} \rfloor + 1, n + \lfloor v_{y,j} \rfloor), & \alpha_2 = \delta_1(1-\delta_2) \\ A_3 = I_j^{(l)}(m + \lfloor v_{x,j} \rfloor, n + \lfloor v_{y,j} \rfloor + 1), & \alpha_3 = (1-\delta_1)\delta_2 \\ A_4 = I_j^{(l)}(m + \lfloor v_{x,j} \rfloor + 1, n + \lfloor v_{y,j} \rfloor + 1), & \alpha_4 = \delta_1\delta_2. \end{cases} \tag{4.3}$$

For simplicity, the notation $I_j^{(l)}(m + v_{x,j}, n + v_{y,j})$ will also be employed in this case. Note that even though this bilinear interpolation can yield a disparity compensated image with real values, the proposed VLS satisfy the perfect reconstruction property since the decompositions are applied to the reference and the target images.

The objective of the vector lifting scheme is to simultaneously exploit the dependence existing between $I_j^{(l)}$ and $I_j^{(r)}$ by producing 2 kinds of outputs: the detail coefficients $\widetilde{d}_{j+1}^{(l)}$,

$\widetilde{d}^{(r)}_{j+1}$ and the approximation ones $\widetilde{I}^{(l)}_{j+1}$, $\widetilde{I}^{(r)}_{j+1}$ for both images. Similar lifting structures operating along the image columns allow us to generate the approximation coefficients $I^{(l)}_{j+1}$ and $I^{(r)}_{j+1}$, as well as the associated detail coefficients in the horizontal, vertical and diagonal directions at the resolution level $(j+1)$.[1] A wide range of nonlinear operators can be applied to reduce the intra and inter-image redundancies. However, for tractability purposes, we will only use combinations of shift operators, linear filters and rounding operations. For the reference image $I^{(l)}$, the detail coefficients can be interpreted as intra-image prediction errors at resolution $(j+1)$ expressed as:

$$\widetilde{d}^{(l)}_{j+1}(m,n) = I^{(l)}_{j}(m,2n+1) - \lfloor \sum_{k\in\mathcal{P}^{(l)}_j} p^{(l)}_{j,k} I^{(l)}_{j}(m,2n-2k) \rfloor \tag{4.4}$$

where the set $\mathcal{P}^{(l)}_j$ and the coefficients $p^{(l)}_{j,k}$ denote respectively the support and the weights of the predictor of $I^{(l)}_j(m,2n+1)$. Then, at the update step, the approximation coefficients are computed as follows:

$$\widetilde{I}^{(l)}_{j+1}(m,n) = I^{(l)}_{j}(m,2n) + \lfloor \sum_{k\in\mathcal{U}^{(l)}_j} u^{(l)}_{j,k} \widetilde{d}^{(l)}_{j+1}(m,n-k) \rfloor, \tag{4.5}$$

where the set $\mathcal{U}^{(l)}_j$ is the spatial support of the update operator whose coefficients are $u^{(l)}_{j,k}$. The reversibility of the basic lifting scheme is ensured since the prediction in Eq. (4.4) only makes use of even indexed samples. The main difference between a vector lifting scheme and a basic one is that for the image $I^{(r)}$, the prediction of the odd sample $I^{(r)}_j(m,2n+1)$ involves even samples from the same image *and* also neighbors of the matching sample taken from the reference image. The location of the latter is derived from the transformed disparity vector $\mathbf{v}_j = (v_{x,j}, v_{y,j})$ associated with the pixel $(m,2n+1)$ to be predicted, yielding:

$$\begin{aligned}\widetilde{d}^{(r)}_{j+1}(m,n) =& I^{(r)}_{j}(m,2n+1) - \lfloor \sum_{k\in\mathcal{P}^{(r)}_j} p^{(r)}_{j,k} I^{(r)}_{j}(m,2n-2k) \\ &+ \sum_{k\in\mathcal{P}^{(r,l)}_j} p^{(r,l)}_{j,k} I^{(l)}_{j}(m+v_{x,j}(m,2n+1), 2n+1+v_{y,j}(m,2n+1)-k) \rfloor\end{aligned} \tag{4.6}$$

where $\mathcal{P}^{(r)}_j$ (resp. $\mathcal{P}^{(r,l)}_j$) denotes the spatial support of the intra-image (resp. inter-images) whereas its weights are designated by $p^{(r)}_{j,k}$ (resp. $p^{(r,l)}_{j,k}$). Consequently, the vector lifting

[1] As we apply a separable decomposition, we denote by $\widetilde{I}_j$ the approximation coefficients after the first mono-dimensional processing at the $(j-1)^{th}$ level, and by $I_j$ the final approximation subband.

involves a hybrid predictor that exploits at the same time the intra and inter-image redundancies in the stereo pair. The update step for $\widetilde{I}_{j+1}^{(r)}$ can be performed similarly to Eq. (4.5). The decomposition is iterated on the columns $n$ of the resulting subbands, leading to $2 \times 4$ sub-images for the left and right images at each resolution level $j$ and the decomposition is again repeated on the approximation sub-images over $J$ resolution levels. It is worth pointing out that the disparity based vector lifting scheme is perfectly reversible and that it maps integers to integers. However, an appropriate choice of the involved prediction and update operators remains necessary in order to generate compact representations of both $I^{(l)}$ and $I^{(r)}$.

**An example: VLS-I**

To illustrate the ability of the considered vector lifting structure to produce a compact decomposition, we provide a simple example (denoted by VLS-I) of the considered lifting structure. The image $I^{(l)}$ is first decomposed following the well-known integer-to-integer 5/3 scheme retained for the lossless mode of JPEG 2000 [Sweldens, 1995]. This decomposition can be described, using our notations, by the spatial supports for the prediction and update operators: $\mathcal{P}_j^{(l)} = \{-1, 0\}$, $\mathcal{U}_j^{(l)} = \{0, 1\}$ and the prediction and update weights are set to: $p_{j,-1}^{(l)} = p_{j,0}^{(l)} = \frac{1}{2}$, $u_{j,0}^{(l)} = u_{j,1}^{(l)} = \frac{1}{4}$. Thus, equations (4.4) and (4.5) become:

$$\widetilde{d}_{j+1}^{(l)}(m,n) = I_j^{(l)}(m, 2n+1) - \lfloor \frac{1}{2}(I_j^{(l)}(m,2n) + I_j^{(l)}(m,2n+2)) \rfloor \tag{4.7}$$

$$\widetilde{I}_{j+1}^{(l)}(m,n) = I_j^{(l)}(m,2n) + \lfloor \frac{1}{4}(\widetilde{d}_{j+1}^{(l)}(m,n-1) + \widetilde{d}_{j+1}^{(l)}(m,n)) \rfloor, \tag{4.8}$$

The hybrid intra-inter prediction step related to $I^{(r)}$ is then expressed via the following spatial supports: $\mathcal{P}_j^{(r)} = \{-1, 0\}$, $\mathcal{P}_j^{(r,l)} = \{0\}$. In other words, the prediction mask contains the same spatial prediction indices as those used in the 5/3 scheme and the co-located position in the left image. In this case, the detail signal given by Eq. (4.6) becomes:

$$\begin{aligned} \widetilde{d}_{j+1}^{(r)}(m,n) = I_j^{(r)}(m,2n+1) - \lfloor & p_{j,0}^{(r)} I_j^{(r)}(m,2n) + p_{j,-1}^{(r)} I_j^{(r)}(m,2n+2) \\ & + p_{j,0}^{(r,l)} I_j^{(l)}(m + v_{x,j}(m,2n+1), 2n+1+v_{y,j}(m,2n+1)) \rfloor. \end{aligned} \tag{4.9}$$

As the detail coefficients can be viewed as prediction errors, the prediction coefficients $p_{j,k}^{(r)}$ and $p_{j,k}^{(r,l)}$ can be optimized at each resolution level by solving the well-known Yule-Walker equations. More precisely, by omitting the rounding operators and imposing the following symmetry property: $p_{j,-1}^{(r)} = p_{j,0}^{(r)}$ (which allows us to obtain linear phase filters often

considered as desirable for image coding [Belzer *et al.*, 1995]), the detail signal $\widetilde{d}_{j+1}^{(r)}(m,n)$ will be expressed as:

$$\begin{aligned}\widetilde{d}_{j+1}^{(r)}(m,n) = I_j^{(r)}(m,2n+1) - p_{j,0}^{(r)}\Big(I_j^{(r)}(m,2n) + I_j^{(r)}(m,2n+2)\Big) \\ -p_{j,0}^{(r,l)} I_j^{(l)}(m + v_{x,j}(m,2n+1), 2n+1+v_{y,j}(m,2n+1)).\end{aligned} \tag{4.10}$$

The prediction weighting vector $\mathbf{p}_j = (p_{j,0}^{(r)}, p_{j,0}^{(r,l)})^\top$ is then chosen so as to minimize the variance of $\widetilde{d}_{j+1}^{(r)}(m,n)$. Consequently, the Yule-Walker equations reduce to:

$$\mathsf{E}\left[\mathbf{I}_j(m,n)\mathbf{I}_j(m,n)^\top\right]\mathbf{p}_j = \mathsf{E}\left[I_j^{(r)}(m,2n+1)\mathbf{I}_j(m,n)\right], \tag{4.11}$$

where

$$\mathbf{I}_j(m,n) = \Big(I_j^{(r)}(m,2n) + I_j^{(r)}(m,2n+2), I_j^{(l)}(m + v_{x,j}(m,2n+1), 2n+1+v_{y,j}(m,2n+1)\Big)^\top \tag{4.12}$$

and $\mathsf{E}[.]$ denotes the mathematical expectation.

Concerning the update step, it is possible to generalize the optimization procedure described in [Gouze *et al.*, 2004; Pesquet-Popescu, 1999] in order to adapt the underlying operators to the statistical properties of the input image. A straightforward alternative solution that we have retained in our experiments has consisted in choosing the same update operator at any resolution level, the update employed for $I^{(r)}$ being the same as the two-tap filter employed for $I^{(l)}$ in Eq .(4.5).

### 4.2.3 An improved VLS

If we examine the previous VLS-I structure, we can note that it presents the following drawback. It can be seen from Eq. (4.6) that the detail coefficients of the right image are computed by using some samples of the left image. Then, these detail coefficients are used through the update step to compute the approximation coefficients of the right image (see Fig. 4.2). Therefore, a potential drawback of this structure is that it generates an update leakage effect in the sense that the information coming from the left view in re-injected into the approximation coefficients of the right view.

An alternative solution is given by the predict-update-predict (P-U-P) lifting structure shown in Fig. 4.3.

The improved decomposition is described as follows:

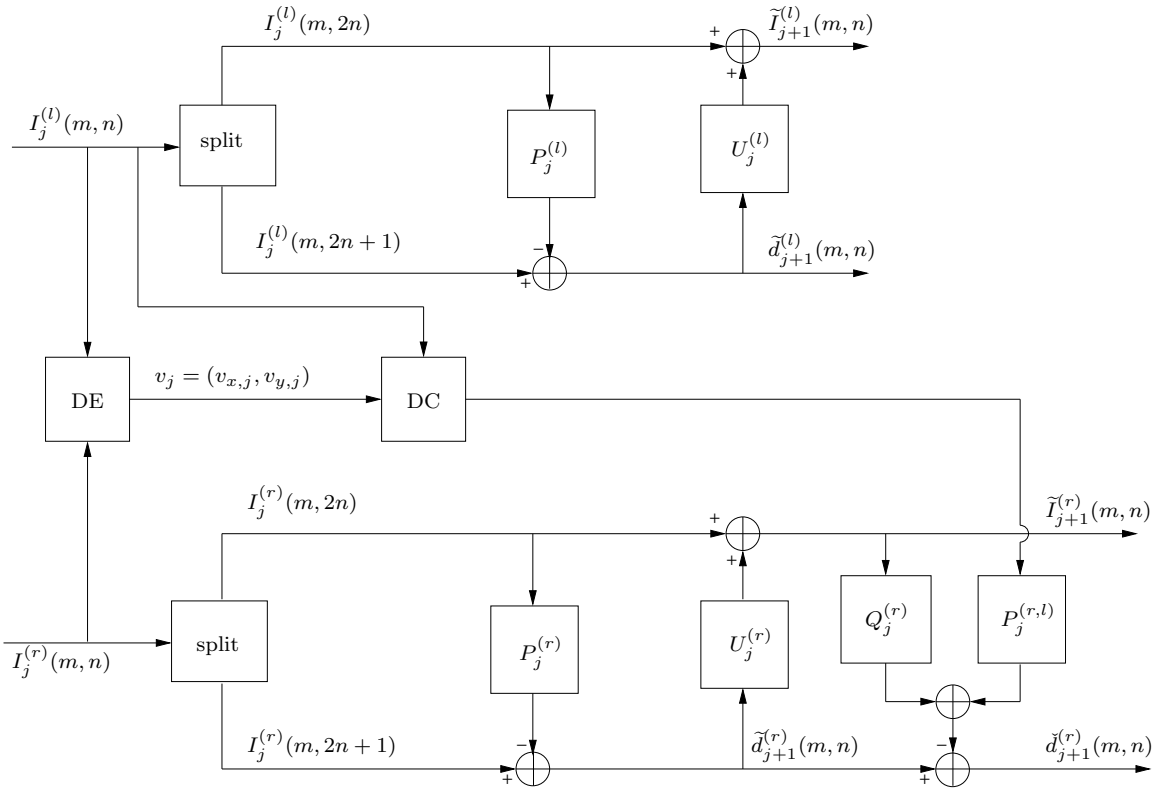



Figure 4.3: Principle of the VLS-II decomposition.

$$\widetilde{d}_{j+1}^{(r)}(m,n) = I_j^{(r)}(m,2n+1) - \lfloor \sum_{k\in\mathcal{P}_j^{(r)}} p_{j,k}^{(r)} I_j^{(r)}(m,2n-2k) \rfloor, \tag{4.13}$$

$$\widetilde{I}_{j+1}^{(r)}(m,n) = I_j^{(r)}(m,2n) + \lfloor \sum_{k\in\mathcal{U}_j^{(r)}} u_{j,k}^{(r)} \widetilde{d}_{j+1}^{(r)}(m,n-k) \rfloor, \tag{4.14}$$

$$\begin{aligned}\check{d}_{j+1}^{(r)}(m,n) = \widetilde{d}_{j+1}^{(r)}(m,n) - \lfloor \sum_{k\in\mathcal{Q}_j} q_{j,k} \widetilde{I}_{j+1}^{(r)}(m,n-k) \\ + \sum_{k\in\mathcal{P}_j^{(r,l)}} p_{j,k}^{(r,l)} I_j^{(l)}(m+v_{x,j}(m,2n+1), 2n+1+v_{y,j}(m,2n+1)-k) \rfloor\end{aligned} \tag{4.15}$$

where notations similar to those used in Section 4.2.2 are used and $\mathcal{Q}_j$ is the support of the second intra-image predictor for the right view, the corresponding prediction weights being denoted by $q_{j,k}$. It is worth noting that a prediction and an update as in Eqs. (4.13) and (4.14) (with the same weights) are applied to $I^{(l)}$. In addition, at the last resolution level $j = J$, instead of directly coding the approximation $I_J^{(r)}$, we predict it from the approximation subband at the last level of the left image and only code the residual subband

$e_J^{(r)}$ given by:

$$e_J^{(r)}(m,n) = I_J^{(r)}(m,n) - \lfloor \sum_{k \in \mathcal{P}_J^{(r,l)}} p_{J,k}^{(r,l)} I_J^{(l)}(m + v_{x,J}(m,n), m + v_{y,J}(m,n) - k) \rfloor. \quad (4.16)$$

Let the coefficients $q_{j,k}$ and $p_{j,k}^{(r,l)}$ (resp. $p_{J,k}^{(r,l)}$) be optimized so as to minimize the variance of $\check{d}_{j+1}^{(r)}$ (resp. $e_J^{(r)}$) at each resolution level $j < J$ (resp. at the coarsest resolution level $J$). Note that in Eq. (4.16), only an inter-prediction step is used. However, an hybrid intra-inter prediction step can be also used showing thus a great flexibility of the proposed structure.

An interesting property of the proposed decomposition is the following: in the ideal situation corresponding to $I^{(l)} = I^{(r)}$, the multiresolution representation of $I^{(r)}$ reduces to zero under some constraints that we are going to define in the following. Indeed, in the ideal case when $I^{(l)} = I^{(r)}$, the disparity vectors are zero: $(v_x, v_y) = (0,0)$. Therefore, Eq. (4.15) becomes:

$$\begin{aligned}
\check{d}_{j+1}^{(r)}(m,n) =& \widetilde{d}_{j+1}^{(r)}(m,n) - \lfloor \sum_{k \in \mathcal{Q}_j} q_{j,k} \widetilde{I}_{j+1}^{(r)}(m, n-k) + \sum_{k \in \mathcal{P}_j^{(r,l)}} p_{j,k}^{(r,l)} I_j^{(l)}(m, 2n+1-k) \rfloor \\
=& I_j^{(r)}(m, 2n+1) - \lfloor \sum_{k \in \mathcal{P}_j^{(r)}} p_{j,k}^{(r)} I_j^{(r)}(m, 2n-2k) \rfloor \\
& - \lfloor \sum_{k \in \mathcal{Q}_j} q_{j,k} \widetilde{I}_{j+1}^{(r)}(m, n-k) + \sum_{k \in \mathcal{P}_j^{(r,l)}} p_{j,k}^{(r,l)} I_j^{(l)}(m, 2n+1-k) \rfloor. \quad (4.17)
\end{aligned}$$

It is worth pointing out that the coefficients $q_{j,k}$ and $p_{j,k}^{(r,l)}$ are optimized, at each decomposition level, by solving the Yule-Walker equations, the rounding operator being omitted. Thus, the detail coefficients $\check{d}_{j+1}^{(r)}(m,n)$ can be viewed as the error involved in the prediction of $\widetilde{d}_{j+1}^{(r)}(m,n)$ by the signal $p(m,n) = \sum_{k \in \mathcal{Q}_j} q_{j,k} \widetilde{I}_{j+1}^{(r)}(m, n-k) + \sum_{k \in \mathcal{P}_j^{(r,l)}} p_{j,k}^{(r,l)} I_j^{(l)}(m, 2n+1-k)$. In this way, we can ensure that the detail coefficients of the right image $\check{d}_{j+1}^{(r)}(m,n)$ are zero if the prediction signal $p(m,n)$ is a linear combination of (at least) the same samples as those used by the reference signal $\widetilde{d}_{j+1}^{(r)}(m,n)$ to be predicted. This can be guaranteed provided that the support of the hybrid predictor $\mathcal{P}_j^{(r,l)}$ satisfies the following conditions:

**(i)** The first term $I_j^{(r)}(m, 2n+1)$ in the expression of the signal $\widetilde{d}_{j+1}^{(r)}(m,n)$ can be found in the expression of the prediction signal $p(m,n)$ if $0 \in \mathcal{P}_j^{(r,l)}$.

**(ii)** The second term $(I_j^{(r)}(m, 2n-2k))_{k\in\mathcal{P}_j^{(r)}}$ in the expression of the signal $\widetilde{d}_{j+1}^{(r)}(m,n)$ can be found in the expression of the prediction signal $p(m,n)$ if $\{2k{+}1, k \in \mathcal{P}_j^{(r)}\} \subset \mathcal{P}_j^{(r,l)}$.

When these two conditions are satisfied and $I_j^{(l)} = I_j^{(r)}$, the decomposition of $I_j^{(r)}$ first provides a detail subband $\check{d}_{j+1}^{(r)}(m,n)$ which is equal to zero and an approximation subband $\widetilde{I}_{j+1}^{(r)}$ which is equal to that of $I_j^{(l)}$: $\widetilde{I}_{j+1}^{(r)}(m,n) = \widetilde{I}_{j+1}^{(l)}(m,n)$. Then, while processing the image along the columns, the decomposition of $\widetilde{I}_{j+1}^{(r)}$ generates in the same way a detail subband which is equal to zero and an approximation subband $I_{j+1}^{(r)}$ which is equal to that of $\widetilde{I}_{j+1}^{(l)}$: $I_{j+1}^{(r)}(m,n) = I_{j+1}^{(l)}(m,n)$. Finally, the decomposition of $\check{d}_{j+1}^{(r)}$ provides two null detail subbands since $\check{d}_{j+1}^{(r)}(m,n) = 0$. Consequently, the resulting multiresolution representation of $I_j^{(r)}$ based on the new scheme allows us to generate an approximation subband which is identical to that of $I_j^{(l)}$ and three detail subbands equal to zero. Since at each resolution level the approximation subbands of $I_j^{(r)}$ and $I_j^{(l)}$ are equal, the residual sub-image $e_J^{(r)}$ (Eq. (4.16)) becomes null if $0 \in \mathcal{P}_J^{(r,l)}$. Therefore, the P-U-P decomposition satisfies the property of cancelling the values of the wavelet coefficients of the multiresolution representation of $I^{(r)}$ provided that $\{0\} \cup \{2k+1, k \in \mathcal{P}_j^{(r)}\} \subset \mathcal{P}_j^{(r,l)}$, when $j < J$, and $0 \in \mathcal{P}_J^{(r,l)}$. This is a desirable property of the considered decomposition in order to get a consistent joint representation of $I^{(l)}$ and $I^{(r)}$. In contrast, this property does not hold for VLS-I.

#### An example: VLS-II

As a supporting example, we design a scheme, which will be denoted in the following by VLS-II, by adding a prediction stage to the conventional 5/3 lifting structure. This amounts to choose $\mathcal{P}_j^{(r)} = \{-1, 0\}$, $\mathcal{U}_j^{(r)} = \{0, 1\}$, and $p_{j,-1}^{(r)} = p_{j,0}^{(r)} = \frac{1}{2}$, $u_{j,0}^{(r)} = u_{j,1}^{(r)} = \frac{1}{4}$, while the last prediction stage is performed by setting $\mathcal{Q}_j = \{-1, 0\}$ and $\mathcal{P}_j^{(r,l)} = \{-3, \ldots, 3\}$ for $j \in \{0, \ldots, J-1\}$ and $\mathcal{P}_J^{(r,l)} = \{0\}$. The coefficients $q_{j,k}$ and $p_{j,k}^{(r,l)}$ are determined by solving the Yule-Walker equations (still omitting the rounding operations) and imposing again the symmetry properties: $q_{j,-1} = q_{j,0}$ and $p_{j,k}^{(r,l)} = p_{j,-k}^{(r,l)}$.

### 4.2.4 Coding cost of prediction coefficients

The prediction coefficients involved in the proposed VLS decompositions have to be transmitted to the decoder in order to proceed to the inverse transform with perfect reconstruction of the stereo pairs. The prediction weights correspond to an amount of $o_p = 3\,L\,J$ floating point coefficients, where $L$ is the number of prediction weights in the VLS and

$J$ represents the number of resolution levels (the factor 3 stems from the fact that one horizontal prediction and two vertical predictions, one in the low-pass horizontal subband and the other in the high-pass horizontal subband, are performed). Generally, the filter coefficients can be stored on 32 bits, inducing a negligible increase of the bitrate. More precisely, for a stereo pair of size $M \times N$, the transmission cost of the prediction coefficients will increase the bitrate achieved by VLS-I and VLS-II, by $\frac{o_p}{2MN}$ bits per pixel. For example, when $M = N = 512$ and $J = 2$, the gain will be increased by 0.0007 bpp (resp. 0.0018 bpp) in the case of VLS-I (resp. VLS-II) which is a very small fraction of the overall data bitrate.

## 4.3 Theoretical analysis

In this section, we perform a theoretical analysis of the performances of VLS-I and VLS-II in terms of prediction efficiency, which is directly related to the coding efficiency [Gersho, Gray, 1993; Jayant, Noll, 1984].

### Notations

In the following, we will develop our analysis in the case of 1D signals since we have considered a separable decomposition. More precisely, let $(m, n)$ be a given pixel, we consider the pair of 1D signals defined for all $y \in \mathbb{Z}$ by:

$$\begin{cases} i_j^{(r)}(y) &= I_j^{(r)}(m, y) \\ i_j^{(l)}(y) &= I_j^{(l)}(m + v_{x,j}(m, 2n+1), y + v_{y,j}(m, 2n+1)). \end{cases} \tag{4.18}$$

We assume that, at a given resolution level $j$, these signals satisfy the following symmetric linear statistical model:

$$\begin{cases} i_j^{(r)}(y) &= \alpha_j a_j(y) + \beta_j b_j(y) \\ i_j^{(l)}(y) &= \beta_j a_j(y) + \alpha_j b_j(y) \end{cases} \tag{4.19}$$

where $(\alpha_j, \beta_j) \in \mathbb{R}^2$ such that $\alpha_j^2 + \beta_j^2 = 1$, and $a_j$ and $b_j$ are two stationary random processes which are mutually independent. For the sake of simplicity, we assume that they are zero-mean and they have the same autocorrelation function $R_j$ with $R_j(0) > 0$:

$$\mathsf{E}\left[i_j^{(r)}(y) i_j^{(r)}(y-k)\right] \triangleq \mathsf{E}\left[i_j^{(l)}(y) i_j^{(l)}(y-k)\right] = R_j(k). \tag{4.20}$$

Then, it is easy to show that:

$$\mathsf{E}\left[i_j^{(r)}(y) i_j^{(l)}(y-k)\right] = s_j R_j(k) \tag{4.21}$$

where

$$s_j \triangleq \sin(2\theta_j) \tag{4.22}$$

$$\theta_j \triangleq \arg(\alpha_j + \imath\beta_j) \quad \text{with} \quad \imath^2 = -1. \tag{4.23}$$

At this point, it is worth noticing that the spatial similarities between the samples of $i_j^{(r)}$ (or $i_j^{(l)}$) are related to the autocorrelation function $R_j$. The factor $\theta_j$ controls the cross-redundancies between the samples of $i_j^{(r)}$ and $i_j^{(l)}$.

### 4.3.1 Minimum prediction error variance of VLS-I

By considering the support and the weights of the prediction operator involved in VLS-I (still omitting the rounding operators), the detail signal $\widetilde{d}_{j+1}^{(r)}(y)$ is expressed as follows:

$$\widetilde{d}_{j+1}^{(r)}(y) = i_j^{(r)}(2y+1) - p_{j,0}^{(r)}\Big(i_j^{(r)}(2y) + i_j^{(r)}(2y+2)\Big) - p_{j,0}^{(r,l)} i_j^{(l)}(2y+1). \tag{4.24}$$

Thus, $\widetilde{d}_{j+1}^{(r)}(y)$ can be viewed as the error in the prediction of $i_j^{(r)}(2y+1)$ by the following multivariate reference signal $\mathbf{i}_j(y) \triangleq \Big(i_j^{(r)}(2y) + i_j^{(r)}(2y+2), i_j^{(l)}(2y+1)\Big)^\top$. The prediction weight vector $\mathbf{p}_j = (p_{j,0}^{(r)}, p_{j,0}^{(r,l)})^\top$ satisfies the normal equations:

$$\mathsf{E}\left[\mathbf{i}_j(y)\mathbf{i}_j(y)^\top\right]\mathbf{p}_j = \mathsf{E}\left[i_j^{(r)}(2y+1)\mathbf{i}_j(y)\right]. \tag{4.25}$$

Hence, the optimal weights can be deduced as follows:

$$\begin{cases} p_{j,0}^{(r)} = & \gamma_{1,j}(\theta_j) R_j(0) R_j(1)(s_j^2 - 1) \\ p_{j,0}^{(r,l)} = & \gamma_{1,j}(\theta_j) s_j \Big(2R_j(1)^2 - Rj(0)^2 - R_j(0)R_j(2)\Big) \end{cases} \tag{4.26}$$

where $\gamma_{1,j}(\theta_j) \triangleq \Big(2s_j^2 R_j(1)^2 - R_j(0)^2 - R_j(0)R_j(2)\Big)^{-1}$.
Consequently, the minimum value $\varepsilon_{1,j}$ of the prediction error variance achieved by VLS-I is:

$$\begin{aligned} \varepsilon_{1,j}(R_j, \theta_j) =& \mathsf{E}\left[i_j^{(r)}(2n+1)^2\right] - \mathbf{p}_j^\top \mathsf{E}\left[i_j^{(r)}(2n+1)\mathbf{i}_j(n)\right] \\ =& \gamma_{1,j}(\theta_j)\cos^2(2\theta_j) R_j(0)\Big(2R_j(1)^2 - R_j(0)^2 - R_j(0)R_j(2)\Big). \end{aligned} \tag{4.27}$$

### 4.3.2 Minimum prediction error variance of VLS-II

Considering now VLS-II (still omitting the rounding operators), the detail signal $\check{d}_{j+1}^{(r)}(y)$ is given by:

$$\begin{aligned}\check{d}_{j+1}^{(r)}(y) = i_j^{(r)}(2y+1) - \frac{1}{2}\Big(i_j^{(r)}(2y) + i_j^{(r)}(2y+2)\Big) - q_{j,0}\Big(\widetilde{i}_{j+1}^{(r)}(y) + \widetilde{i}_{j+1}^{(r)}(y+1)\Big) \\ - p_{j,0}^{(r,l)} i_j^{(l)}(2y+1) - \sum_{k=1}^{3} p_{j,k}^{(r,l)}\Big(i_j^{(l)}(2y+1-k) + i_j^{(l)}(2y+1+k)\Big)\end{aligned} \tag{4.28}$$

where, as shown by Eq. (4.14), the signal $\widetilde{i}_{j+1}^{(r)}(y)$ can be expressed as:

$$\begin{aligned}\widetilde{i}_{j+1}^{(r)}(y) &= i_j^{(r)}(2y) + \frac{1}{4}\Big(\widetilde{d}_{j+1}^{(r)}(y-1) + \widetilde{d}_{j+1}^{(r)}(y)\Big) \\ &= \frac{1}{4}\Big(3i_j^{(r)}(2y) + i_j^{(r)}(2y+1) + i_j^{(r)}(2y-1)\Big) - \frac{1}{8}\Big(i_j^{(r)}(2y-2) + i_j^{(r)}(2y+2)\Big).\end{aligned} \tag{4.29}$$

Therefore, it can be checked that:

$$\check{d}_{j+1}(y) = r_j(y) - q_{j,0}u_j(y) - p_{j,0}^{(r,l)} i_j^{(l)}(2y+1) - \sum_{k=1}^{3} p_{j,k}^{(r,l)}\Big(i_j^{(l)}(2y+1-k) + i_j^{(l)}(2y+1+k)\Big) \tag{4.30}$$

where

$$r_j(y) \stackrel{\triangle}{=} i_j^{(r)}(2y+1) - \frac{1}{2}\Big(i_j^{(r)}(2y) + i_j^{(r)}(2y+2)\Big), \tag{4.31}$$

$$\begin{aligned}u_j(y) \stackrel{\triangle}{=} &\frac{1}{2} i_j^{(r)}(2y+1) + \frac{5}{8}\Big(i_j^{(r)}(2y) + i_j^{(r)}(2y+2)\Big) + \frac{1}{4}\Big(i_j^{(r)}(2y-1) + i_j^{(r)}(2y+3)\Big) \\ &- \frac{1}{8}\Big(i_j^{(r)}(2y-2) + i_j^{(r)}(2y+4)\Big).\end{aligned} \tag{4.32}$$

From Eq. (4.30), $\check{d}_{j+1}^{(r)}(y)$ can be viewed as the error in the prediction of $r_j(y)$ by the reference samples grouped into the vector $\tilde{\mathbf{r}}_j(y)$ given by:

$$\begin{aligned}\tilde{\mathbf{r}}_j(y) \stackrel{\triangle}{=} &\Big(u_j(y), i_j^{(l)}(2y+1), i_j^{(l)}(2y) + i_j^{(l)}(2y+2), i_j^{(l)}(2y-1) + i_j^{(l)}(2y+3), \\ &i_j^{(l)}(2y-2) + i_j^{(l)}(2y+4)\Big)^{\top}.\end{aligned} \tag{4.33}$$

The vector of prediction weights $\mathbf{p}_j = \Big(q_{j,0}, p_{j,0}^{(r,l)}, p_{j,1}^{(r,l)}, p_{j,2}^{(r,l)}, p_{j,3}^{(r,l)}\Big)^{\top}$ is found by minimizing the variance of $\check{d}_{j+1}^{(r)}(n)$. Consequently, the following set of normal equations $\mathbf{\Gamma}_j \mathbf{p}_j = \mathbf{c}_j$ must be solved where $\mathbf{\Gamma}_j = \mathsf{E}\left[\tilde{\mathbf{r}}_j(y)\tilde{\mathbf{r}}_j(y)^{\top}\right]$ and $\mathbf{c}_j = \mathsf{E}\left[r_j(y)\tilde{\mathbf{r}}_j(y)\right]$. Once the auto-correlation matrix $\mathbf{\Gamma}_j$ and the cross-correlation vector $\mathbf{c}_j$ are determined, the

optimal weights are obtained as follows:

$$q_{j,0} = -4\gamma_{2,j}\Big(R_j(0) - 4R_j(1) + 4R_j(3) - R_j(4)\Big), \tag{4.34}$$

$$p_{j,0}^{(r,l)} = \gamma_{2,j}s_j\Big(40R_j(0) + 48R_j(1) + 31R_j(2) + 20R_j(3) - 8R_j(4) - 4R_j(5) + R_j(6)\Big), \tag{4.35}$$

$$p_{j,1}^{(r,l)} = -\gamma_{2,j}s_j\Big(33R_j(0) + 76R_j(1) + 31R_j(2) - 8R_j(3) - R_j(4) - 4R_j(5) + R_j(6)\Big)/2, \tag{4.36}$$

$$p_{j,2}^{(r,l)} = \gamma_{2,j}s_j\Big(R_j(0) - 4R_j(1) + 4R_j(3) - R_j(4)\Big), \tag{4.37}$$

$$p_{j,3}^{(r,l)} = -\gamma_{2,j}s_j\Big(R_j(0) - 4R_j(1) + 4R_j(3) - R_j(4)\Big)/2 \tag{4.38}$$

where $\gamma_{2,j} = \Big(38R_j(0) + 56R_j(1) + 31R_j(2) + 12R_j(3) - 6R_j(4) - 4R_j(5) + R_j(6)\Big)^{-1}$. Finally, the minimal value of the variance $\varepsilon_{2,j}$ of the prediction error generated by VLS-II is:

$$\begin{aligned}
\varepsilon_{2,j}(R_j,\theta_j) =& \mathsf{E}\left[r_j^2(y)\right] - \mathbf{p}_j^\top \mathsf{E}\left[r_j(y)\tilde{\mathbf{r}}_j(y)\right] \\
=& \frac{1}{2}\gamma_{2,j}\cos^2(2\theta_j)\Big(113R_j(0)^2 - 240R_j(1)^2 + 31R_j(2)^2 - 16R_j(3)^2 - R_j(4)^2 \\
& - 4R_j(1)R_j(6) + 28R_j(0)R_j(3) - 16R_j(0)R_j(4) + 131R_j(0)R_j(2) \\
& - 4R_j(2)R_j(5) + 3R_j(0)R_j(6) - 16R_j(1)R_j(3) + 16R_j(1)R_j(4) \\
& - 68R_j(1)R_j(2) + 16R_j(1)R_j(5) + 12R_j(2)R_j(3) + R_j(2)R_j(6) \\
& + 8R_j(3)R_j(4) + 24R_j(0)R_j(1) - 6R_j(2)R_j(4) - 12R_j(0)R_j(5)\Big).
\end{aligned} \tag{4.39}$$

### 4.3.3 Discussion

It is interesting to note that, unlike $\varepsilon_{1,j}$, $\varepsilon_{2,j}$ has a separable form in $R_j$ and $\theta_j$. Furthermore, it should be noticed that this expression is not dependent on a particular form of the autocorrelation function $R_j$, so it can be applied to any second-order stationary process. In order to emphasize the advantages of VLS-I and VLS-II, we will consider a simple multivariate random process model driven by two autoregressive processes of order 1, $a_j(y)$ and $b_j(y)$ in Eq. (4.19). In this particular case, the autocorrelation function is given by:

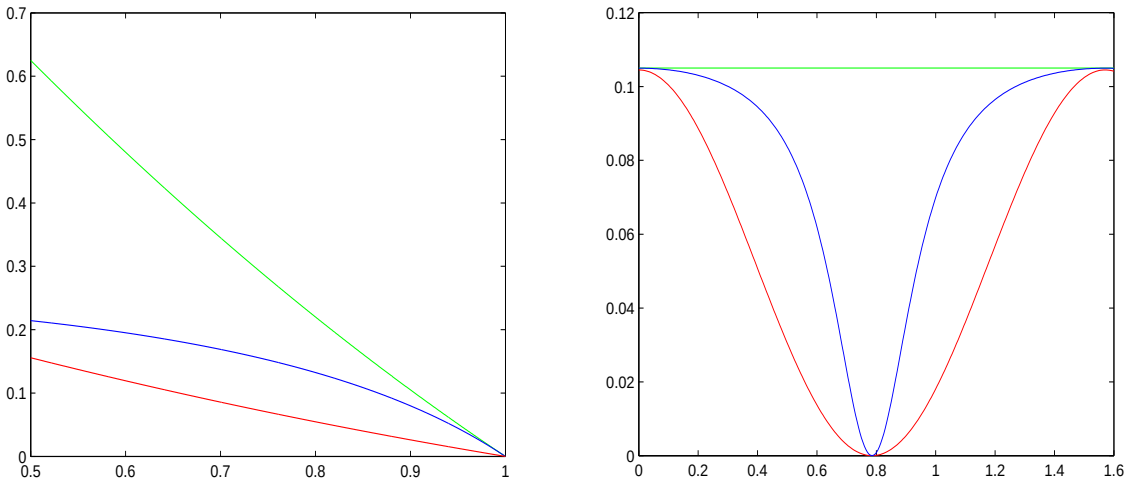
$$\forall k \in \mathbb{Z}, \quad R_j(k) = \sigma_j^2\rho_j^{|k|} \tag{4.40}$$

where $\rho_j \in [-1, 1]$ is the correlation factor. Therefore, the variances $\varepsilon_{1,j}$ and $\varepsilon_{2,j}$ of the prediction errors reduce to:

$$\varepsilon_{1,j}(\rho_j, \theta_j) = \sigma_j^2 \tilde{\gamma}_{1,j}(\theta_j) \cos^2(2\theta_j)(\rho_j^2 - 1), \tag{4.41}$$

$$\varepsilon_{2,j}(\rho_j, \theta_j) = \frac{1}{2}\sigma_j^2 \tilde{\gamma}_{2,j} \cos^2(2\theta_j)(1 - \rho_j)(3\rho_j^4 - 16\rho_j^3 + 4\rho_j^2 + 24\rho_j + 113) \tag{4.42}$$

where $\tilde{\gamma}_{1,j}(\theta_j) = (2s_j^2\rho_j^2 - \rho_j^2 - 1)^{-1}$ and $\tilde{\gamma}_{2,j} = (\rho_j^5 - 5\rho_j^4 - \rho_j^3 + 13\rho_j^2 + 18\rho_j + 38)^{-1}$. Fig. 4.4(a) shows the variations of $\mathsf{E}\left[r_j(y)^2\right]$, $\varepsilon_{1,j}$ and $\varepsilon_{2,j}$ with respect to $\rho_j$ for a given value of $\theta_j$.




(a): Variations w.r.t $\rho_j$ when $\sigma_j = 1$ and $\theta_j = \frac{\pi}{6}$. (b): Variations w.r.t $\theta_j$ when $\sigma_j = 1$ and $\rho_j = 0.9$.
Figure 4.4: Prediction efficiency: $\mathsf{E}[r_j^2(y)]$ (in green), $\varepsilon_{1,j}(\rho_j, \theta_j)$ (in blue), $\varepsilon_{2,j}(\rho_j, \theta_j)$ (in red).

We recall that $\mathsf{E}\left[r_j(y)^2\right]$ represents the variance of the prediction error generated by the 5/3 transform as indicated by Eq. (4.31). Thus, by taking into account the spatial redundancies (controlled by $\rho_j$), the variance $\varepsilon_{1,j}$ is lower than $\mathsf{E}\left[r_j(y)^2\right]$. Lower values of the prediction error variance $\varepsilon_{2,j}$ are further achieved by the VLS-II transform for any value of $\rho_j$. We are also interested in comparing the variations of these three prediction errors with respect to $\theta_j$ for a given value of $\rho_j$ as depicted by Fig. 4.4(b). It can be noticed that VLS-II gives also the best results by taking into account the inter-image redundancies (controlled by $\theta_j$). This study has clearly shown the benefit that can be drawn from the use of the VLS-II structure in simultaneously exploiting the spatial and the cross-view redundancies.

## 4.4 Experimental results

Simulations are performed on 6 image pairs of size $512 \times 512$ which have been extracted from a SPOT5 scene. The full scene, which corresponds to an urban zone, is shown in Fig. 4.5 and the six image pairs are represented in white squares.

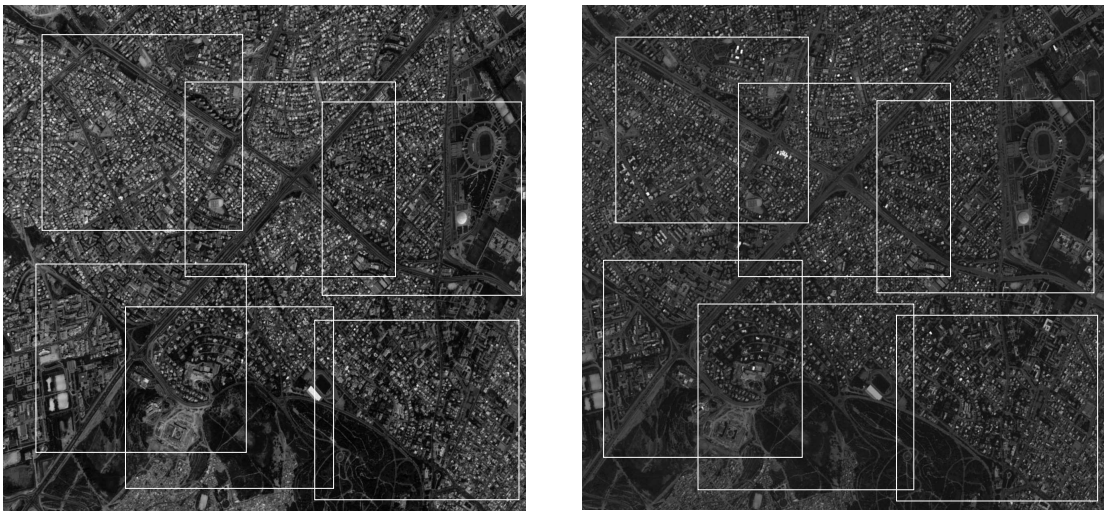


Figure 4.5: Original SI pair "spot5": the left and right images.

All the test images are coded at 8 bpp. We have also used 4 pairs of natural stereo images ("fruit", "pentagon", "shrub" and "birch") downloaded from [2] and [3]. It should be noted that some stereo images have significant illumination variations between the views. For this reason, DC is performed by applying to the original SI the reversible remapping technique based on sorting permutations introduced in [Arnavut, Narumalani, 1996]. This preprocessing step is often used to improve the coding efficiency of pairs of images [Benazza-Benyahia, Pesquet, 2002; U. Fecker, 2007]. The disparity map is computed using a block-matching technique with a $8 \times 8$ block size and a search area that depends on the acquisition of the stereo pair (+50 pixels in the horizontal direction and $\pm 2$ in the vertical direction for SPOT5 stereo images, and +30 pixels in the horizontal direction and $\pm 4$ in the vertical direction for natural stereo images). The SSD is the retained matching criterion. The resulting disparity vectors are losslessly encoded using a median prediction and DPCM with arithmetic encoding. In order to show the benefit of the joint coding by

[2] http://vasc.ri.cmu.edu/idb/html/stereo/index.html
[3] http://vasc.ri.cmu.edu/idb/html/jisct/

VLS, we compare VLS-I and VLS-II decompositions carried out over two resolution levels with some conventional SI wavelet-based coding methods:

• The first one is the baseline coder which consists of coding the SI pairs by applying a wavelet transform to the left image $I^{(l)}$ and the DC-residual $I^{(e)}$ [Boulgouris, Strintzis, 2002]. In the following, this method will be designated by scheme B.

• The second one is the subspace projection approach in the wavelet domain proposed by Jiang *et al.* [Q. Jiang, Hayes, 1999]. We recall that this method is described in Section 3.3.3 and it will be designated by SPT-WT.

• We have also tested a version of JPEG2000 (Annex I of part II) dedicated to multi-component images. This method has also been described in Section 3.3.3 and it will be designated in the following by scheme C.

The compression measure is given by the final bitrates of the multiresolution representations. Let us denote by $R^{(v)}$, $R^{(l)}$, $R^{(r)}$ and $R^{(e)}$, respectively, the bitrate of the disparity vectors $\mathbf{v}$ and, the images $I^{(l)}$, $I^{(r)}$ and $I^{(e)}$. For the methods based on the coding of the residual image, we have computed the following average bitrate:

$$R_{\text{av}} = \frac{R^{(l)} + R^{(e)} + R^{(v)}}{2} \tag{4.43}$$

whereas the average bitrate for the proposed decompositions is given by:

$$R_{\text{av}} = \frac{R^{(l)} + R^{(r)} + R^{(v)}}{2}. \tag{4.44}$$

It can be noticed that the average coding cost $R^{(v)}$ of the disparity vectors is equal to about 0.07 bpp. Table 6.3 shows the final bitrates obtained in a lossless context by applying the JPEG2000 codec used only as an entropy codec.

Our simulations indicate that VLS-I results in an average gain of about 0.1 bpp over conventional methods. If we now compare the performance of VLS-II to those provided by VLS-I, our experiments show that VLS-II leads to a further improvement of about 0.1 bpp.

We have also tested the performance of our methods when applied as a lossy codec. In this case, the improved VLS are also compared in terms of Peak-Signal-to-Noise Ratio (PSNR) given by:

$$\text{PSNR} = 10 \log_{10} \Big( \frac{255^2}{(\text{MSE}^{(l)} + \text{MSE}^{(r)})/2} \Big), \tag{4.45}$$

where $\text{MSE}^{(l)}$ and $\text{MSE}^{(r)}$ correspond respectively to the mean squared error of the left and right images reconstructed at the rates $R^{(l)}$ and $R^{(r)}$. We also used the SSIM quality

Table 4.1: Performance of SI wavelet-based lossless codecs in terms of average bitrate (in bpp) using JPEG2000.

| Transform | scheme B | SPT-WT | scheme C | VLS-I | VLS-II |
|---|---|---|---|---|---|
| spot5-1 | 3.63 | 3.59 | 3.58 | **3.49** | **3.35** |
| spot5-2 | 3.85 | 3.80 | 3.78 | **3.67** | **3.53** |
| spot5-3 | 4.27 | 4.21 | 4.24 | **4.03** | **3.93** |
| spot5-4 | 4.22 | 4.18 | 4.21 | **4.05** | **3.92** |
| spot5-5 | 3.91 | 3.87 | 3.89 | **3.80** | **3.73** |
| spot5-6 | 3.89 | 3.84 | 3.81 | **3.73** | **3.63** |
| fruit | 4.05 | 3.99 | 3.97 | **3.78** | **3.72** |
| shrub | 3.73 | 3.69 | 3.69 | **3.81** | **3.63** |
| birch | 4.52 | 4.49 | 4.47 | **4.44** | **4.37** |
| pentagon | 5.37 | 5.32 | 5.20 | **5.12** | **5.04** |
| Average | 4.14 | 4.09 | 4.08 | **3.99** | **3.88** |

metrics, which is based on models of visual perceptions, to evaluate the reconstruction quality of each compression method [Wang *et al.*, 2004; Rouse, Hemami, 2008]. We are first interested in studying the evolution of the PSNR versus the bitrates achieved by VLS-I, VLS-II, the conventional schemes B and C, and the independent SI coder. In order to decode the SI pair, two alternatives can be envisaged:

• The most basic one consists of firstly decoding exactly the reference image. Then, the target image is decoded by using the original left image and the disparity vectors.

• However, in order to minimize the latency on the decoder side and to achieve the reconstruction of both images for a given bandwidth, we choose to simultaneously decode the SI pair. In other words, the decoding of the target image $I^{(r)}$ at a specified bitrate $R^{(r)}$ is achieved by using the decoded left image $\overline{I}^{(l)}$ at a bitrate $R^{(l)}$ without waiting for the final decoding of the reference image.

More precisely, for the coding scheme B, the reconstructed target image $\overline{I}^{(r)}$ is obtained by using the reconstructed left image $\overline{I}^{(l)}$ and residual image $\overline{I}^{(e)}$ decoded respectively at

$R^{(l)}$ and $R^{(e)}$:

$$\overline{I}^{(r)}(m,n) = \overline{I}^{(e)}(m,n) + \overline{I}^{(l)}(m+v_x, n+v_y). \tag{4.46}$$

Then, by comparing the original images $I^{(l)}$ and $I^{(r)}$ with the reconstructed ones $\overline{I}^{(l)}$ and $\overline{I}^{(r)}$, we can evaluate the quality of reconstruction of the SI pair at the average bitrate defined by Eq. (4.43).

Concerning the proposed methods, the reference image is decoded at different bitrates in the same way as in the previously mentioned methods. Then, the right image is decoded at some bitrate $R^{(r)}$ by using the reference image decoded at a bitrate $R^{(l)}$. Thus, we still evaluate the quality of reconstruction of the SI pair at the average bitrate given by Eq. (4.44). Figures 4.6 and 4.7 show the scalability in quality of the reconstruction procedure by providing the variations of the PSNR versus the bitrate for the SIs pair "shrub" and "spot5-5" using JPEG2000 as an entropy codec. These plots show that schemes B and C (based on the coding of the residual image) outperform the independent decomposition scheme especially at low bitrates. VLS-I performs more poorly than these schemes at low bitrates but beyond some bitrate it is more performant than scheme B. Finally, VLS-II outperforms VLS-I and improves the PSNR by at least 0.5 dB at high bitrate and the difference becomes much more important at low bitrates.

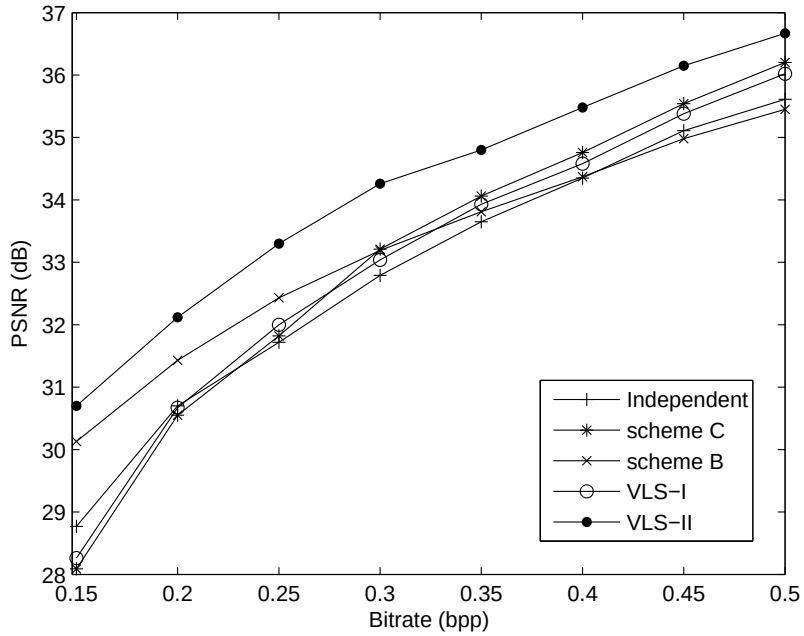



Figure 4.6: PSNR (in dB) versus the bitrate (bpp) after JPEG 2000 encoding for the SI pair "shrub": for independent scheme, schemes B and C, a 5/3 integer transform was applied.

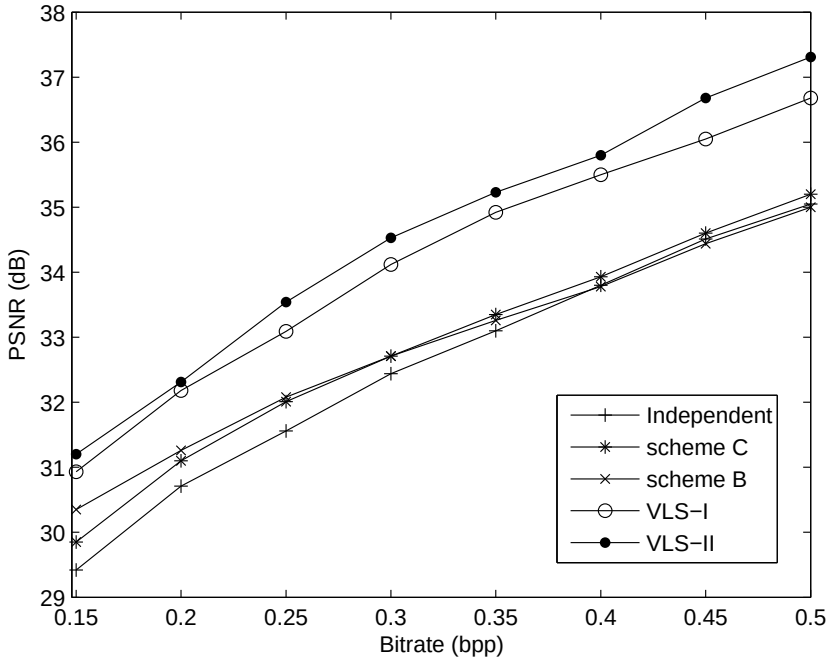



Figure 4.7: PSNR (in dB) versus the bitrate (bpp) after JPEG 2000 encoding for the SI pair "spot5-5": for independent scheme, schemes B and C, a 5/3 integer transform was applied.

Figure 4.8 shows the performance of our structures at high bitrate for the SIs pair "spot5-5". It can be noticed that the proposed schemes remain more performant than the conventional ones. Since the human eye cannot distinguish the subjective image quality at middle and high bitrate, we will focus in the rest of this thesis on the low bitrate when simulations are performed in a lossy coding context.

Figures 4.9 and 4.10 display a zoom applied on the reconstructed target image of the SI pairs "pentagon" and "spot5-5" for scheme B and VLS-II.

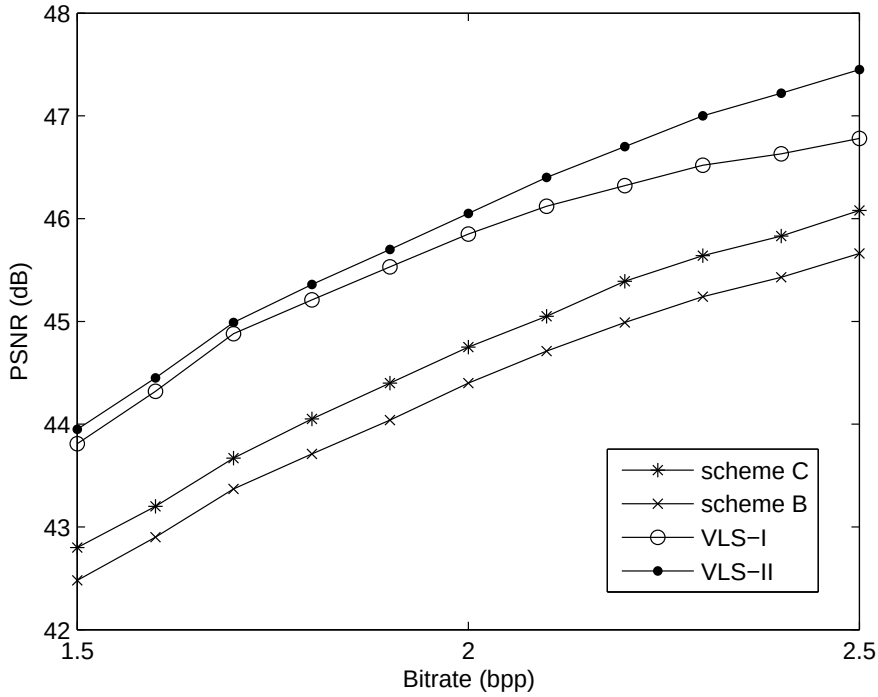
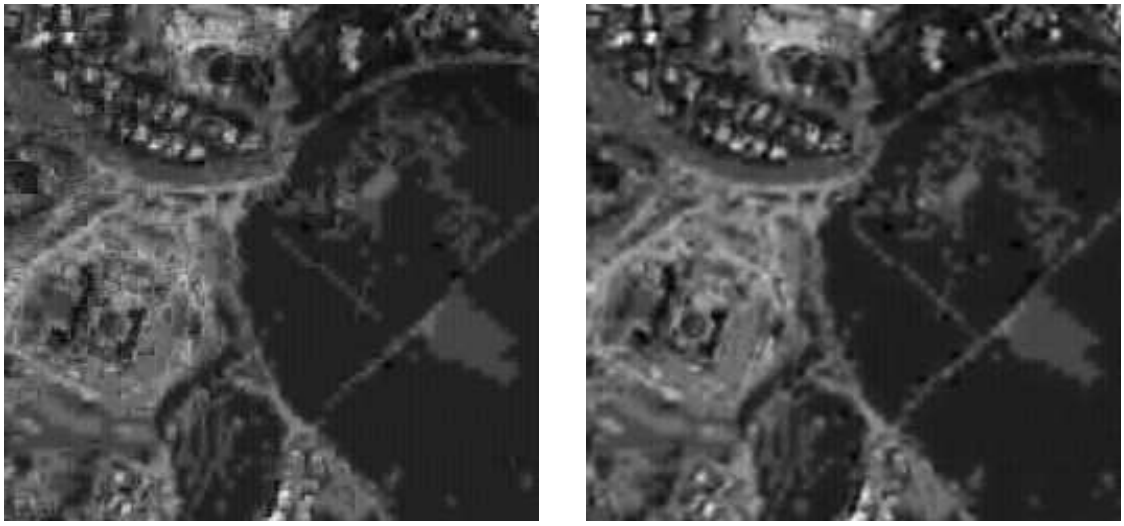



Figure 4.8: PSNR (in dB) versus the bitrate (bpp) after JPEG 2000 encoding for the SI pair "spot5-5": for schemes B and C, a 5/3 integer transform was applied: illustration at high bitrate

(a): PSNR=25.10 dB, SSIM=0.647 (b): PSNR=25.99 dB, SSIM=0.676

Figure 4.9: Reconstructed target image $I^{(r)}$ of the "pentagon" SI at 0.15 bpp: (a) scheme B (using the 5/3 integer transform); (b) VLS-II.



(a): PSNR=30.78 dB, SSIM=0.837 (b): PSNR=33.04 dB, SSIM=0.870

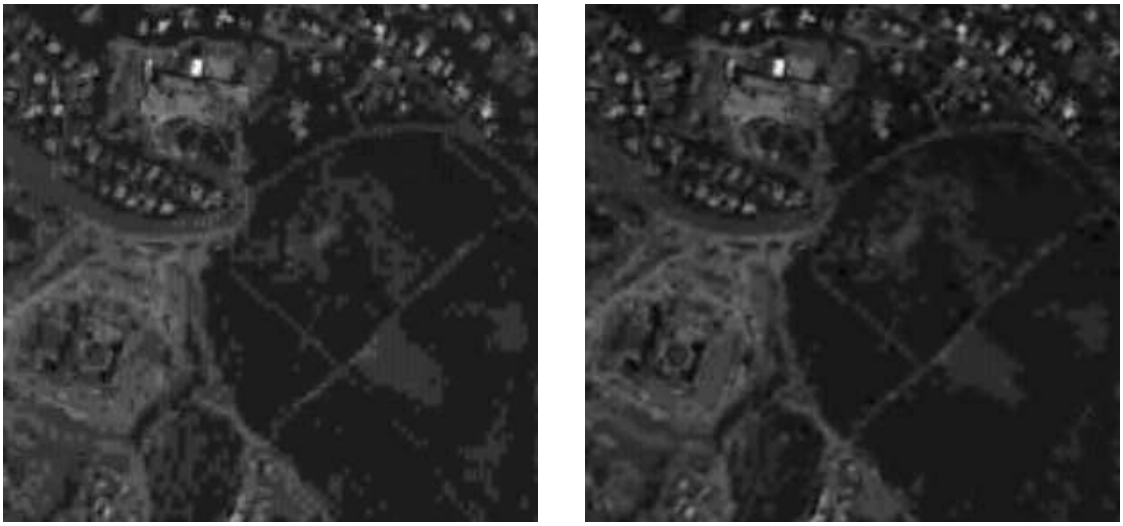
Figure 4.10: Reconstructed target image $I^{(r)}$ of the "spot5-5" SI at 0.25 bpp: (a) scheme B (using the 5/3 integer transform); (b) VLS-II.

We notice that the coding of the residual image leads to blocking artifacts at low bitrates. This problem is significantly reduced by using a VLS based decomposition, which jointly takes into account the intra and inter predictions.

Figures 4.11 and 4.12 display the reconstructed target image of the SI pairs "shrub" and "spot5-5" for VLS-I and VLS-II. Fig. 4.13 illustrates the reconstructed right image of the "shrub" pair at the decoder side corresponding to a progressive reconstruction. The quality of these images is compared both in terms of PSNR and SSIM. The difference in PSNR (resp. SSIM) between VLS-I and VLS-II ranges from 0.5 dB to 1.5 dB (resp. 0.03 to 0.1).

(a): PSNR=30.28 dB, SSIM=0.770 (b): PSNR=30.75 dB, SSIM=0.800

Figure 4.11: Reconstructed target image $I^{(r)}$ of the "shrub" SI at 0.2 bpp: (a) VLS-I; (b) VLS-II.



(a): PSNR=32.55 dB, SSIM=0.860 (b): PSNR=33.04 dB, SSIM=0.870

Figure 4.12: Reconstructed target image $I^{(r)}$ of the "spot5-5" SI at 0.25 bpp: (a) VLS-I; (b) VLS-II.

Bitrate=0.15 bpp, PSNR=27.31 dB, SSIM=0.656

Bitrate=0.15 bpp, PSNR=28.81 dB, SSIM=0.742

Bitrate=0.2 bpp, PSNR=30.28 dB, SSIM=0.770

Bitrate=0.2 bpp, PSNR=30.75 dB, SSIM=0.800

Figure 4.13: Reconstructed target image $I^{(r)}$ of the "shrub" SI at different bitrates: (a) VLS-I; (b) VLS-II.

Additional results in terms of scalability and quality of reconstruction are given when using the 9/7 wavelet transform for scheme B, scheme C and independent scheme. More precisely, figures 4.14 and 4.15 (resp. figures 4.16 and 4.17) show the evolution of the PSNR versus the bitrate when the integer version (resp. non integer version) of the 9/7 transform

was applied.

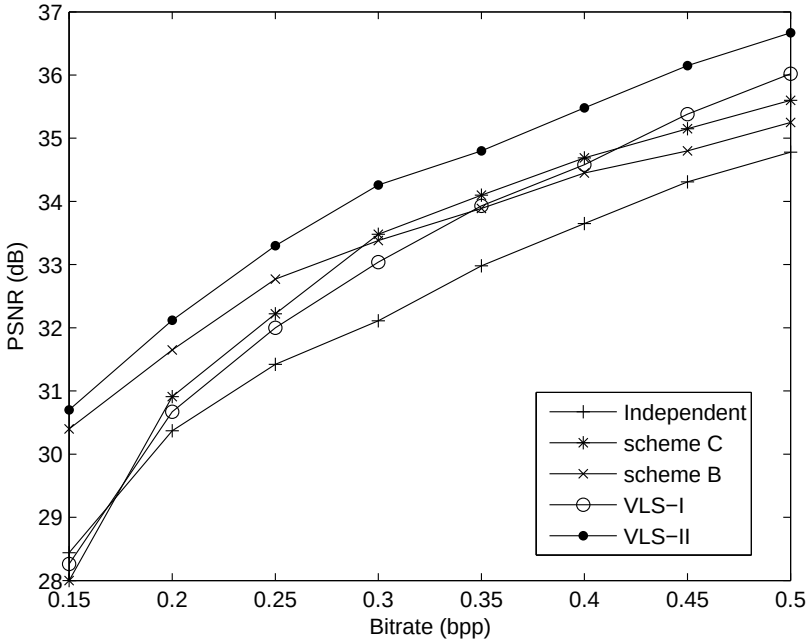



Figure 4.14: PSNR (in dB) versus the bitrate (bpp) after JPEG 2000 encoding for the SI pair "shrub": for independent scheme, schemes B and C, a 9/7 integer transform was applied.

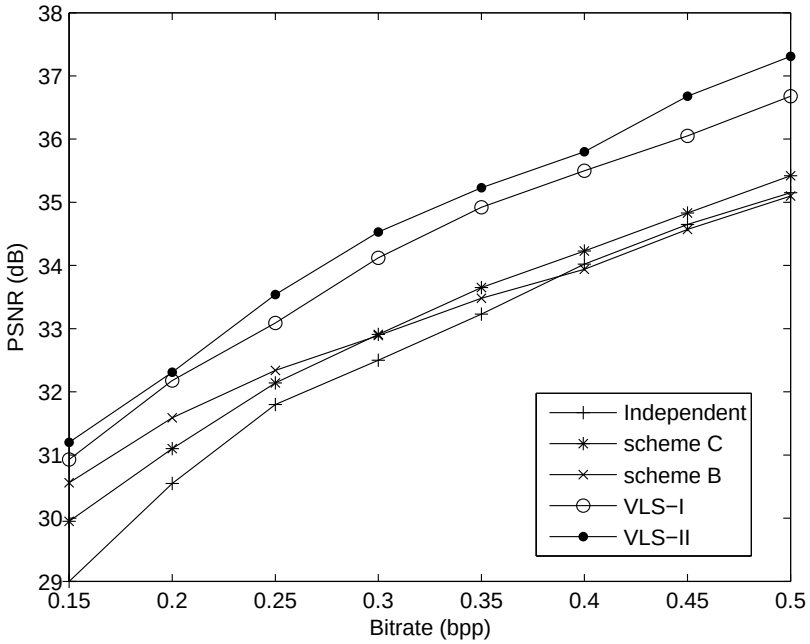



Figure 4.15: PSNR (in dB) versus the bitrate (bpp) after JPEG 2000 encoding for the SI pair "spot5-5": for independent scheme, schemes B and C, a 9/7 integer transform was applied.

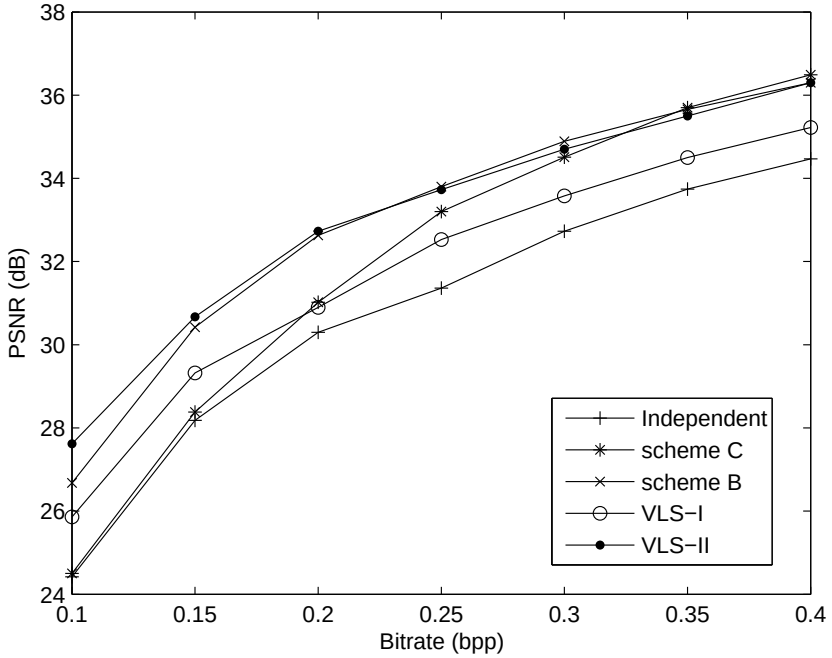



Figure 4.16: PSNR (in dB) versus the bitrate (bpp) after JPEG 2000 encoding for the SI pair "shrub": for independent scheme, schemes B and C, a 9/7 non integer transform was applied.

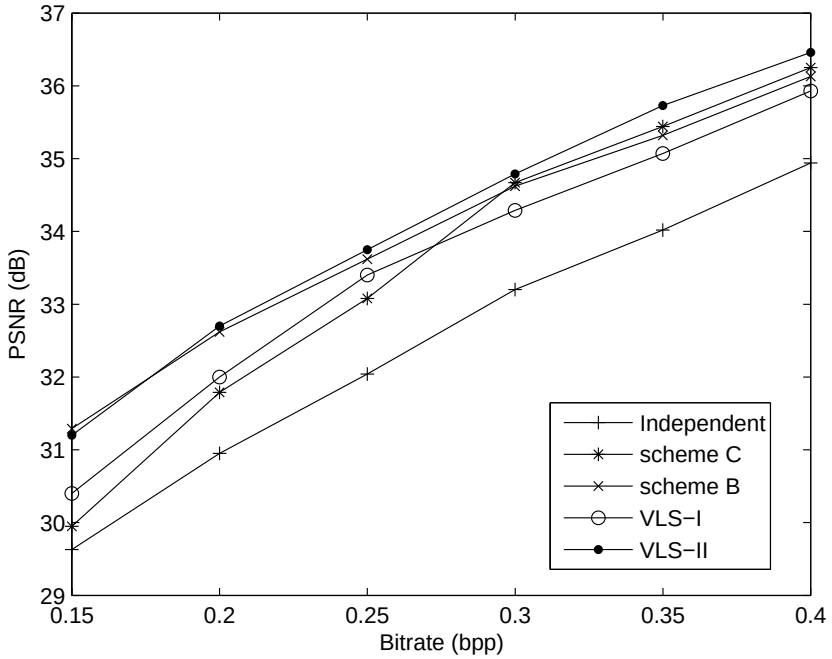
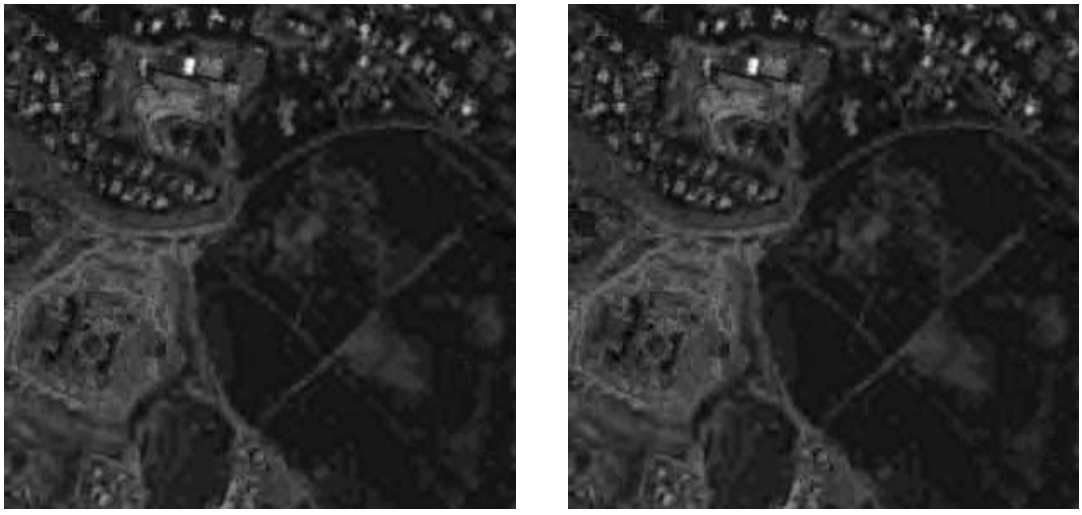


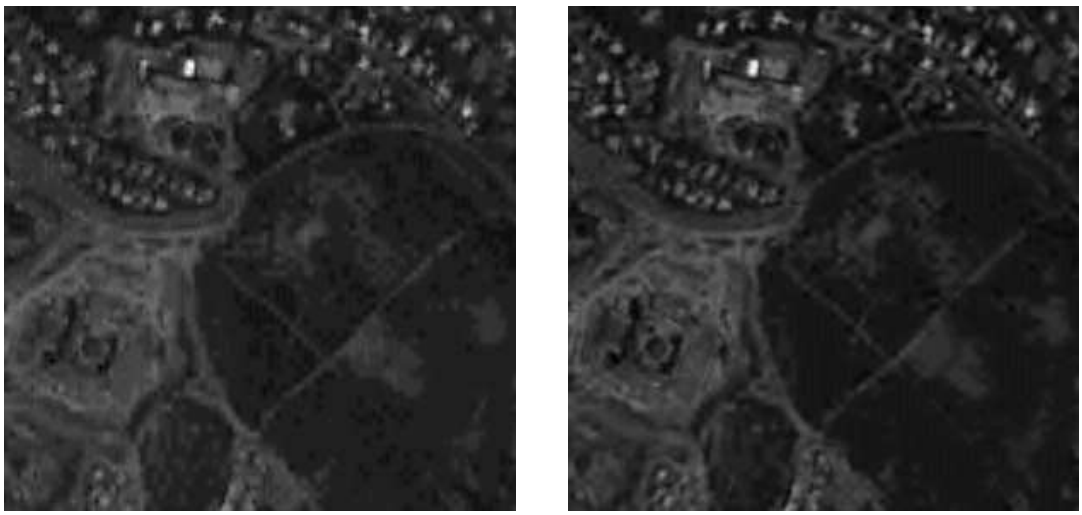

Figure 4.17: PSNR (in dB) versus the bitrate (bpp) after JPEG 2000 encoding for the SI pair "spot5-5": for independent scheme, schemes B and C, a 9/7 non integer transform was applied.

Figure 4.18 displays an example of the reconstructed target image of the SI pairs "spot5-5" for scheme B (using the non integer version of the 9/7 transform), scheme C (using the non integer version of the 9/7 transform), VLS-I and VLS-II.



(a): PSNR=32.44 dB, SSIM=0.849

(b): PSNR=32.02 dB, SSIM=0.843



(c): PSNR=32.78 dB, SSIM=0.856

(d): PSNR=33.31 dB, SSIM=0.874

Figure 4.18: The reconstructed target image $I^{(r)}$ of the "spot5-5" SI at 0.25 bpp using: (a) scheme B (using the non-integer 9/7 transform) (b) scheme C (using the non-integer 9/7 transform) (c) VLS-I (d) VLS-II.

It can be noticed that VLS-II still outperforms all the conventional methods. Note that even though the PSNR illustrated in these curves is the *average* PSNR (as defined in Eq. 4.45), the gain on the quality of the target image $I^{(r)}$ may be significantly bigger, especially when using VLS-II.

Finally, we propose to compare the different schemes in terms of execution time. Table 6.1 presents the encoding and decoding time of a Matlab implementation of the tested methods, at 0.2 bpp, for two stereo images of size $512 \times 512$.

Table 4.2: Execution time of the proposed methods (in seconds).

| | independent scheme | | scheme B | | VLS-I | | VLS-II | |
|---|---|---|---|---|---|---|---|---|
| | encoding | decoding | encoding | decoding | encoding | decoding | encoding | decoding |
| spot5-6 | 0.57 | 0.15 | 0.83 | 0.49 | 2.29 | 1.20 | 2.44 | 1.46 |
| fruit | 0.55 | 0.15 | 0.84 | 0.50 | 2.31 | 1.25 | 2.58 | 1.48 |

Simulations are carried out by using a computer Intel Core 2 (3 GHz). Indeed, we can note that the proposed methods VLS-I and VLS-II respectively require an additional average time of about 1.1 and 1.3 seconds compared to the residual image coding based method (scheme B). However, this difference in execution time is compensated by the good compression performance of the proposed VLS.

## 4.5 Conclusion

In this chapter, we have presented a new technique for lossy-to-lossless compression of stereo image pairs. In order to take advantage of the correlations between the two images, we have proposed two schemes based on the vector lifting concept. Unlike conventional methods which generate a residual image to encode the stereo pair, the proposed schemes rely on a joint multiscale decomposition directly applied to the left and the right views. They exploit the intra and inter-image redundancies by using the estimated disparity map between the two views. Furthermore, the proposed decompositions guarantee the perfect reconstruction of the original stereo images. It is worth pointing out that these decompositions are also adapted to the content of the images. A theoretical analysis in terms of prediction error variance was conducted in order to show the benefits of the underlying VLS structure. Experimental results, carried out on a set of remote sensing and natural stereoscopic images,

have indicated the good performance of the VLS over the conventional approaches in terms of bitrate and quality of reconstruction.

# Chapter 5

# Dense disparity estimation for stereo image coding

## 5.1 Introduction

In the previous chapter, we have mainly focused on the transform coding schemes used to encode the reference and the target images of a stereo pair. However, it is shown that all the proposed approaches rely on the disparity estimation/compensation process to exploit the inter-view redundancies. Therefore, the overall encoding performance can be improved by using an efficient disparity estimation technique instead of a block-matching method as it has been considered in the previous work. Due to the limitations of a Block-based Disparity Estimation (BDE) technique (discussed in Section 2.4.1), we propose in this chapter to replace this technique by employing a recent Dense Disparity Estimation (DDE) algorithm.

Indeed, since the pioneering work by Lukacs [Lukacs, 1986], the most widely used approaches in stereo image coding have been based on the fixed size block-matching technique. However, as mentioned in Chapter 2, the limitation of such techniques is that they fail around depth discontinuities and in textureless regions. Further improvements can be achieved by employing variable block size [Woo, Ortega, 2000] and resorting to rate-distortion based algorithms similar to those employed in the framework of motion-compensated video coding [Tzovaras, Strintzis, 1998; Deever, Hemami, 1998]. However, the classical block-based techniques do not always guarantee a consistent disparity map which may affect the disparity compensation process.

Dense pixel-based approaches attempt to overcome this drawback by assigning a disparity value to each point visible in the stereo pair. Recently, the dense disparity estimation problem has been formulated as a convex programming problem within a global variational framework [Miled *et al.*, 2009a]. Intensive experiments have shown that variational-based disparity estimation methods are among the most powerful techniques, meanwhile preserving the depth discontinuities. Indeed, a quantitative comparison with results from other stereo algorithms available at the Middlebury website[1] shows that this approach is comparable with state-of-the-art methods, such as graph cuts [Boykov *et al.*, 2001; Kolmogorov, Zabih, 2001] and dynamic programming based methods [Veksler, 2005; Kim *et al.*, 2005; Sun, 2002]. This naturally motivates our choice to integrate this global convex variational framework, along with the disparity estimation, within a generic joint stereo image coding scheme.

However, the obvious price for generating an accurate disparity field is its very expensive cost in terms of bitrate, compared to a BDE method. To reduce the bitrate needed to transmit/store the resulting dense disparity field, we propose to perform a quadtree decomposition, which was found to be a very effective method for encoding a dense disparity field [Tzovaras *et al.*, 1998; Morvan *et al.*, 2007] in order to achieve a tradeoff between the accuracy of the disparity representation and the coding cost.

The remainder of this chapter is organized as follows. In Section 5.2, we describe the chosen DDE method recently presented in [Miled *et al.*, 2009a]. In Section 5.3, we address the problem of a suitable representation of the resulting dense disparity map for coding purposes. Finally, in Section 7.6, experimental results are given and some conclusions are drawn in Section 5.5.

## 5.2 Dense disparity estimation: Variational framework

The basic principle of the considered DDE method is to formulate the matching problem as a convex programming problem. More precisely, a quadratic objective function is minimized under some specific constraints and the resulting optimization problem is solved via a block-iterative algorithm, as we shall address next.

[1] http://vision.middlebury.edu/stereo/

### 5.2.1 Problem statement

#### Energy function

In this part, the stereo images are assumed to be geometrically rectified. Hence, the disparity field is restricted to the horizontal component which will be denoted by $v$. Indeed, DDE methods aim at assigning a disparity value to every pixel $(m,n)$ in $I^{(r)}$ so that the resulting field minimizes a given cost functional $\tilde{\mathcal{J}}$. Let $\mathbf{v} = (v(m,n))_{(m,n)\in\mathcal{D}}$ be the dense disparity field to be estimated where $\mathcal{D} = \{1,\dots,M\}\times\{1,\dots,N\}$ is the image support. Generally, the chosen criterion $\tilde{\mathcal{J}}$ is the sum of squared intensity differences:

$$\begin{aligned}\hat{\mathbf{v}} = (\hat{v}(m,n))_{(m,n)\in\mathcal{D}} &= \arg\min \tilde{\mathcal{J}}(\mathbf{v}) \\ &= \arg\min_{v\in\Omega} \sum_{(m,n)\in\mathcal{D}} [I^{(r)}(m,n) - I^{(l)}(m+v(m,n),n)]^2 \end{aligned} \tag{5.1}$$

where $\Omega$ is the range of candidate disparity values. Generally, an initial coarse estimate $\bar{\mathbf{v}} = (\bar{v}(m,n))_{(m,n)\in\mathcal{D}}$ of $\mathbf{v}$ is made available, say thanks to a correlation based-method. Assuming that the magnitude difference of both fields $\mathbf{v}$ and $\bar{\mathbf{v}}$ is relatively small, the warped left image around $\bar{\mathbf{v}}$ is approximated by a Taylor expansion:

$$I^{(l)}(m+v(m,n),n) \simeq I^{(l)}(m+\bar{v}(m,n),n) + \nabla I^{(l)}(m+\bar{v}(m,n),n)(v(m,n)-\bar{v}(m,n)) \tag{5.2}$$

where $\nabla I^{(l)}(m+\bar{v}(m,n),n)$ is the horizontal gradient of the warped left image.
In the following, the spatial position $(m,n)$ in each image will be denoted by $s$ for notation concision. Using the linearization (5.2), the criterion $\tilde{\mathcal{J}}$ in (6.23) can be approximated by a quadratic functional $\mathcal{J}$ which is convex in $v$:

$$\mathcal{J}(\mathbf{v}) = \sum_{s\in\mathcal{D}} [r(s) - L(s)\ v(s)]^2, \qquad \text{with } \mathbf{v} = (v(s))_{s\in\mathcal{D}}, \tag{5.3}$$

where

$$\begin{aligned} L(s) &= \nabla I^{(l)}(m+\bar{v}(s),n) \\ r(s) &= I^{(r)}(s) - I^{(l)}(m+\bar{v}(s),n) + \bar{v}(s)\ L(s). \end{aligned}$$

Now, the goal is to recover the disparity image from the observed fields $L(s)$ and $r(s)$. Since $L(s)$ may be zero, an additive term has been introduced in the criterion $\mathcal{J}$ to make it strictly convex, in compliance with the assumption required to guarantee the convergence of the algorithm we will use. Therefore, $\mathcal{J}$ becomes:

$$J(\mathbf{v}) = \sum_{s\in\mathcal{D}} [r(s) - L(s)\ v(s)]^2 + \alpha \sum_{s\in\mathcal{D}} [v(s) - \bar{v}(s)]^2, \qquad \text{with } \mathbf{v} = (v(s))_{s\in\mathcal{D}} \tag{5.4}$$

where $\alpha$ is a positive real number. Although this additional term tends to make the disparity field close to the initial values, we emphasize that the primary role of this term is not to regularize the solution but to make $\mathcal{J}$ strictly convex.

Minimizing this quadratic objective function is an ill-posed problem as the components of $L$ may locally vanish (for example in homogeneous regions). Thus, to convert this problem to a well-posed one, it is useful to incorporate additional constraints so as to formulate the problem within a set theoretic framework [Combettes, 1993].

### Convex constraints

The goal of set theoretic estimation is to obtain a feasible solution satisfying multiple constraints reflecting the prior knowledge about the disparity field. Typically, each constraint is represented by a closed convex set $S_k$ with $k \in \{1, \ldots, K\}$, in a Hilbert image space $\mathcal{H}$, endowed with the standard scalar product $\langle . \mid . \rangle$ and the associated Euclidean norm $\| \, . \, \|$. The intersection $S$ of all the $K$ sets $(S_k)_{1 \leq k \leq K}$, the feasibility set, constitutes the family of admissible solutions [Miled *et al.*, 2006b; Miled *et al.*, 2009a]. Therefore, the stereo matching problem can be stated as a constrained optimization problem where the quadratic objective function (given by Eq. (5.4)) is minimized over the feasibility set. A new formulation of the matching problem is then:

$$\text{Find } \hat{\mathbf{v}} \in S = \bigcap_{k=1}^{K} S_k \text{ such that } J(\hat{\mathbf{v}}) = \inf_{\mathbf{v} \in S} J(\mathbf{v}). \tag{5.5}$$

The constraint sets can be modeled as level sets:

$$\forall k \in \{1, \ldots, K\}, \qquad S_k = \{\mathbf{v} \in \mathcal{H} \mid f_k(\mathbf{v}) \leq \delta_k\} \tag{5.6}$$

where $f_k : \mathcal{H} \to \mathbb{R}$ is a continuous convex function for every $k \in \{1, \ldots, K\}$ and $(\delta_k)_{1 \leq k \leq K}$ are real-valued parameters such that $S = \bigcap_{k=1}^{K} S_k \neq \emptyset$.

Hence, it is required to define the convex sets $(S_k)_{1 \leq k \leq K}$ to proceed to the DDE procedure within the set theoretic framework. Indeed, the construction of the convex sets is based on the various properties of the field to be estimated. In what follows, we will focus on $K = 2$ constraints.

An example of possible prior knowledge is the range of the disparity values. Given a set of candidate disparity values, we can restrict the variation of their amplitudes within a

specific range $[v_{\min}, v_{\max}]$. This can be expressed by the following constraint set $S_1$:

$$S_1 = \{\mathbf{v} \in \mathcal{H} \mid \forall s \in \mathcal{D}, \quad v_{\min} \leq v(s) \leq v_{\max}\}. \tag{5.7}$$

Note that in practice $v_{\min}$ and $v_{\max}$ are often available.

Most importantly, another constraint can be incorporated in order to strengthen the smoothness of the disparity field in the homogeneous areas while preserving edges. Indeed, neighboring pixels belonging to the same object should have similar disparities. This can be achieved by considering the total variation $\mathsf{tv}(\mathbf{v})$ which can be defined as the sum over $\mathcal{D}$ of the norm of the spatial gradient of $\mathbf{v}$ [Rudin *et al.*, 1992]. The total variation of the discrete disparity image $\mathbf{v}$ is given by:

$$\begin{aligned}\mathrm{tv}(\mathbf{v}) = &\sum_{m=1}^{M-1}\sum_{n=1}^{N-1}\sqrt{|v(m+1,n)-v(m,n)|^2+|v(n,n+1)-v(m,n)|^2}\\ &+\sum_{m=1}^{M-1}|v(m+1,N)-v(m,N)|\\ &+\sum_{n=1}^{N-1}|v(M,n+1)-v(M,n)|.\end{aligned}$$

Hence, a total variation based regularization constraint amounts to impose an upper bound $\tau$ on $\mathsf{tv}$:

$$S_2 = \{\mathbf{v} \in \mathcal{H} \mid \mathsf{tv}(\mathbf{v}) \leq \tau\}. \tag{5.8}$$

The influence of the bound $\tau$ parameter will be discussed in Section 7.6.

Finally, the disparity estimation problem is formulated by minimizing the quadratic objective function $\mathcal{J}$ in (5.4) under the two mentioned constraint sets. Among the existing optimization algorithms, we employ a parallel block-iterative algorithm based on subgradient projections onto the convex sets [Combettes, 2003].

### 5.2.2 Optimization algorithm

Before describing the optimization algorithm developed in [Miled *et al.*, 2009a], we recall some basic facts on subgradient projections which are necessary for our minimization problem.

#### Subgradient Projections

Let $S_k$ be nonempty closed and convex subset of the Hilbert image space $\mathcal{H}$. For every point $\mathbf{v}$ in $\mathcal{H}$, there exists a unique point $P_k(\mathbf{v}) \in S_k$, called the projection of $\mathbf{v}$ onto $S_k$,

such that $\|\mathbf{v} - P_k(\mathbf{v})\| = d_{S_k}(\mathbf{v})$, where $d_{S_k}(\mathbf{v}) = \inf_{\mathbf{u}\in S_k} \|\mathbf{v} - \mathbf{u}\|$. Now suppose that $S_k$ is given by Eq. (5.6), where $f_k$ is a continuous convex function. For every $\mathbf{v} \in \mathcal{H}$, $f_k$ possesses at least one subgradient at $\mathbf{v}$, *i.e.*, a vector $\mathbf{g}_k \in \mathcal{H}$ such that

$$\forall \mathbf{z} \in \mathcal{H}, \quad \langle \mathbf{z} - \mathbf{v} \mid \mathbf{g}_k \rangle + f_k(\mathbf{v}) \leq f_k(\mathbf{z}). \tag{5.9}$$

The set of all subgradients of $f_k$ at $\mathbf{v}$ is the subdifferential of $f_k$ at $\mathbf{v}$ and it is denoted by $\partial f_k(\mathbf{v})$. If $f_k$ is differentiable at $\mathbf{v}$, then $\partial f_k(\mathbf{v}) = \{\nabla f_k(\mathbf{v})\}$. Now, fix $\mathbf{v} \in \mathcal{H}$ and a subgradient $\mathbf{g}_k \in \partial f_k(\mathbf{v})$. The subgradient projection $G_k(\mathbf{v})$ of $\mathbf{v}$ onto $S_k$ is given by:

$$G_k(\mathbf{v}) = \begin{cases} \mathbf{v} - \dfrac{f_k(\mathbf{v}) - \delta_k}{\|\mathbf{g}_k\|^2}\mathbf{g}_k & \text{, if} \quad f_k(\mathbf{v}) > \delta_k \\ \mathbf{v} & \text{, otherwise.} \end{cases} \tag{5.10}$$

Note that computing $G_k(\mathbf{v})$ requires only the availability of a subgradient (the gradient in the differentiable case) and is, therefore, often much easier than computing the exact projection $P_k(\mathbf{v})$ onto $S_k$. However, when the projection is simple to compute, one can use it as a subgradient projection corresponding to $f_k : \mathbf{v} \mapsto \|\mathbf{v} - P_k(\mathbf{v})\|$ and $\delta_k = 0$. Subsequently, we will need to compute the subgradient projections onto the sets $S_1$ and $S_2$. Exact projection onto $S_1$ is straightforwardly obtained, whereas the expression of a subgradient projection $G_2(\mathbf{v})$ of a point $\mathbf{v}$ onto $S_2$ is given by:

$$G_2(\mathbf{v}) = \begin{cases} \mathbf{v} - \dfrac{\mathrm{tv}(\mathbf{v}) - \tau}{\|\mathbf{t}_2\|^2}\mathbf{t}_2 & \text{if} \quad \mathrm{tv}(\mathbf{v}) > \tau \\ \mathbf{v} & \text{otherwise.} \end{cases} \tag{5.11}$$

where

$$\begin{aligned} \mathbf{t}_2 \in & \sum_{m=1}^{M-1}\sum_{n=1}^{N-1} \partial\sqrt{|v(m+1,n) - v(m,n)|^2 + |v(n,n+1) - v(m,n)|^2} \\ & + \sum_{m=1}^{M-1} \partial|v(m+1,N) - v(m,N)| \\ & + \sum_{n=1}^{N-1} \partial|v(M,n+1) - v(M,n)| \end{aligned} \tag{5.12}$$

More details about computing the subdifferential of the different terms involved in Eq. (5.12) can be found in [Combettes, Pesquet, 2004].

We now proceed with the description of the proposed algorithm to estimate the displacement vector field $\mathbf{v}$. This algorithm starts from an initial point $\mathbf{v}_0$ and iteratively constructs a sequence $(\mathbf{v}_i)_{i\in\mathbb{N}}$, converging to the optimal solution $\hat{\mathbf{v}}$, as follows.

**Algorithm** [Miled *et al.*, 2006b; Miled *et al.*, 2009a]

① Fix $\epsilon \in ]0, 1/K[$ and set the iteration number $i$ to zero ($i = 0$). Compute $\mathbf{v}_0$ as:

$$v_0(s) = (L(s)\ r(s) + \alpha\bar{v}(s))/(L(s)^2 + \alpha)$$

② Take a nonempty index set $\mathbb{K}_i \subseteq \{1, \ldots, K\}$.

③ For every index $k \in \mathbb{K}_i$, set $\mathbf{a}_{i,k} = G_k(\mathbf{v}_i) - \mathbf{v}_i$, where $G_k(\mathbf{v}_i)$ is a subgradient projection of $\mathbf{v}_i$ onto $S_k$ as in Eq. (5.10).

④ Choose weights $\{\zeta_{i,k}\}_{k\in\mathbb{K}_i} \subset ]\epsilon, 1[$ such that $\sum_{k\in\mathbb{K}_i} \zeta_{i,k} = 1$. Set:

- $\mathbf{z}_i = \sum_{k\in\mathbb{K}_i} \zeta_{i,k}\mathbf{a}_{i,k}$,
- $\kappa_i = \sum_{k\in\mathbb{K}_i} \zeta_{i,k} \|\mathbf{a}_{i,k}\|^2$,

⑤ If $\kappa_i = 0$, exit iteration. Otherwise set:

- $\mathbf{b}_i$ such that
$$b_i(s) = v_0(s) - v_i(s),$$
- $\mathbf{c}_i$ such that
$$c_i(s) = (L(s)^2 + \alpha)b_i(s),$$
- $\mathbf{d}_i$ such that
$$d_i(s) = z_i(s)/(L(s)^2 + \alpha),$$
- $\tilde{\kappa}_i = \kappa_i/\langle \mathbf{d}_i, \mathbf{z}_i\rangle$.

⑥ Choose $\lambda_i \in [\epsilon\tilde{\kappa}_i, \tilde{\kappa}_i]$ and set $\tilde{\mathbf{d}}_i = \lambda_i\mathbf{d}_i$.

⑦ Set:

- $\pi_i = -\langle \mathbf{c}_i, \tilde{\mathbf{d}}_i\rangle$,
- $\mu_i = \langle \mathbf{b}_i, \mathbf{c}_i\rangle$,
- $\nu_i = \lambda_i\langle \tilde{\mathbf{d}}_i, \mathbf{z}_i\rangle$,
- $\rho_i = \mu_i\nu_i - \pi_i^2$.

⑧ Set:

$$\mathbf{v}_{i+1} = \begin{cases} \mathbf{v}_i + \tilde{\mathbf{d}}_i, & \text{if} \quad \rho_i = 0, \quad \pi_i \geq 0, \\ \mathbf{v}_0 + (1 + \pi_i/\nu_i)\tilde{\mathbf{d}}_i & \text{if} \quad \rho_i > 0, \quad \pi_i\nu_i \geq \rho_i, \\ \mathbf{v}_i + \frac{\nu_i}{\rho_i}(\pi_i \mathbf{b}_i + \mu_i \tilde{\mathbf{d}}_i) & \text{if} \quad \rho_i > 0, \quad \pi_i\nu_i < \rho_i. \end{cases}$$

⑨ Increment $i$ and go to step ②.

As stop criterion, a difference magnitude observation of the current solution and the solution at the previous iteration may be used. Otherwise, a maximum number of iterations can be specified.

Note that, as proved in [Miled *et al.*, 2009a], if there exists a positive integer $W$ such that

$$\forall i \in \mathbb{N}, \qquad \bigcup_{l=i}^{i+W-1} \mathbb{K}_l = \{1, \ldots, K\}, \tag{5.13}$$

then every sequence $(\mathbf{v}_i)_{i\in\mathbb{N}}$ generated by Algorithm 1 converges to the unique solution of (5.5).

Finally, some comments about this optimization algorithm can be done at this point:

- This algorithm allows us to easily incorporate additional convex constraints if these are available. Its ability to use approximate (subgradient) projections onto the constraint sets makes it possible to handle a wide range of complex convex constraints.
- Due to its block iterative structure, this algorithm offers a lot of flexibility in terms of parallel implementation. Indeed, the set $\mathbb{K}_i$ defines the constraints to be activated at iteration $i$ and, according to Eq. (5.13), different blocks of constraint sets may be used. Therefore, the optimization algorithm can be efficiently implemented on parallel computing architectures by adapting the number of elements in $\mathbb{K}_i$ to the number of available parallel processors.
- Concerning the computational complexity of this algorithm, the main computation at one iteration is related to the calculation of a subgradient for the total variation constraint which can be estimated in $O(MN + M + N)$, where $M$ and $N$ are respectively the width and the height of the image. Concerning the global computational cost, complexity can be estimated in $O(i(MN + M + N))$ where $i$ is the number of iterations required for the convergence of the optimization algorithm.

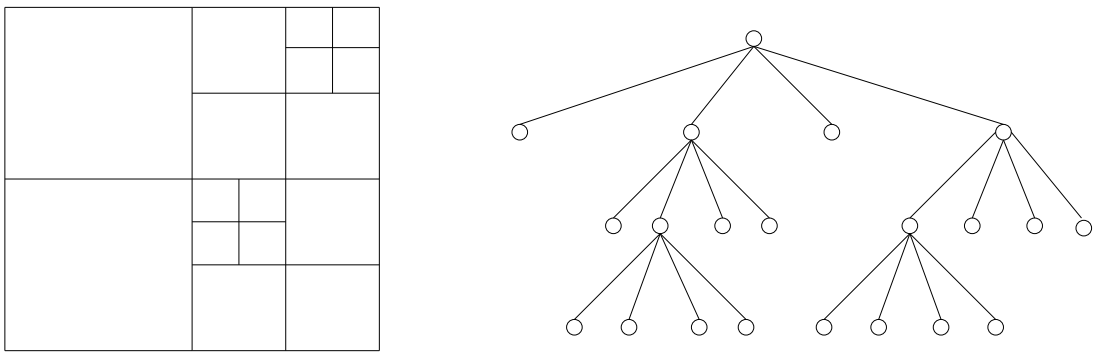
In the next section, we will describe our main contribution which consists of representing and efficiently encoding the dense field issued from this algorithm.

## 5.3 Proposed dense disparity map coding method

Unlike BDE method where the disparity is estimated for each block of the images, a DDE method computes the disparity information for each pixel of the images. Therefore, the resulting dense field raises an important issue in the context of coding applications: the expensive cost of transmission/storage of the disparity map in terms of bitrate. To overcome this problem, we propose to apply a quadtree decomposition (QTD) to the estimated field $\hat{\mathbf{v}}$ followed by an entropy coding of the segmented field.

### 5.3.1 Partition-based segmentation

Generally speaking, a quadtree decomposition method aims at subdividing an image or a block into four subblocks according to a given criterion. The segmentation is repeatedly applied until the size of subblock reaches $1 \times 1$ or some criterion is met. An example of quadtree decomposition is illustrated in Fig. 5.1.



(a) (b)

Figure 5.1: Illustration of quadtree decomposition: (a): image partitions, (b): corresponding quadtree structure.

In our context, the segmentation procedure attempts to adapt the generated dense disparity map into a block-based representation as defined in the H.264 AVC standard [ITU-T Recommentation H.264 & ISO/IEC 14496-10 AVC, 2003]. Note that in this standard, the field is estimated for variable size blocks.

To this end, the map $\hat{\mathbf{v}}$ is firstly segmented into a set of nonoverlapping macroblocks $\mathcal{B}_k$ of size $16 \times 16$. Then, the homogeneity of $\mathcal{B}_k$, retained as a criterion for the segmentation

step, is measured by its local variance $\sigma^2_{\mathcal{B}_k}$. Thus, for a given threshold value $T$, if $\sigma^2_{\mathcal{B}_k} \leq T$, all the disparity values within $\mathcal{B}_k$ are replaced by a single disparity value: it may be their average or their median value. At the opposite, if $\sigma^2_{\mathcal{B}_k} > T$, $\mathcal{B}_k$ is subdivided into four sub-macroblocks and then the test of homogeneity is recursively repeated until the size of the current sub-macroblock reaches $4 \times 4$ (which corresponds to the minimum sub-macroblock size defined in the H.264 AVC standard). Therefore, homogeneous areas will be represented by larger blocks, whereas small blocks will correspond to object boundaries or occluded regions.

Consequently, unlike block-based technique in which the disparity estimation is causal and local, our proposed estimation has a global approach which promotes the regularization of the disparity field. Furthermore, the main relevance of the segmentation of the dense disparity map which finally ends in a block-based representation is to make good use of the smoothness of the dense disparity field. Thus, the neighboring disparity values become highly correlated which may result in decreasing the bitrate of the disparity information.

### 5.3.2 Entropy coding with H.264 JM software

It is worth pointing out that the segmented disparity map and the mode information corresponding to the block size assigned to each disparity value must be transmitted as side information. For this purpose, we have employed the encoding module integrated in the H.264 JM software based on the classical DPCM technique with median predictors and variable-length codes.

An alternative approach for encoding the dense disparity map would be to consider this map as a component picture, and thus, utilizing some near-lossless image encoders like H.264 AVC in *intra* mode or other wavelet-based coding methods [Daribo *et al.*, 2008]. For the moment, these methods lead to a much higher amount of bits for encoding smooth field, and therefore, in our joint stereo image coding scheme, we prefer the described approach based on segmenting the dense disparity field.

## 5.4 Experimental results

Simulations results are performed on the following real stereo image pairs: “Shrub”, “Spot5”, “Book arrival” (taken from the “Book arrival” multiview sequence, frame 1) and “Outdoor”

(taken from the "Outdoor" multiview sequence, frame 1) which are all rectified. Before illustrating the performances of integrating the proposed dense disparity estimation method in a joint stereo image coding scheme, we firstly discuss the influence of the parameters defined in the estimation algorithm.

### 5.4.1 Influence of the parameters

In practice, the optimal value of the parameters $[v_{\min}, v_{\max}]$, $\tau$ and $\alpha$ may not be known exactly and it is, therefore, important to evaluate their impact in terms of coding rate and PSNR of the disparity compensated image.

#### Range values

The choice of the range $[v_{\min}, v_{\max}]$ can be accurately found by matching certain points of interest selected manually in the two stereo views.

#### Regularization parameters

The upper bound $\tau$, used to enforce the smoothness of the estimated disparity map, may be estimated from a scale value of the total variation of the initial disparity map $\bar{v}_x$, as shown in Fig. 5.2. A low scale value results in smoothing more the disparity map, and so, reducing the number of bits required for the transmission.
The parameter $\alpha$ introduced in (5.4) has to be set in order to weight the influence of the additive term introduced to make $\mathcal{J}$ strictly convex. By choosing a high value of $\alpha$, the criterion $\mathcal{J}$ simply reduces to the distance between $\mathbf{v}$ and $\bar{\mathbf{v}}$. Conversely, assigning a small value privileges the first data term.

Table 5.1: Example of the influence of the parameter $\tau$ with $\alpha = 6$ on the bitrate and the PSNR of the dense disparity compensated image for the "Book arrival" stereo pair.

| | bitrate (H.264 intra, QP=0) | PSNR |
|---|---|---|
| $\tau = 50000$ | 0.641 bpp | 37.13 dB |
| $\tau = 40000$ | 0.522 bpp | 37.23 dB |
| $\tau = 30000$ | 0.403 bpp | 37.24 dB |
| $\tau = 20000$ | 0.305 bpp | 36.48 dB |
| $\tau = 10000$ | 0.287 bpp | 33.44 dB |

Table 5.2: Example of the influence of the parameter $\alpha$ with $\tau = 30000$ on the bitrate and the PSNR of the dense disparity compensated image for the "Book arrival" stereo pair.

| | bitrate (H.264 intra, QP=0) | PSNR |
|---|---|---|
| $\alpha = 0.1$ | 0.466 bpp | 35.79 dB |
| $\alpha = 6$ | 0.403 bpp | 37.24 dB |
| $\alpha = 10$ | 0.394 bpp | 37.20 dB |
| $\alpha = 50$ | 0.390 bpp | 36.50 dB |
| $\alpha = 100$ | 0.383 bpp | 35.78 dB |

Table 5.1 and Table 5.2 show the impact of the parameters $\tau$ and $\alpha$ on the coding rate of the disparity map and on the quality of the disparity compensated image, evaluated through the PSNR between the original view and the disparity-compensated one. Note that the bitrate of the dense disparity field has been computed using the dense disparity field as a picture with H.264 JM software in *intra* mode at a quantization parameter QP of 0. First, an arbitrary fixed value of $\alpha$ is used to determine the parameter $\tau$. Then, the optimal value of $\alpha$ is determined. The value of both parameters is selected according on the highest PSNR value of the disparity compensated picture.
Based on this strategy, we have obtained heuristically the set of parameters for each stereo pair as depicted in Table 5.3.

Table 5.3: Parameter settings.

| | range values | $\tau$ | $\alpha$ |
|---|---|---|---|
| Shrub | [1,15] | 10000 | 10 |
| Spot5-5 | [1,64] | 30000 | 10 |
| Book arrival | [14,35] | 30000 | 6 |
| Outdoor | [8,32] | 15000 | 10 |

### 5.4.2 Performances in stereo image coding context

Figure 5.3 shows the "Outdoor" stereo image and the disparity maps provided by the Block-based disparity estimation method (BDE) and the proposed Dense disparity estimation method (DDE). It can be noticed that the DDE method generates the smoothest map. Furthermore, the resulting bitrate $R^{(\mathbf{v})}$ for the dense map is 50 times greater than the

bitrate of the block-based map. However, by exploiting the smoothness of the dense map owing to a QTD with median filtering (DDE+QTD) followed by a rounding operation, $R^{(\mathbf{v})}$ is dramatically reduced. The corresponding disparity map is displayed in Fig. 5.3(e): it is more robust to noise estimation compared with the BDE. Similar illustrations are also given for "Book arrival" (Fig. 5.4) and "Shrub" (Fig. 5.5) stereo images.

In order to emphasize the benefit from using the DDE, we will consider two stereo image coding schemes (designated by scheme B and VLS-II) which have been already described in the previous chapter. Recall that scheme B is the basic algorithm which consists of coding the left and the residual images, whereas VLS-II corresponds to our proposed Vector Lifting Scheme. Figures 6.10 and 5.7 illustrate the scalability in quality of the reconstruction procedure by providing the PSNR versus the bitrate for the stereo pairs "Shrub" and "Outdoor". It is worth pointing out that the retained DDE method followed by a quadtree decomposition outperforms the classical BDE method by 0.3-1 dB for our VLS-II (resp. by 0.1-0.9 dB for the scheme B) whatever the employed decomposition scheme of the stereo pair.

It is worth pointing that our proposed stereo still image coding scheme based on VLS is a lossy-to-lossless technique in which the disparity map is estimated only once, and from this point of view it differs from existing stereo video schemes based on H.264 coding tools where the disparity is estimated for different bitrates (i.e for each point of the PSNR curve). However, the benefit of the retained DDE has been also confirmed when considering a multiview video coding framework [Daribo *et al.*, 2009]. Some related results will be given in the next part.

### 5.4.3 Performances in multiview video coding context

Multiview video coding is an emerging application where, in addition to classical temporal estimation, an efficient disparity estimation should be performed in order to achieve the best compression performance. A popular coder is the multiview video coding (MVC) extension of H.264/AVC, which uses a block-based disparity estimation (just like temporal estimation in H.264/AVC) [Chen *et al.*, 2009].

Within the MVC framework, the disparity estimation module generates different disparity fields at different QP values. More precisely, for each point of the PSNR-plot, an optimal field (in a rate-distortion sense) is estimated by minimizing a Lagrangian cost function

according to the available bit budget or allowable distortion.

Inspired by these works in video coding, the retained DDE method has been recently employed in a Multiview Video Coding (MVC) framework [Daribo *et al.*, 2009]. Thus, the estimated dense disparity field is segmented at different QP values by performing a rate-distortion optimization technique. Figures 5.8 and 5.9 show the results in terms of rate-distortion performance when using the block-based reference estimation in the MVC and the DDE method followed by a segmentation optimized in a rate-distortion sense. These plots indicate that the proposed approach achieves a gain of about 0.1-1.5 dB compared with the classical approach.

## 5.5 Conclusion

In this chapter, we have presented the benefits of integrating a dense disparity map in stereo image coding and multiview video coding frameworks. Unlike conventional methods that employ a block-based disparity estimation method, our method uses a dense and smooth disparity map. Experimental results showed that our approach produces better results than classical methods.

Left image

Right image

Initial disparity map $\bar{\mathbf{v}}$

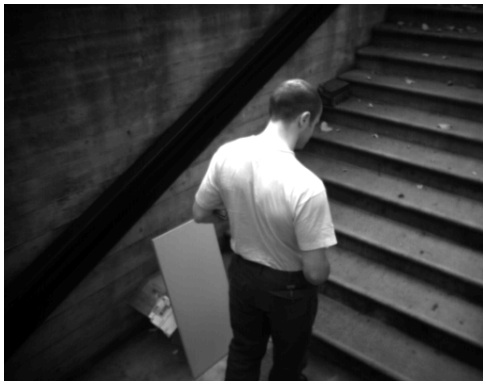
$\tau = 0.15 \cdot \mathrm{tv}(\bar{\mathbf{v}})$

$\tau = 0.10 \cdot \mathrm{tv}(\bar{\mathbf{v}})$

$\tau = 0.05 \cdot \mathrm{tv}(\bar{\mathbf{v}})$

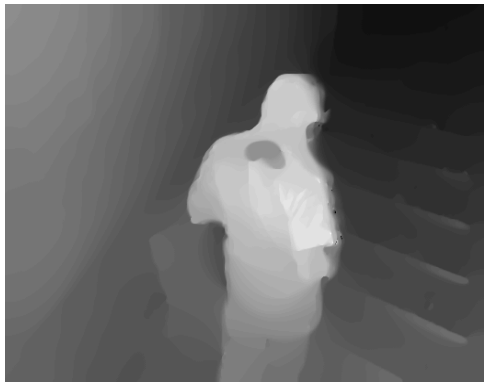
Figure 5.2: Example of dense disparity maps at different values of the upper bound $\tau$ parameter for the "Book arrival" stereo pair.

(a) Left image

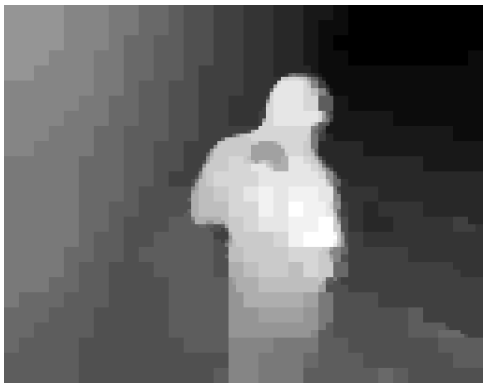


(b) Right image

(c) BDE method ($R^{(v)} = 0.072$ bpp)

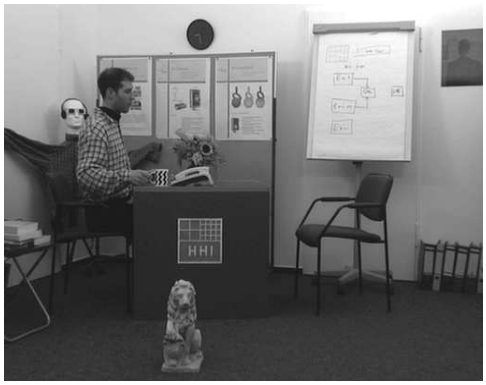


(d) DDE method ($R^{(v)} = 3.990$ bpp)



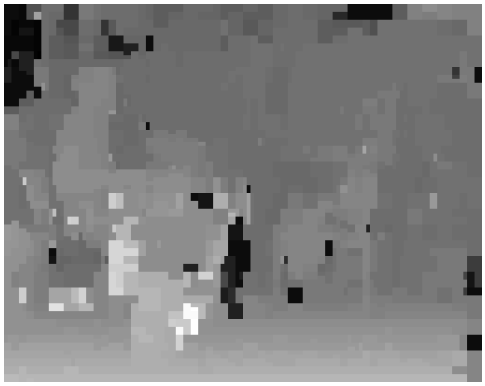
(e) DDE + QTD method ($R^{(v)} = 0.041$ bpp)

Figure 5.3: Extracted disparity maps for the "Outdoor" stereo image using (c) a block-based DE method (d) the proposed DDE method (e) a quadtree decomposition of the computed dense disparity map ($T$=0.2).



(a) Left image

(b) Right image



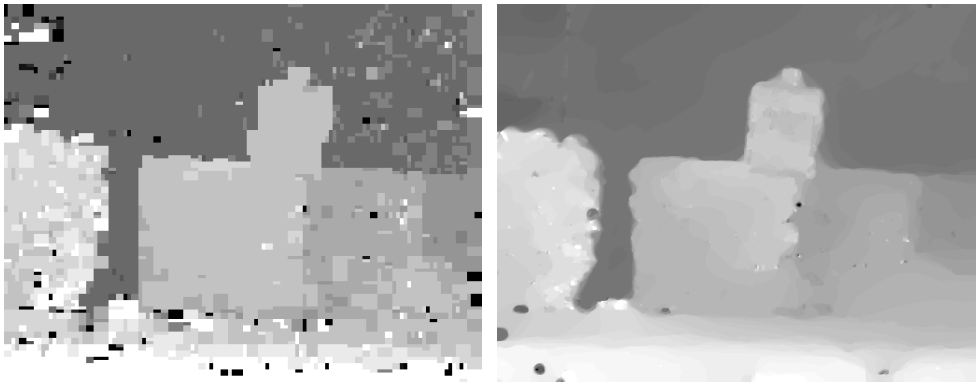
(c) BDE method ($R^{(v)} = 0.068$ bpp)

(d) DDE method ($R^{(v)} = 3.283$ bpp)

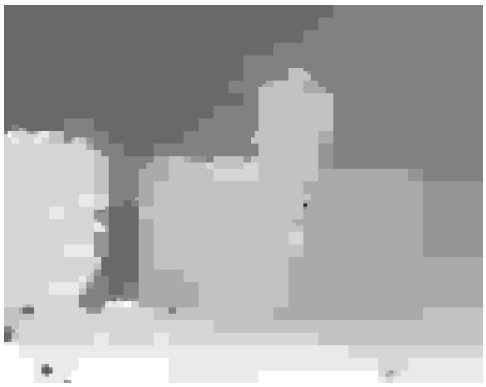
(e) DDE + QTD method ($R^{(v)} = 0.059$ bpp)

Figure 5.4: Extracted disparity maps for the "Book arrival" stereo image using (c) a block-based DE method (d) the proposed DDE method (e) a quadtree decomposition of the computed dense disparity map ($T$=0.1).


(a) Left image (b) Right image



(c) BDE method ($R^{(v)} = 0.061$ bpp) (d) DDE method ($R^{(v)} = 3.011$ bpp)



(e) DDE + QTD method ($R^{(v)} = 0.041$ bpp)

Figure 5.5: Extracted disparity maps for the "Shrub" stereo image using (c) a block-based DE method (d) the proposed DDE method (e) a quadtree decomposition of the computed dense disparity map ($T$=0.2).

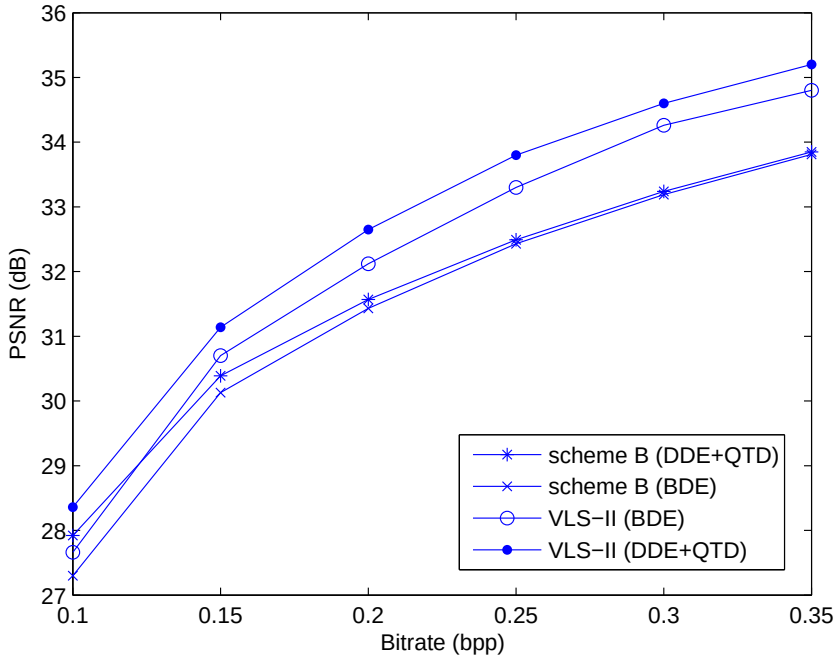



Figure 5.6: PSNR (in dB) versus the bitrate (bpp) after JPEG 2000 encoding for the SI pair "shrub".

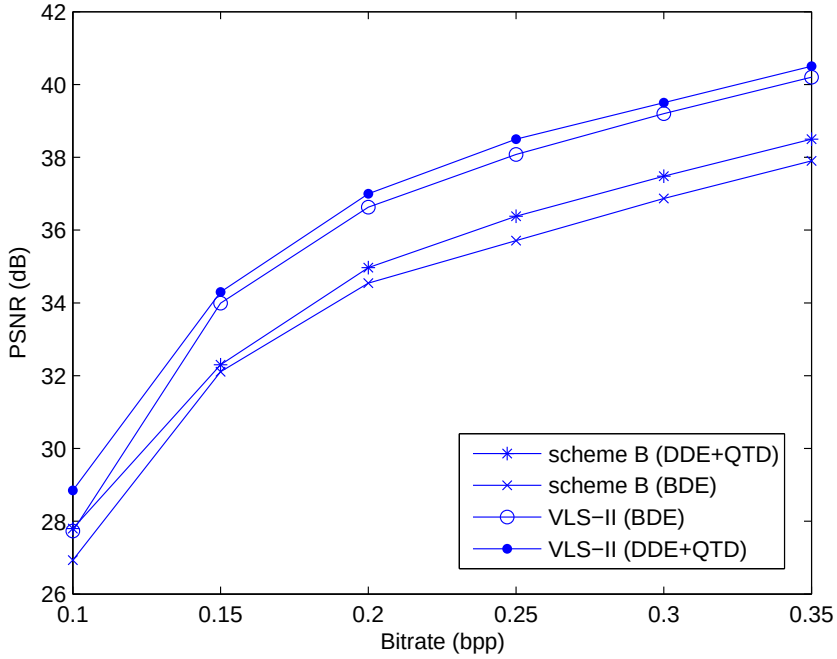



Figure 5.7: PSNR (in dB) versus the bitrate (bpp) after JPEG 2000 encoding for the SI pair "Outdoor".

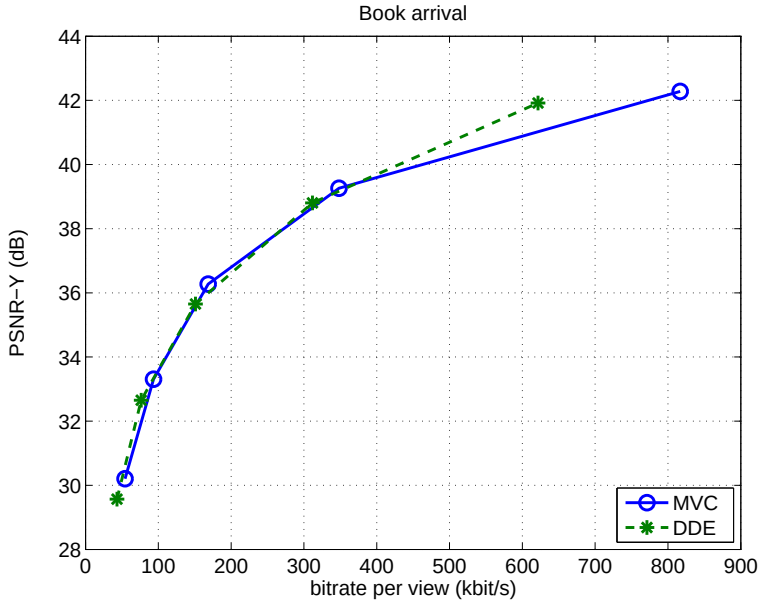



Figure 5.8: Rate-distortion coding results for the multiview video sequence "Book arrival".

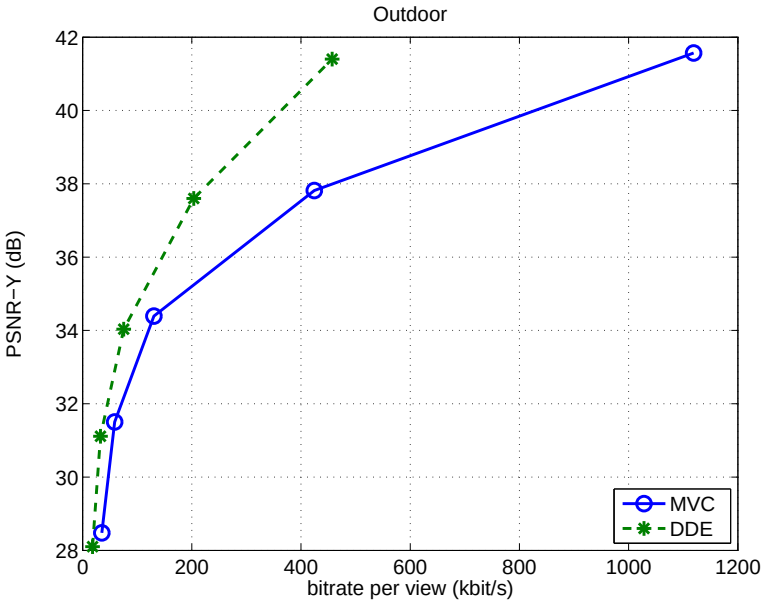



Figure 5.9: Rate-distortion coding results for the multiview video sequence "Outdoor".

# Chapter 6

# Non separable lifting scheme with adaptive update step

## 6.1 Introduction

In the two previous chapters, the stereo image coding process has been performed by using separable lifting schemes (LS). Such structures are generally handled in a separable way for two-dimensional (2D) signals by cascading 1D LS along the horizontal direction, then along the vertical direction. However, it is well known that conventional separable LS may not appear very efficient to cope with the two-dimensional characteristics of contours which are neither horizontal nor vertical. For instance, Moellenhoff and Maier [Moellenhoff, Maier, 1998a] analyzed the characteristics of the residual images and proved that such images have properties different from natural images. This suggests that transforms working well for natural images may not be as well-suited for residual images.

Based on these observations, we propose in this chapter to use 2D non separable lifting schemes (NSLS) that still enable progressive reconstruction and exact decoding of images. More precisely, we will focus on the optimization of *all* the operators (i.e prediction and update filters) in order to build a content-adaptive decomposition. Indeed, we design the prediction operators by minimizing the detail signal variance, which is a common criterion. Concerning the update filter, we propose a new optimization criterion which aims at reducing the aliasing effects. Furthermore, we show that the proposed optimization method leads to the same optimal update filter when the optimization is performed either before or after the decimation step.

The remainder of this chapter is organized as follows. In Section 6.2, a 2D non separable lifting structure is presented and studied. In Section 6.3, we describe the proposed approach for the design of both optimal prediction and update filters. In Section 6.4, we conduct a theoretical analysis of the proposed method in order to evaluate the gain achieved by the proposed update filter optimization together with a simple statistical modeling of images. We show that in the case of a first order 2D autoregressive model for the input images, the filters can be easily deduced only based on the spatial correlation coefficients. Finally, in Section 6.6, experimental results are given and some conclusions are drawn in Section 6.7.

## 6.2 Non separable lifting schemes

### 6.2.1 Motivation

Instead of using samples from the same rows (resp. columns) while processing the image along the lines (resp. columns), 2D NSLS provide smarter choices in the selection of the samples by using horizontal, vertical and oblique directions at the prediction step [Gerek, Çetin, 2000]. As an example, quincunx lifting schemes were found to be efficient for coding satellites images acquired on a quincunx sampling grid [Gouze *et al.*, 2001; Benazza-Benyahia *et al.*, 2007]. However, for natural images, we should note that it has been observed that quincunx LS present an important limitation in the sense that they are less performant than the conventional separable LS. To confirm these observations, one can refer to results presented in [Benazza-Benyahia *et al.*, 2007; Gouze *et al.*, 2004]. In [Heijmans *et al.*, 2002], a 2D wavelet decomposition comprising an adaptive update lifting step and three consecutive fixed prediction lifting steps is proposed. Another structure, which is composed of three prediction lifting steps followed by an update lifting step, has also been considered in the nonadaptive case [Chokchaitam, 2004; Sun, 2004]. The interest of this structure is twofold. First, its integer version is different from the separable lifting structure and shows a better coding performance due to the reduction of lifting steps and rounding operations. Furthermore, it is equivalent to any separable predict-update LS structure.

Due to these advantages, we will retain the latter 2D NSLS structure in the rest of this work.

### 6.2.2 Principle of the retained 2D NSLS structure

Let $x$ denote the digital image to be coded. At each resolution level $j$ and each pixel location $(m, n)$, its approximation coefficient is denoted by $x_j(m, n)$ and the associated four polyphase components by $x_{0,j}(m, n) = x_j(2m, 2n)$, $x_{1,j}(m, n) = x_j(2m, 2n + 1)$, $x_{2,j}(m, n) = x_j(2m + 1, 2n)$, and $x_{3,j}(m, n) = x_j(2m + 1, 2n + 1)$. Without loss of generality, we assume that the polyphase components are the input coefficients of the 2D NSLS depicted in Fig. 6.1, where $\mathbf{P}_j^{(HH)}$, $\mathbf{P}_j^{(LH)}$, $\mathbf{P}_j^{(HL)}$ and $\mathbf{U}_j$ represent the four analysis filters employed to generate the detail coefficients $x_{j+1}^{(HH)}$ oriented diagonally, $x_{j+1}^{(LH)}$ oriented vertically, $x_{j+1}^{(HL)}$ oriented horizontally, and the approximation coefficients $x_{j+1}$. It is easy to

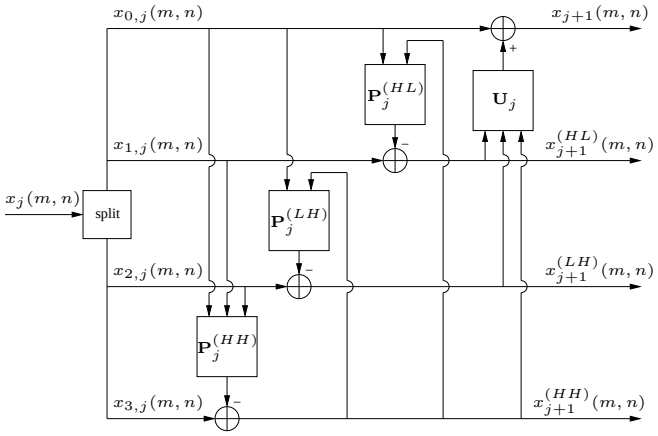



Figure 6.1: NSLS decomposition structure.

derive the expressions of the resulting coefficients in the 2D $z$-transform domain.[1] Indeed, the $z$-transforms of the output coefficients can be written as follows:

$$X_{j+1}^{(HH)}(z_1, z_2) = X_{3,j}(z_1, z_2) - \lfloor P_{0,j}^{(HH)}(z_1, z_2)X_{0,j}(z_1, z_2) + P_{1,j}^{(HH)}(z_1, z_2)X_{1,j}(z_1, z_2) + P_{2,j}^{(HH)}(z_1, z_2)X_{2,j}(z_1, z_2)\rfloor, \tag{6.1}$$

$$X_{j+1}^{(LH)}(z_1, z_2) = X_{2,j}(z_1, z_2) - \lfloor P_{0,j}^{(LH)}(z_1, z_2)X_{0,j}(z_1, z_2) + P_{1,j}^{(LH)}(z_1, z_2)X_{j+1}^{(HH)}(z_1, z_2)\rfloor, \tag{6.2}$$

$$X_{j+1}^{(HL)}(z_1, z_2) = X_{1,j}(z_1, z_2) - \lfloor P_{0,j}^{(HL)}(z_1, z_2)X_{0,j}(z_1, z_2) + P_{1,j}^{(HL)}(z_1, z_2)X_{j+1}^{(HH)}(z_1, z_2)\rfloor, \tag{6.3}$$

[1] The $z$-transform of a signal $x$ will be denoted in capital letters by $X$.

$$X_{j+1}(z_1,z_2) = X_{0,j}(z_1,z_2) + \lfloor U_j^{(HL)}(z_1,z_2)X_{j+1}^{(HL)}(z_1,z_2) + U_j^{(LH)}(z_1,z_2)X_{j+1}^{(LH)}(z_1,z_2) + U_j^{(HH)}(z_1,z_2)X_{j+1}^{(HH)}(z_1,z_2)\rfloor \quad (6.4)$$

where, for every $i \in \{0,1,2\}$ and $o \in \{HH, HL, LH\}$,

$$P_{i,j}^{(o)}(z_1,z_2) = \sum_{(k,l)\in\mathcal{P}_{i,j}^{(o)}} p_{i,j}^{(o)}(k,l) z_1^{-k} z_2^{-l}, \quad (6.5)$$

$$U_j^{(o)}(z_1,z_2) = \sum_{(k,l)\in\mathcal{U}_j^{(o)}} u_j^{(o)}(k,l) z_1^{-k} z_2^{-l}. \quad (6.6)$$

The set $\mathcal{P}_{i,j}^{(o)}$ (resp. $\mathcal{U}_j^{(o)}$) and the coefficients $p_{i,j}^{(o)}(k,l)$ (resp. $u_j^{(o)}(k,l)$) denote the support and the weights of the three prediction filters (resp. of the update filter). Note that we introduced the rounding operations $\lfloor . \rfloor$ in order to allow lossy-to-lossless encoding of the coefficients.

Once the considered NSLS structure has been defined, it may appear instructive to see how it is related to some conventional separable lifting structures.

### 6.2.3 Links with conventional separable lifting structures

It can be checked that the conventional 5/3 transform and the Haar one are particular cases of the structure illustrated in Figure 6.1. For example, for the separable 5/3 transform which was selected for the lossless mode of the JPEG2000 standard, we get:

$$P_{0,j}^{(HH)}(z_1,z_2) = -\frac{1}{4}(1+z_1+z_2+z_1z_2),$$

$$P_{1,j}^{(HH)}(z_1,z_2) = \frac{1}{2}(1+z_1), P_{2,j}^{(HH)}(z_1,z_2) = \frac{1}{2}(1+z_2), \quad (6.7)$$

$$P_{0,j}^{(LH)}(z_1,z_2) = \frac{1}{2}(1+z_1), P_{1,j}^{(LH)}(z_1,z_2) = -\frac{1}{4}(1+z_2^{-1}), \quad (6.8)$$

$$P_{0,j}^{(HL)}(z_1,z_2) = \frac{1}{2}(1+z_2), P_{1,j}^{(HL)}(z_1,z_2) = -\frac{1}{4}(1+z_1^{-1}), \quad (6.9)$$

$$U_j^{(HL)}(z_1,z_2) = \frac{1}{4}(1+z_1^{-1}), U_j^{(LH)}(z_1,z_2) = \frac{1}{4}(1+z_2^{-1}),$$

$$U_j^{(HH)}(z_1,z_2) = -\frac{1}{16}(1+z_1^{-1}+z_2^{-1}+z_1^{-1}z_2^{-1}). \quad (6.10)$$

More generally, it was shown in [Chokchaitam, 2004] that any separable predict-update LS structure (whose 1D prediction and update filters will be denoted by $\mathbf{P}_j^{1D}$ and $\mathbf{U}_j^{1D}$, respectively) has an equivalent 2D-NSLS counterpart given by Figure 6.1. For instance,

this equivalence is guaranteed by taking the 2D operators as follows:

$$P_{0,j}^{(HH)}(z_1, z_2) = -P_j^{1D}(z_1)P_j^{1D}(z_2),$$
$$P_{1,j}^{(HH)}(z_1, z_2) = P_j^{1D}(z_1), P_{2,j}^{(HH)}(z_1, z_2) = P_j^{1D}(z_2), \tag{6.11}$$
$$P_{0,j}^{(LH)}(z_1, z_2) = P_j^{1D}(z_1), P_{1,j}^{(LH)}(z_1, z_2) = -U_j^{1D}(z_2), \tag{6.12}$$
$$P_{0,j}^{(HL)}(z_1, z_2) = P_j^{1D}(z_2), P_{1,j}^{(HL)}(z_1, z_2) = -U_j^{1D}(z_1), \tag{6.13}$$
$$U_j^{(HL)}(z_1, z_2) = U_j^{1D}(z_1), U_j^{(LH)}(z_1, z_2) = U_j^{1D}(z_2),$$
$$U_j^{(HH)}(z_1, z_2) = -U_j^{1D}(z_1)U_j^{1D}(z_2). \tag{6.14}$$

Subsequently, we will show that this equivalence facilitates the derivation of adaptive extensions of the associated operators. Indeed, in a coding framework, the sparsity of any LS-based multiresolution representation clearly depends on the choice of the prediction and update operators.

In the next section, we aim at fully exploiting the flexibility of a NSLS through an adaptation of *all* the involved operators. Note that adaptive NSLS were already investigated [Taubman, 1999; Benazza-Benyahia *et al.*, 2007; Gouze *et al.*, 2004] by putting emphasis on the optimization of the predictor. One of the main contributions of this paper is the optimization of the update filter by using a criterion which allows us to reduce aliasing effects. In this way, we build a fully-optimized NSLS which is better adapted to the characteristics of the input image. Moreover, we show that the proposed optimization method leads to the same optimal update filter when the optimization is performed either before or after the decimation step.

## 6.3 Proposed optimization method

In what follows, for the sake of simplicity, we assume that the initial image $x$ (after subtracting its mean value) is modeled as the realization of a zero-mean wide-sense stationary random process. If we omit the rounding operations, at a given resolution level $j$, the approximation image $x_j$ is also the realization of a 2D wide sense stationary process with autocorrelation function $R_{x_j}$:

$$\forall (k,l) \in \mathbb{Z}^2, \quad R_{x_j}(k,l) = \mathsf{E}[x_j(m,n)x_j(m-k,n-l)]. \tag{6.15}$$

### 6.3.1 Optimization of the predictors

Since the detail coefficients are defined as prediction errors, the prediction operators can be optimized so as to minimize the variance of the coefficients at each resolution level. The rounding operators being omitted, it is readily shown that the minimum variance predictors must satisfy the well-known Yule-Walker equations. For example, for the prediction vector $\mathbf{p}_j^{(HH)}$, the normal equations read

$$\mathsf{E}[\tilde{\mathbf{x}}_j^{(HH)}(m,n)\tilde{\mathbf{x}}_j^{(HH)}(m,n)^\top]\mathbf{p}_j^{(HH)} = \mathsf{E}[x_{3,j}(m,n)\tilde{\mathbf{x}}_j^{(HH)}(m,n)] \tag{6.16}$$

where

- $\mathbf{p}_j^{(HH)} = (\mathbf{p}_{0,j}^{(HH)}, \mathbf{p}_{1,j}^{(HH)}, \mathbf{p}_{2,j}^{(HH)})^\top$ is the prediction vector where, for each $i \in \{0,1,2\}$,

$$\mathbf{p}_{i,j}^{(HH)} = \Big(p_{i,j}^{(HH)}(k,l)\Big)_{(k,l)\in\mathcal{P}_{i,j}^{(HH)}},$$

- $\tilde{\mathbf{x}}_j^{(HH)}(m,n) = (\mathbf{x}_{0,j}^{(HH)}(m,n), \mathbf{x}_{1,j}^{(HH)}(m,n), \mathbf{x}_{2,j}^{(HH)}(m,n))^\top$ is the reference vector with

$$\mathbf{x}_{i,j}^{(HH)}(m,n) = \Big(x_{i,j}(m-k,n-l)\Big)_{(k,l)\in\mathcal{P}_{i,j}^{(HH)}}.$$

The other optimal prediction filters $\mathbf{p}_j^{(HL)}$ and $\mathbf{p}_j^{(LH)}$ are obtained in a similar way.

### 6.3.2 Optimization of the update operator

Optimizing the update operator is less obvious than optimizing the predictions. An update optimization method has been proposed in [Pesquet-Popescu, 1999; Gouze *et al.*, 2004]. It consists of designing the update operator in order to minimize a reconstruction error. In other words, the update operator is optimized by minimizing the distortion between the original image $x_j$ and the reconstructed image $\hat{x}_j$, when the detail coefficients are canceled. In the case of the NSLS, this criterion is given by:

$$\begin{aligned}\mathcal{J} &= \mathsf{E}[(x_j(m,n) - \hat{x}_j(m,n))^2] \\ &= \frac{1}{4}\Big(\mathsf{E}[(x_{0,j}(m,n) - \hat{x}_{0,j}(m,n))^2] + \mathsf{E}[(x_{1,j}(m,n) - \hat{x}_{1,j}(m,n))^2] \\ &+ \mathsf{E}[(x_{2,j}(m,n) - \hat{x}_{2,j}(m,n))^2] + \mathsf{E}[(x_{3,j}(m,n) - \hat{x}_{3,j}(m,n))^2]\Big).\end{aligned} \tag{6.17}$$

The sets $\left(\hat{x}_{i,j}(m,n)\right)_{i\in\{0,1,2,3\}}$ are defined by the synthesis lifting scheme while setting the details coefficients to zero. After some calculations, we get:

$$\hat{x}_{0,j}(m,n) = x_{0,j}(m,n) + \mathbf{u}_j^\top \mathbf{x}_{j+1}(m,n) \tag{6.18}$$

$$\hat{x}_{1,j}(m,n) = \left(\mathbf{p}_{0,j}^{(HL)}\right)^\top \mathbf{x}_{0,j}^{(HL)}(m,n) + \mathbf{u}_j^\top \mathbf{D}_{0,j+1}^{(HL)}(m,n)\mathbf{p}_{0,j}^{(HL)} \tag{6.19}$$

$$\hat{x}_{2,j}(m,n) = \left(\mathbf{p}_{0,j}^{(LH)}\right)^\top \mathbf{x}_{0,j}^{(LH)}(m,n) + \mathbf{u}_j^\top \mathbf{D}_{0,j+1}^{(LH)}(m,n)\mathbf{p}_{0,j}^{(LH)} \tag{6.20}$$

$$\begin{aligned}\hat{x}_{3,j}(m,n) = &\left(\mathbf{p}_{0,j}^{(HH)}\right)^\top \mathbf{x}_{0,j}^{(HH)}(m,n) + \mathbf{u}_j^\top \mathbf{D}_{0,j+1}^{(HH)}(m,n)\mathbf{p}_{0,j}^{(HH)} \\ &+ \left(\mathbf{p}_{1,j}^{(HH)} \otimes \mathbf{p}_{0,j}^{(HL)}\right)^\top \mathbf{x}_{0,j}^{(HH)}(m,n) + \mathbf{u}_j^\top \mathbf{D}_{1,j+1}^{(HH)}(m,n)\left(\mathbf{p}_{1,j}^{(HH)} \otimes \mathbf{p}_{0,j}^{(HL)}\right) \\ &+ \left(\mathbf{p}_{0,j}^{(LH)} \otimes \mathbf{p}_{2,j}^{(HH)}\right)^\top \mathbf{x}_{0,j}^{(HH)}(m,n) + \mathbf{u}_j^\top \mathbf{D}_{2,j+1}^{(HH)}(m,n)\left(\mathbf{p}_{2,j}^{(HH)} \otimes \mathbf{p}_{0,j}^{(LH)}\right)\end{aligned} \tag{6.21}$$

where $\otimes$ is the Kronecker product and

- $\mathbf{u}_j = \left(u_j^{(o)}(k,l)\right)^\top_{(k,l)\in\mathcal{U}_j^{(o)},o\in\{HL,LH,HH\}}$ is the update weight vector,
- $\mathbf{x}_{j+1}(m,n) = \left(x_{j+1}^{(o)}(m-k,n-l)\right)^\top_{(k,l)\in\mathcal{P}_{i,j}^{(o)},o\in\{HL,LH,HH\}}$ is the update reference vector containing the detail signals.
- $\forall o \in \{HL,LH,HH\}, \quad \mathbf{x}_{0,j}^{(o)}(m,n) = \left(x_{0,j}(m-k,n-l)\right)^\top_{(k,l)\in\mathcal{P}_{0,j}^{(o)}}$
- $\forall o \in \{HL,LH,HH\}, \quad \mathbf{D}_{0,j+1}^{(o)}(m,n) = \left(\mathbf{x}_{j+1}(m-k,n-l)\right)^\top_{(k,l)\in\mathcal{P}_{0,j}^{(o)}}$
- $\mathbf{D}_{1,j+1}^{(HH)}(m,n) = \left(\mathbf{D}_{0,j+1}^{(HL)}(m-k,n-l)\right)_{(k,l)\in\mathcal{P}_{1,j}^{(HH)}}$
- $\mathbf{D}_{2,j+1}^{(HH)}(m,n) = \left(\mathbf{D}_{0,j+1}^{(LH)}(m-k,n-l)\right)_{(k,l)\in\mathcal{P}_{2,j}^{(HH)}}$

Once the different terms involved in the criterion (6.17) are defined by Eq. (6.18)-(6.21), the optimal update operator can be deduced by minimizing (6.17). Due to the complexity of the resulting linear system of equations when a NSLS is considered, its expression will not be provided.

However, when examining the expressions of $\left(\hat{x}_{i,j}(m,n)\right)_{i\in\{0,1,2,3\}}$, it can be noticed that the minimization of the criterion (6.17) requires the determination of many terms (Eq. (6.18)-(6.21)) which will increase the computational effort of this optimization method. The complexity of this method will be compared with the proposed one in Section 6.6.

To reduce the complexity of the previous method, we propose to employ an optimization criterion, which aims at reducing the inherent aliasing artefacts [Kaaniche *et al.*, 2010]. Firstly, it can be noticed that the transfer function of the low-pass filter relating $x_j$ to the undecimated version of $x_{j+1}$ can be expressed as

$$F_j(z_1,z_2) = 1 + \sum_{o\in\{HL,LH,HH\}} \sum_{(k,l)\in\mathcal{U}_j^{(o)}} u_j^{(o)}(k,l) z_1^{-2k} z_2^{-2l} A_j^{(o)}(z_1,z_2) \tag{6.22}$$

where $A_j^{(o)}(z_1,z_2)$ is a transfer function which depends on the prediction coefficients. Since $F_j$ aims at computing a smooth version of the image $x_j$, we propose to design the update filter in order to reduce the aliasing effects. More precisely, we adopt a new criterion $\tilde{\mathcal{J}}$ which accounts for the difference between the output of the $F_j$ filter and the output of an ideal low-pass filter

$$\tilde{\mathcal{J}}(u_j) = \int_{-\pi}^{\pi} \left| F_j(e^{\imath\omega_1}, e^{\imath\omega_2}) - H(e^{\imath\omega_1}, e^{\imath\omega_2}) \right|^2 S_{x_j}(\omega_1,\omega_2) d\omega_1 d\omega_2 \tag{6.23}$$

where $S_{x_j}$ is the power spectrum density of $x_j$ and $H$ denotes the transfer function of the ideal filter ($h$ will subsequently designate its impulse response). Note that $\tilde{\mathcal{J}}$ is not limited to a simple quadratic mean square error between the update filter and the ideal low-pass filter: the proposed criterion exploits the characteristics of the input signals, so making the method contents adaptive. By cancelling the gradient of $\tilde{\mathcal{J}}$, the optimal update weights satisfy, for each $(k',l') \in \mathcal{U}_j^{(o')}$ and $o' \in \{HL, LH, HH\}$, the following condition:

$$\begin{aligned}
&\sum_{o\in\{HL,LH,HH\}} \sum_{(k,l)\in\mathcal{U}_j^{(o)}} u_j^{(o)}(k,l) \sum_{(p,q)} \sum_{(p',q')} a_j^{(o)}(p,q) a_j^{(o')}(p',q') \\
&\qquad\qquad \times R_{x_j}(p'-p+2k'-2k, q'-q+2l'-2l) \\
&= \sum_{(p,q)} \sum_{(p',q')} h(p,q) a_j^{(o')}(p',q') R_{x_j}(p'-p+2k', q'-q+2l') \\
&\qquad\qquad - \sum_{(p',q')} a_j^{(o')}(p',q') R_{x_j}(p'+2k', q'+2l').
\end{aligned} \tag{6.24}$$

It must be emphasized that this criterion $\tilde{\mathcal{J}}$ measures the quadratic error between the output of the low-pass filter $F_j$ and the ideal one $H$ before decimation. However, it may also be interesting to minimize the error after the decimation step. For this purpose, we propose to use an alternative criterion which measures the error between the decimated

versions:

$$\begin{aligned}\mathcal{J}(u_j) &= \mathsf{E}\Big[\Big(x_{j+1}(m,n) - y_{j+1}(m,n)\Big)^2\Big] \\ &= \mathsf{E}\Big[\Big(x_{0,j}(m,n) + \sum_{o\in\{HL,LH,HH\}} \sum_{(k,l)\in\mathcal{U}_j^{(o)}} u_j^{(o)}(k,l)x_{j+1}^{(o)}(m-k,n-l) \\ &\quad - y_{j+1}(m,n)\Big)^2\Big] \end{aligned} \tag{6.25}$$

where

$$y_{j+1}(m,n) = \tilde{y}_j(2m,2n) = (h * x_j)(2m,2n).$$

Recall that the impulse response of the 2D ideal rectangular low-pass filter is defined in the spatial domain by:

$$\forall (m,n)\in\mathbb{Z}^2, \quad h(m,n) = \frac{1}{4}\mathrm{sinc}\left(\frac{m\pi}{2}\right)\mathrm{sinc}\left(\frac{n\pi}{2}\right). \tag{6.26}$$

The optimal update coefficients minimizing the second criterion $\mathcal{J}$ verify, for each $(k',l')\in\mathcal{U}_j^{(o')}$ and $o'\in\{HL,LH,HH\}$, the following equation:

$$\begin{aligned}&\sum_{o\in\{HL,LH,HH\}} \sum_{(k,l)\in\mathcal{U}_j^{(o)}} u_j^{(o)}(k,l)\mathsf{E}[x_{j+1}^{(o)}(m-k,n-l)x_{j+1}^{(o')}(m-k',n-l')] \\ &= \mathsf{E}[y_{j+1}(m,n)x_{j+1}^{(o')}(m-k',n-l')] - \mathsf{E}[x_{0,j}(m,n)x_{j+1}^{(o')}(m-k',n-l')]. \end{aligned} \tag{6.27}$$

It is worth pointing out that the resulting solution (6.27) is equivalent to the one obtained by minimizing $\tilde{\mathcal{J}}$ (6.24). Indeed, by using (6.4) and (6.22), it can be checked that:

$$x_{j+1}^{(o)}(m,n) = \sum_{(p,q)} a_j^{(o)}(p,q)x_j(2m-p,2n-q) \tag{6.28}$$

where $a_j^{(o)}(p,q)$ are the coefficients of the transfer function $A_j^{(o)}(z_1,z_2)$. Thus, by substituting this expression into (6.27), we deduce that the minimizations of criteria $\tilde{\mathcal{J}}$ and $\mathcal{J}$ lead to the same optimal solution.

As computing an error between the decimated versions of the image allows us to reduce the complexity of our optimization method, we propose to use the second criterion $\mathcal{J}$. In this case, it can be noticed that (6.27) can be rewritten concisely as follows:

$$\mathsf{E}[\mathbf{x}_{j+1}(m,n)\mathbf{x}_{j+1}(m,n)^\top]\mathbf{u}_j = \mathsf{E}[y_{j+1}(m,n)\mathbf{x}_{j+1}(m,n)] - \mathsf{E}[x_{0,j}(m,n)\mathbf{x}_{j+1}(m,n)] \tag{6.29}$$

where

- $\mathbf{u}_j = \left(u_j^{(o)}(k,l)\right)^\top_{(k,l)\in\mathcal{U}_j^{(o)},o\in\{HL,LH,HH\}}$ is the update weight vector,
- $\mathbf{x}_{j+1}(m,n) = \left(x_{j+1}^{(o)}(m-k,n-l)\right)^\top_{(k,l)\in\mathcal{P}_{i,j}^{(o)},o\in\{HL,LH,HH\}}$ is the update reference vector containing the detail signals.

Consequently, the optimal update weights are solutions of a *linear* system of equations given the second order characteristics of the involved signals. More specifically, in order to solve (6.29), we need to compute

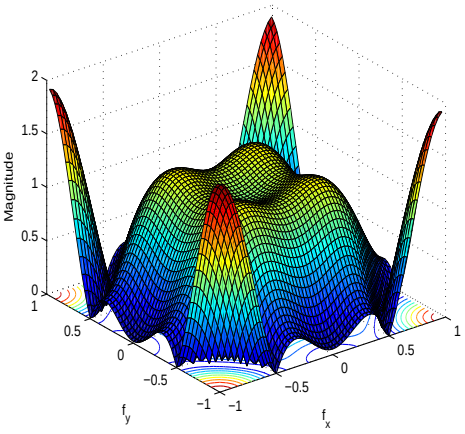
- the autocorrelation matrix $\mathbf{\Gamma}_j = \mathsf{E}[\mathbf{x}_{j+1}(m,n)\mathbf{x}_{j+1}(m,n)^\top]$,
- the cross-correlation vector $\mathbf{c}_{x_j} = \mathsf{E}[x_{0,j}(m,n)\mathbf{x}_{j+1}(m,n)]$ and,
- the cross-correlation vector $\mathbf{c}_{x_j\tilde{y}_j} = \mathsf{E}[y_{j+1}(m,n)\mathbf{x}_{j+1}(m,n)]$.

It is important to note that (6.16) and (6.29) can be solved if on the one hand, the NSLS decomposition is explicitly defined through the reference prediction and update vectors and, on the other hand, the autocorrelation function $R_{x_j}$ and cross-correlation one $R_{x_j\tilde{y}_j}$ are known at each resolution level $j$. Figures 6.2 and 6.3 display the magnitude plots of the frequency responses of the low-pass filter $F_0$ for the 5/3 wavelet transform obtained before and after each optimization step applied respectively on a natural image ("castle") and a residual one obtained from the "spot5-3" stereo image.
It is clearly seen that the optimization of the update filter yields an optimal low-pass filter which better attenuates the high frequency components than that obtained without the update optimization step. Furthermore, it is worth pointing out that the optimal low-pass filters corresponding respectively to the texture image and the residual one are different since the update filter depends on the contents of the input image to be encoded.

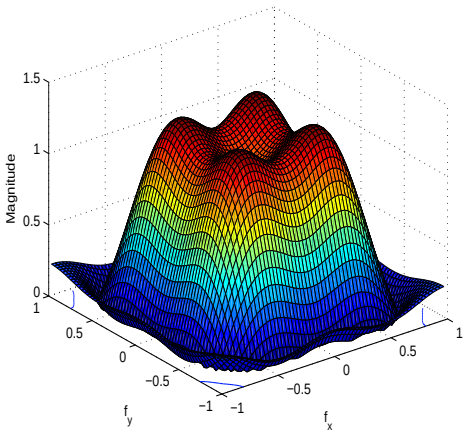
## 6.4 Theoretical analysis

In this part, we perform a statistical analysis of our method in terms of the adaptation criterion used in the optimization of the update filter. Firstly, we give explicit expressions of the optimal update coefficients for a given class of input signals. Then, we confirm that optimizing the update filter leads to lower values of criterion $\mathcal{J}$ than the ones obtained with non-optimized coefficients.

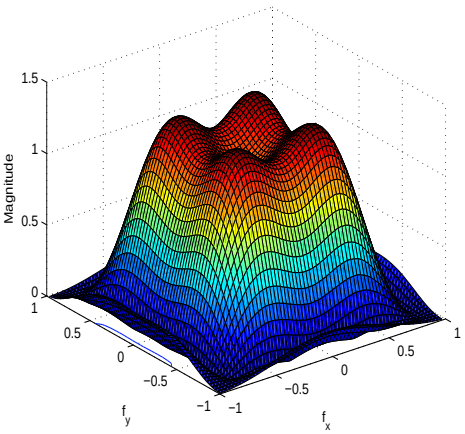
(a): “castle” image




(b)




(c)

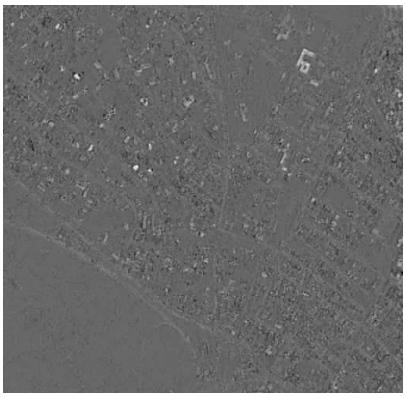



(d)

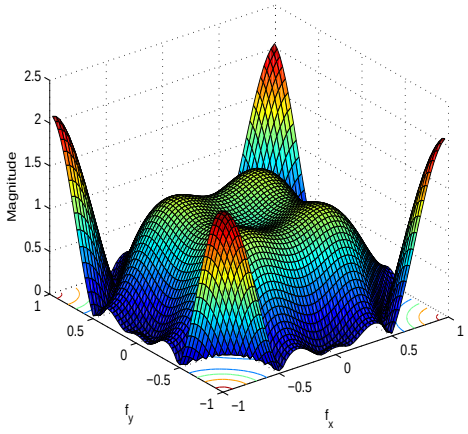
Figure 6.2: Frequency responses of the low-pass filter $F_0$ when the prediction filters are optimized by minimizing the variance of the detail coefficients, and (b) the update filter is not optimized, (c) the update filter is optimized using Gouze's method (d) the update filter is optimized using the proposed method.

### 6.4.1 Notations

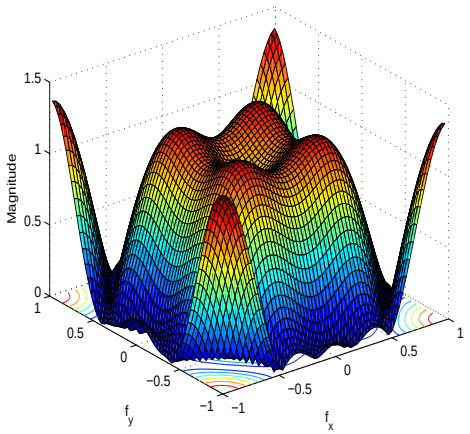
In order to study the theoretical performance of the optimization method, we will consider the optimization of the 2D non separable 5/3 transform whose underlying operators are given by (6.7)-(6.10). This amounts to choosing the following spatial supports for the



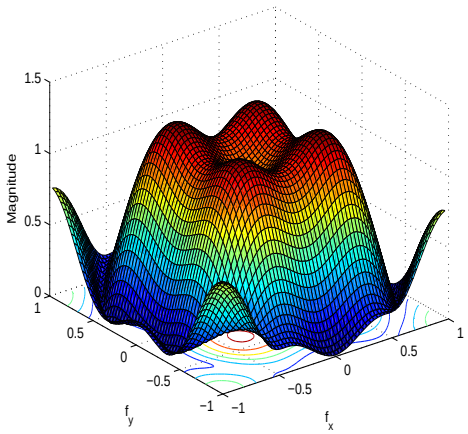
(a): “SPOT5-3” residual image




(b)




(c)




(d)

Figure 6.3: Frequency responses of the low-pass filter $F_0$ when the prediction filters are optimized by minimizing the variance of the detail coefficients, and (b) the update filter is not optimized, (c) the update filter is optimized using Gouze's method (d) the update filter is optimized using the proposed method.

prediction and update filters:

$$\mathcal{P}_{0,j}^{(HH)} = \{(0,0),(0,-1),(-1,0),(-1,-1)\},$$

$$\mathcal{P}_{1,j}^{(HH)} = \{(0,0),(-1,0)\}, \mathcal{P}_{2,j}^{(HH)} = \{(0,0),(0,-1)\}, \tag{6.30}$$

$$\mathcal{P}_{0,j}^{(LH)} = \{(0,0),(-1,0)\}, \mathcal{P}_{1,j}^{(LH)} = \{(0,0),(0,1)\}, \tag{6.31}$$

$$\mathcal{P}_{0,j}^{(HL)} = \{(0,0),(0,-1)\}, \mathcal{P}_{1,j}^{(HL)} = \{(0,0),(1,0)\}, \tag{6.32}$$

$$\mathcal{U}_{j}^{(HL)} = \{(0,0),(0,1)\}, \mathcal{U}_{j}^{(LH)} = \{(0,0),(1,0)\}, \tag{6.33}$$

$$\mathcal{U}_{j}^{(HH)} = \{(0,0),(0,1),(1,0),(1,1)\}. \tag{6.34}$$

To obtain tractable expressions, we assume that, at a given resolution level $j$, the approximation image $x_j$ is modeled as the realization of a first order 2D autoregressive process, the auto-correlation function of which is given by:

$$\forall (k,l) \in \mathbb{Z}^2, \quad R_{x_j}(k,l) = \mathsf{E}[x_j(m,n)x_j(m-k,n-l)] = \sigma_j^2 \rho_{1,j}^{|k|} \rho_{2,j}^{|l|} \tag{6.35}$$

where $\sigma_j \in \mathbb{R}_+^*$, and $\rho_{1,j}$ and $\rho_{2,j} \in [-1,1]$ are the correlation factors along the vertical and horizontal directions.

Although the considered autoregressive model is separable, it is worth noticing that such a model has been widely used to represent a large class of textured images [Edward *et al.*, 1979]. Furthermore, it is convenient to derive simple theoretical expressions of the filter coefficients.

### 6.4.2 Optimal prediction coefficients

Concerning the prediction filter $\mathbf{p}_j^{(HH)}$, its optimal coefficients are obtained by solving (6.16) as described at the beginning of Section 6.3. Thus, once the autocorrelation matrix $\mathsf{E}[\tilde{\mathbf{x}}_j^{(HH)}(m,n)\tilde{\mathbf{x}}_j^{(HH)}(m,n)^\top]$ and the cross-correlation vector $\mathsf{E}[x_{3,j}(m,n)\tilde{\mathbf{x}}_j^{(HH)}(m,n)]$ are calculated using (6.35), the optimal weights of $\mathbf{p}_j^{(HH)}$ are deduced as follows:

$$\begin{aligned}
&p_{0,j}^{(HH)}(0,0) = p_{0,j}^{(HH)}(0,-1) = p_{0,j}^{(HH)}(-1,0) = p_{0,j}^{(HH)}(-1,-1) = -\frac{\rho_{1,j}\rho_{2,j}}{(1+\rho_{1,j}^2)(1+\rho_{2,j}^2)} \\
&p_{1,j}^{(HH)}(0,0) = p_{1,j}^{(HH)}(-1,0) = \frac{\rho_{1,j}}{1+\rho_{1,j}^2} \\
&p_{2,j}^{(HH)}(0,0) = p_{2,j}^{(HH)}(0,-1) = \frac{\rho_{2,j}}{1+\rho_{2,j}^2}.
\end{aligned} \tag{6.36}$$

By applying the same optimization method to the other prediction filters $\mathbf{p}_j^{(LH)}$ and $\mathbf{p}_j^{(HL)}$, we get that their optimal weights are given by:

$$\begin{aligned}
&p_{0,j}^{(HL)}(0,0) = p_{0,j}^{(HL)}(0,-1) = \frac{\rho_{2,j}}{1+\rho_{2,j}^2} \\
&p_{0,j}^{(LH)}(0,0) = p_{0,j}^{(LH)}(-1,0) = \frac{\rho_{1,j}}{1+\rho_{1,j}^2} \\
&p_{1,j}^{(HL)}(0,0) = p_{1,j}^{(HL)}(1,0) = p_{1,j}^{(LH)}(0,0) = p_{1,j}^{(LH)}(0,1) = 0.
\end{aligned} \tag{6.37}$$

Once the optimal prediction filters are determined, we have to find the optimal weights for the update filter.

### 6.4.3 Optimal update coefficients

As mentioned above, the optimal update coefficients are optimized by solving (6.29) where the reference vector $\mathbf{x}_{j+1}(m,n)$ is defined as:

$$\mathbf{x}_{j+1}(m,n) = \big(x_{j+1}^{(HL)}(m,n), x_{j+1}^{(HL)}(m,n-1), x_{j+1}^{(LH)}(m,n), x_{j+1}^{(LH)}(m-1,n), \\ x_{j+1}^{(HH)}(m,n), x_{j+1}^{(HH)}(m,n-1), x_{j+1}^{(HH)}(m-1,n), x_{j+1}^{(HH)}(m-1,n-1)\big)^{\top}. \tag{6.38}$$

The components of $\mathbf{\Gamma}_j$ and $\mathbf{c}_{x_j}$ can be easily found by using (6.35). To derive the expression of $\mathbf{c}_{x_j\tilde{y}_j}$, we have to calculate the cross-correlation between the input signal $x_j$ and the output of the ideal low-pass filter $\tilde{y}_j$:

$$R_{x_j\tilde{y}_j}(k,l) = \mathsf{E}[\tilde{y}_j(m,n)x_j(m-k,n-l)] = \sum_{p=-\infty}^{p=+\infty}\sum_{q=-\infty}^{q=+\infty} h(p,q)R_{x_j}(k-p,l-q) \\ = \sigma_j^2 \sum_{p=-\infty}^{p=\infty}\sum_{q=-\infty}^{q=+\infty} h(p,q)\rho_{1,j}^{|k-p|}\rho_{2,j}^{|l-q|}. \tag{6.39}$$

At this point, from (6.26), we note that $h(p,q)$ has a separable form in $p$ and $q$:

$$h(p,q) = h(p)h(q) \tag{6.40}$$

where

$$h(p) = \frac{1}{2}\mathrm{sinc}(\frac{p\pi}{2}) = \begin{cases} \frac{1}{2} & \text{if } p=0 \\ 0 & \text{if } p \text{ is even} \\ \frac{(-1)^{(p-1)/2}}{p\pi} & \text{if } p \text{ is odd.} \end{cases} \tag{6.41}$$

As a result, the expression of $R_{x_j\tilde{y}_j}(k,l)$ can be rewritten in a separable way as:

$$R_{x_j\tilde{y}_j}(k,l) = \sigma_j^2 R^1_{x_j\tilde{y}_j}(k)R^2_{x_j\tilde{y}_j}(l) \tag{6.42}$$

where

$$R^1_{x_j\tilde{y}_j}(k) = \sum_{p=-\infty}^{p=+\infty} h(p)\rho_{1,j}^{|k-p|}, \tag{6.43}$$

$$R^2_{x_j\tilde{y}_j}(l) = \sum_{q=-\infty}^{q=+\infty} h(q)\rho_{2,j}^{|l-q|}. \tag{6.44}$$

In addition, from (6.43) and (6.41), we deduce that:

$$R^1_{x_j\tilde{y}_j}(0) = \frac{1}{2} + \frac{2}{\pi}\text{arctg}(\rho_{1,j}),$$

$$R^1_{x_j\tilde{y}_j}(-1) = \frac{\rho_{1,j}}{2} + \Big(\frac{\rho_{1,j}^{-1} + \rho_{1,j}}{\pi}\Big)\text{arctg}(\rho_{1,j}),$$

$$R^{1D}_{x_j\tilde{y}_j}(2k) = \begin{cases} \frac{\rho_{1,j}^{2k}}{2} + \frac{\rho_{1,j}^{2k}}{\pi}\Big(\text{arctg}(\rho_{1,j}) + \sum_{p=-k}^{-1}\frac{(-1)^p\rho_{1,j}^{2p+1}}{2p+1}\Big) \\ + \frac{\rho_{1,j}^{-2k}}{\pi}\Big(\text{arctg}(\rho_{1,j}) - \sum_{p=0}^{k-1}\frac{(-1)^p\rho_{1,j}^{2p+1}}{2p+1}\Big) & \text{if } k > 0, \\ \frac{\rho_{1,j}^{-2k}}{2} + \frac{\rho_{1,j}^{2k}}{\pi}\Big(\text{arctg}(\rho_{1,j}) - \sum_{p=0}^{-k-1}\frac{(-1)^p\rho_{1,j}^{2p+1}}{2p+1}\Big) \\ + \frac{\rho_{1,j}^{-2k}}{\pi}\Big(\text{arctg}(\rho_{1,j}) + \sum_{p=k}^{-1}\frac{(-1)^p\rho_{1,j}^{2p+1}}{2p+1}\Big) & \text{if } k < 0, \end{cases}$$

$$R^1_{x_j\tilde{y}_j}(2k+1) = \begin{cases} \frac{\rho_{1,j}^{2k+1}}{2} + \frac{\rho_{1,j}^{2k+1}}{\pi}\Big(\text{arctg}(\rho_{1,j}) + \sum_{p=-k-1}^{-1}\frac{(-1)^p\rho_{1,j}^{2p+1}}{2p+1}\Big) \\ + \frac{\rho_{1,j}^{-(2k+1)}}{\pi}\Big(\text{arctg}(\rho_{1,j}) - \sum_{p=0}^{-k}\frac{(-1)^p\rho_{1,j}^{2p+1}}{2p+1}\Big) & \text{if } k > -1, \\ \frac{\rho_{1,j}^{-(2k+1)}}{2} + \frac{\rho_{1,j}^{2k+1}}{\pi}\Big(\text{arctg}(\rho_{1,j}) - \sum_{p=0}^{-k-2}\frac{(-1)^p\rho_{1,j}^{2p+1}}{2p+1}\Big) \\ + \frac{\rho_j^{-(2k+1)}}{\pi}\Big(\text{arctg}(\rho_{1,j}) + \sum_{p=k+1}^{-1}\frac{(-1)^p\rho_{1,j}^{2p+1}}{2p+1}\Big) & \text{if } k < -1. \end{cases}$$

The expressions of $R^2_{x_j\tilde{y}_j}(l)$ are easily derived by replacing $\rho_{1,j}$ by $\rho_{2,j}$ in the above equalities. Altogether, we obtain $R_{x_j\tilde{y}_j}(k,l)$ by using (6.42).

Once $\mathbf{\Gamma}_j$, $\mathbf{c}_{x_j}$ and $\mathbf{c}_{x_j\tilde{y}_j}$ are determined, after some calculations, the optimal update coefficients are simply given by:

$$\begin{aligned} u_j^{(HL)}(0,0) = u_j^{(HL)}(0,1) &= \frac{\pi + 4\,\text{arctg}(\rho_{1,j})}{2\pi^2} \\ u_j^{(LH)}(0,0) = u_j^{(LH)}(1,0) &= \frac{\pi + 4\,\text{arctg}(\rho_{2,j})}{2\pi^2} \\ u_j^{(HH)}(0,0) = u_j^{(HH)}(0,1) = u_j^{(HH)}(1,0) = u_j^{(HH)}(1,1) &= \frac{1}{\pi^2}. \end{aligned} \tag{6.45}$$

Note that, although the simple separable autoregressive modeling which was adopted induces a separable form of the optimized prediction and update filters, the resulting NSLS cannot be put under the classical form of a separable lifting structure. This can be justified by the fact that the vector $\mathbf{u}_j^{(HH)}$ is not equal to the Kronecker product of $\mathbf{u}_j^{(HL)}$ and $\mathbf{u}_j^{(LH)}$. Indeed, the necessary conditions for satisfying the equivalence between a non separable scheme and a separable one are given at the end of Section 6.2.

### 6.4.4 Adaptation criterion values

In order to show the interest of our optimization approach, we aim at comparing the values of the criterion $\mathcal{J}$ in the two following cases:

- Case I: the update vector $\mathbf{u}_j^{\text{nonopt}}$ has non optimized components corresponding to the conventional 5/3 transform, given by (6.10):
  $\mathbf{u}_j^{\text{nonopt}} = [1/4, 1/4, 1/4, 1/4, -1/16, -1/16, -1/16, -1/16]$.
- Case II: the update vector $\mathbf{u}_j^{\text{opt}}$ corresponds to weights minimizing $\mathcal{J}$.

Our objective is to evaluate the gain resulting from the optimization of $\mathcal{J}$ by calculating $\mathcal{J}(\mathbf{u}_j^{\text{nonopt}})$ and $\mathcal{J}(\mathbf{u}_j^{\text{opt}})$. To this respect, we recall that the optimal value of $\mathcal{J}$ can be expressed as:

$$\begin{aligned}\mathcal{J}(\mathbf{u}_j^{\text{opt}}) =& \mathsf{E}\big[x_{0,j}(m,n)^2\big] + \mathsf{E}\big[y_{j+1}(m,n)^2\big] - 2\mathsf{E}\big[x_{0,j}(m,n)y_{j+1}(m,n)\big] \\ &+ (\mathbf{u}_j^{\text{opt}})^\top \mathsf{E}\big[x_{0,j}(m,n)\mathbf{x}_{j+1}(m,n)\big] - (\mathbf{u}_j^{\text{opt}})^\top \mathsf{E}\big[y_{j+1}(m,n)\mathbf{x}_{j+1}(m,n)\big]. \quad (6.46)\end{aligned}$$

Eq. (6.46) indicates that the evaluation of $\mathcal{J}$ requires to find the expressions of $R_{x_j}(k,l)$, $R_{x_j\tilde{y}_j}(k,l)$, and the value $R_{y_{j+1}}(0,0)$ of the autocorrelation of $y_{j+1}$ at $(0,0)$. The first two expressions are readily given by (6.35) and (6.45) whereas the third one is easily derived:

$$R_{y_{j+1}}(0,0) = \sigma_j^2\Big(\frac{1}{2} + \frac{2}{\pi}\text{arctg}(\rho_{1,j})\Big)^2\Big(\frac{1}{2} + \frac{2}{\pi}\text{arctg}(\rho_{2,j})\Big)^2. \quad (6.47)$$

It is worth pointing out that the calculation of the adaptation criterion has not led to a simple expression as a function of $\rho_{1,j}$ and $\rho_{2,j}$. In order to obtain more easily interpretable results, we propose to consider the case of an approximation image that presents strong correlations in the horizontal and vertical directions. In other words, we focus on the case when the couple $(\rho_{1,j}, \rho_{2,j})$ is close to (1,1). Hence, we propose to approximate $\mathcal{J}$ by its second-order Taylor expansion $\widehat{\mathcal{J}}$ when $(\rho_{1,j}, \rho_{2,j})$ is around (1,1):

$$\begin{aligned}\widehat{\mathcal{J}}\Big(\mathbf{u}_j^{\text{nonopt}}\Big) \simeq& \frac{\sigma_j^2}{2}\Big[\frac{1}{4}(2-\rho_{1,j}-\rho_{2,j}) + \frac{1}{8}(1-\rho_{1,j})^2 + \frac{1}{8}(1-\rho_{2,j})^2 \\ &+ (\frac{3}{\pi^2} + \frac{1}{32})(1-\rho_{1,j})(1-\rho_{2,j})\Big] \\ \widehat{\mathcal{J}}\Big(\mathbf{u}_j^{\text{opt}}\Big) \simeq& \frac{\sigma_j^2}{\pi^2}\Big[(\pi-2)(2-\rho_{1,j}-\rho_{2,j}) + \frac{\pi-2}{2}(1-\rho_{1,j})^2 \\ &+ \frac{\pi-2}{2}(1-\rho_{2,j})^2 + \frac{\pi^2-8\pi+4}{\pi^2}(1-\rho_{1,j})(1-\rho_{2,j})\Big]. \quad (6.48)\end{aligned}$$

### 6.4.5 Discussion

In order to emphasize the usefulness of the optimization of the update filter, we illustrate in Fig. 6.4 the variations of the criterion in (6.48) with respect to $\rho_j$, where we have assumed that $\rho_{1,j} = \rho_{2,j} = \rho_j$.

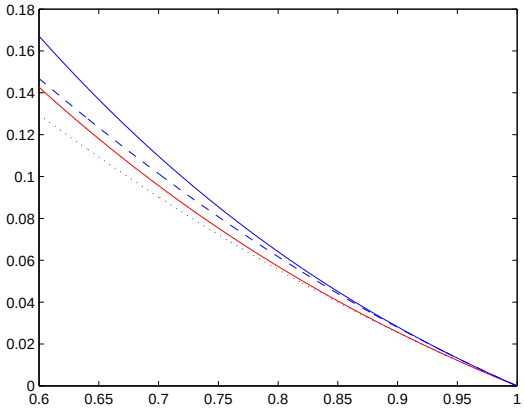



Figure 6.4: Variations of the adaptation criterion w.r.t $\rho_j$: $\mathcal{J}(\mathbf{u}^{\text{nonopt}})$ in solid blue line, $\mathcal{J}(\mathbf{u}^{\text{opt}})$ in solid red line, $\widehat{\mathcal{J}}(\mathbf{u}^{\text{nonopt}})$ in dotted blue line, $\widehat{\mathcal{J}}(\mathbf{u}^{\text{opt}})$ in dotted red line.

At first, Fig. 6.4 shows that the Taylor expansion closely approximates the criterion when $\rho_j \geq 0.85$. Note that in practice, for natural images, the values of $\rho_j$ are within such a range [Chabrier *et al.*, 2005]. Besides, by adapting the update filter coefficients to the image statistical characteristics, we can see that the optimized scheme leads to an improvement compared with the non-optimized one. In summary, the interest of this theoretical study is twofold. First, it allows us to better understand the benefits that can be drawn from the proposed optimization method. Furthermore, it provides closed form expressions of the prediction and update filters which can be used in order to simplify the implementation of the proposed optimization method.

## 6.5 Transmission cost of the filter coefficients

The entire set of optimal weights for the update and the three prediction filters corresponds to an amount of $o_{p,u} = (L_p + L_u)\, J$ floating point coefficients, where $L_p$ and $L_u$ denote respectively the number of prediction and update weights in the adaptive NSLS and $J$ represents the number of resolution levels. For an image of size $N_1 \times N_2$, the transmission cost of the filters coefficients will increase the bitrate achieved by the proposed optimization method, by $\frac{32\, o_{p,u}}{N_1 N_2}$ bits per pixel. For example, if we consider the supports of the prediction and update filters corresponding to the 2D 5/3 transform with $L_p = 16$ and $L_u = 8$ (see (6.7)-(6.10)) , when $N_1 = N_2 = 512$, $J = 2$ and if the weights are stored on 32 bits, the bitrate will be increased by 0.0059 bpp which is a very small fraction of the overall data

bitrate. However, if we use the explicit expressions of the filter coefficients given by the theoretical analysis, it is enough to send only the correlation factors $\rho_{1,j}$ and $\rho_{2,j}$. This leads to a significant reduction of the transmission cost of the filter coefficients, which becomes 0.0004 bpp.

## 6.6 Experimental results

Simulations were carried out on two kinds of still images originally quantized on 8 bpp which are either single views[2] or stereoscopic ones[3]. The gain related to the optimization of the NSLS operators was evaluated in these contexts.

Since we are mainly interested in lossy-to-lossless coding schemes, we will consider the integer-to-integer 5/3 transform recommended for the lossless mode of JPEG2000 [Taubman, Marcellin, 2001]. In order to show the benefits of the optimization of *all* the involved operators, we provide the results for the following decompositions carried out over two resolution levels:

• The first one is the lifting scheme corresponding to the 5/3 integer transform, which is known as the (2,2) wavelet transform [Calderbank *et al.*, 1998]. Its underlying decomposition operators are given by (6.7)-(6.10). In the following, this method will be designated by NSLS(2,2).

• The second variant consists of optimizing only the *prediction* filters of the NSLS(2,2) while the update filter is kept unchanged. Note that the prediction support is preserved. This method will be denoted by NSLS(2,2)-OPT1.

• The third method consists of optimizing *both* the prediction and update operators. In what follows, our proposed method will be designated by NSLS(2,2)-OPT2-PM1 (resp. NSLS(2,2)-OPT2-PM2) when using the experimental (resp. theoretical) optimal update coefficients.

• We have also tested for comparison the update optimization method of Gouze *et al.* which aims at designing the update filter in order to minimize a reconstruction error [Pesquet-Popescu, 1999; Gouze *et al.*, 2004]. This optimization method will be designated by NSLS(2,2)-OPT2-GM.

As mentioned earlier, the proposed method for the design of all filters can be applied

[2] taken from the URL http://sipi.usc.edu/database

[3] http://vasc.ri.cmu.edu/idb/html/stereo/index.html

to any classical P-U lifting structure. To illustrate the performance of the optimization method when using longer filters, we will consider a simple example corresponding to the (4,2) integer wavelet transform. Indeed, after applying this transform to the lines then to the columns, it is easy to show that the corresponding support of the two-dimensional prediction and update filters are given by:

$$\mathcal{P}_{0,j}^{(HH)} = \{(k,l) \mid -2 \leq k \leq 1, -2 \leq l \leq 1\}, \mathcal{P}_{1,j}^{(HH)} = \{(k,0) \mid -2 \leq k \leq 1\},$$
$$\mathcal{P}_{2,j}^{(HH)} = \{(0,l) \mid -2 \leq l \leq 1\}, \tag{6.49}$$
$$\mathcal{P}_{0,j}^{(LH)} = \{(k,l) \mid -2 \leq k \leq 1, -2 \leq l \leq 1\}, \mathcal{P}_{1,j}^{(LH)} = \{(0,0),(0,1)\}, \tag{6.50}$$
$$\mathcal{P}_{0,j}^{(HL)} = \{(k,l) \mid -2 \leq k \leq 1, -2 \leq l \leq 1\}, \mathcal{P}_{1,j}^{(HL)} = \{(0,0),(1,0)\}, \tag{6.51}$$
$$\mathcal{U}_{j}^{(0)} = \{(0,0),(0,1)\}, \mathcal{U}_{j}^{(1)} = \{(0,0),(1,0)\}, \mathcal{U}_{j}^{(2)} = \{(0,0),(0,1),(1,0),(1,1)\}. \tag{6.52}$$

In what follows, NSLS(4,2)-OPT1 and NSLS(4,2)-OPT2-PM1 will designate respectively the decompositions when optimizing the predict and both filters.

First of all, the performance of the proposed method has been assessed on monocular images. Starting with the evaluation of these decompositions in a progressive reconstruction context, Figures 6.5 and 6.6 give the variations of the PSNR versus the bitrate for the "lena" and "castle" images.

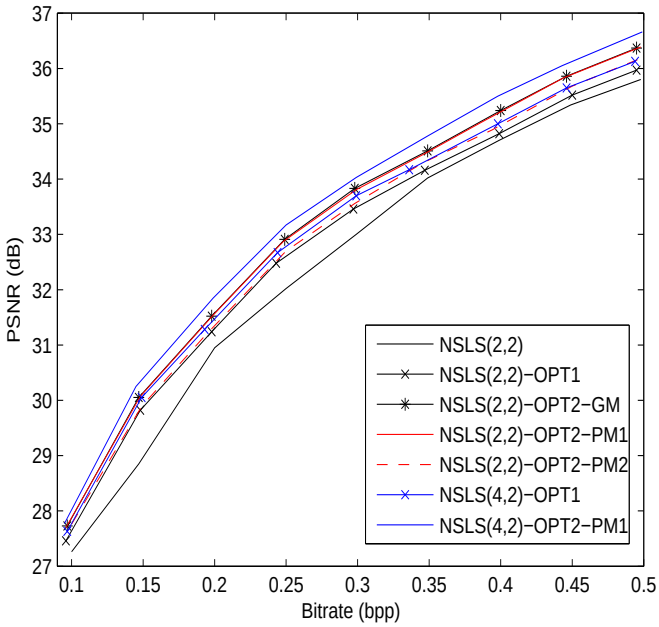



Figure 6.5: PSNR (in dB) versus the bitrate (bpp) after JPEG2000 progressive encoding for the "lena" image.

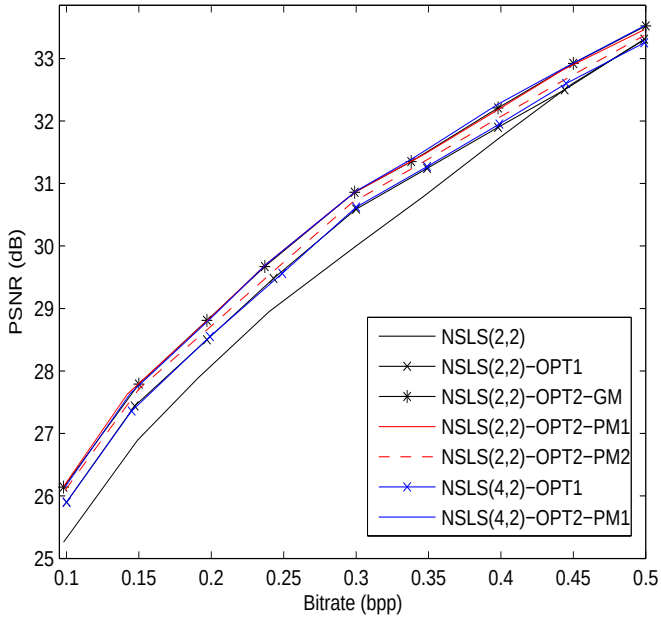



Figure 6.6: PSNR (in dB) versus the bitrate (bpp) after JPEG2000 progressive encoding for the "castle" image.

It is worth pointing out that NSLS(2,2)-OPT2-PM1 outperforms NSLS(2,2) by 0.2-0.9 dB and NSLS(2,2)-OPT1 by about 0.4 dB. It can be noticed that the proposed update optimization method NSLS(2,2)-OPT2-PM1 and Gouze's method NSLS(2,2)-OPT2-GM have similar performances in terms of quality of reconstruction. However, it is worth pointing out that our work present two advantages with respect to Gouze's work. Firstly, the proposed method reduces significantly the complexity of the optimization algorithm. Indeed, although both minimization approaches amount to solving a linear system $\mathbf{A}_j\mathbf{u}_j = \mathbf{b}_j$, it is worth noting that calculating $\mathbf{A}_j$ and $\mathbf{b}_j$ is more intricate and requires more operations in the case of Gouze's method [Pesquet-Popescu, 1999; Gouze *et al.*, 2004]. More precisely, in this case, the matrix $\mathbf{A}_j$ (resp. vector $\mathbf{b}_j$) is completely defined by the computation of *four* matrices of size $8 \times 8$ (resp. *four* vectors of dimension 8) since the reconstruction error is evaluated for each of the four polyphase components (see (6.17)). Our optimization method requires only the computation of *one* matrix of size $8 \times 8$ and *two* vectors of dimension 8 as shown in (6.29). To better illustrate this fact, we propose to compare these two optimization methods in terms of their computation time. Table 6.1 presents the computation time of a Matlab implementation of the optimization methods of the update filter for some images of size $512 \times 512$. Simulations are carried out by using an Intel Core

2 (3 GHz) computer. It can be noticed that the proposed method is three times faster than Gouze's method based on the minimization of the reconstruction error. Secondly, the main advantage of the current work is that it provides explicit expressions of the optimal filter coefficients based only on the horizontal and vertical correlation factors. In order to show the benefit of the statistical analysis we conducted, we evaluate our method when using directly the obtained theoretical filter coefficient expressions. We can see in Figures 6.5 and 6.6 that using the theoretical values of the optimal filters NSLS(2,2)-OPT2-PM2 yields results comparable with the method based on experimental filter coefficients NSLS(2,2)-OPT2-PM1. This confirms that the considered autoregressive model allows us to obtain theoretical results which are close to the experimental ones. We however noticed that the simple autoregressive model becomes inaccurate in the case of some specific images showing a strong correlation in another direction than the horizontal or vertical ones. For

Table 6.1: Computation time of the optimization method of the update filter (in seconds).

| Image | lena | einst | castle | elaine |
|---|---|---|---|---|
| Gouze's method | 3.14 | 3.11 | 3.09 | 3.11 |
| Our method | 1.06 | 1.07 | 1.08 | 1.06 |

example, Table 6.2 illustrates the optimized filter coefficients for the "castle" and "straw" images. An additional gain of 0.3 dB can be achieved by using longer filters NSLS(4,2)-OPT2-PM1. Figures 6.7, 7.7 and 6.9 display the reconstructed images of "airport" and "lena" and "elaine". The quality of these images is compared both in terms of PSNR and Structural SIMilarity (SSIM) [Wang *et al.*, 2004; Rouse, Hemami, 2008]. It can be observed that optimizing *both* the prediction and update filters significantly improves the quality of reconstruction. The difference in PSNR (resp. SSIM) ranges from 0.35 dB to 0.4 dB (resp. from 0.03 to 0.07) compared with the decompositions in which *only* the prediction filters are optimized. In addition, we have tested these methods in a lossless coding context and the related final bitrates are given in Table 6.3. Slight improvements (about 0.02 bpp compared to the Gouze's method) are obtained when our optimization method is employed in a lossless coding scheme.

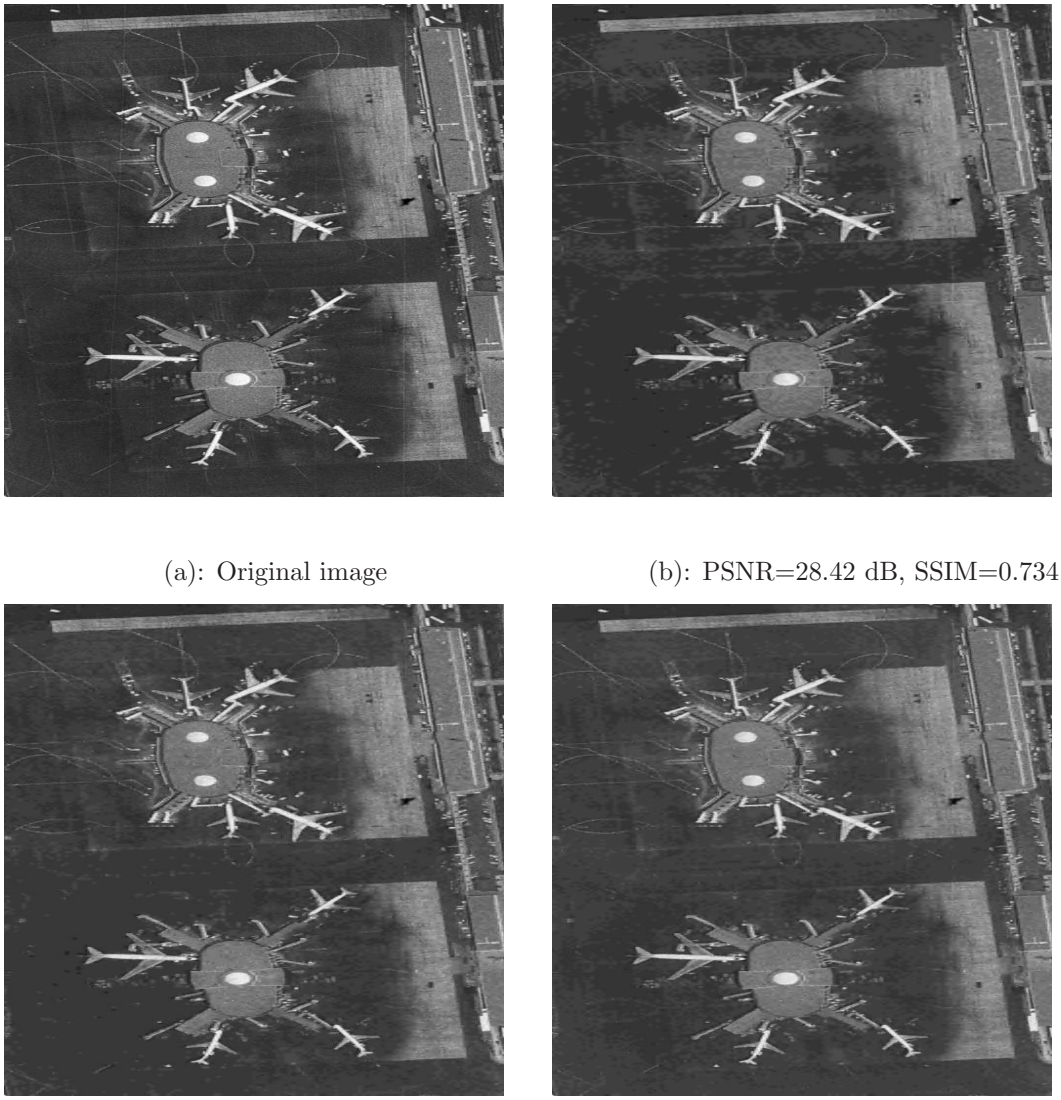
Table 6.2: Experimental and theoretical results of filter coefficients for the "castle" and "straw" images

| Image | Filter | Experimental optimized coefficients | Theoretical optimized coefficients |
|---|---|---|---|
| castle | $\mathbf{p}_0^{(HH)}$ | $[-0.2153, -0.2041, -0.2081, -0.2139, 0.4749, 0.4798, 0.4457, 0.4419]^\top$ | $[-0.2496, -0.2496, -0.2496, -0.2496, 0.4997, 0.4997, 0.4997, 0.4997]^\top$ |
| | $\mathbf{p}_0^{(HL)}$ | $[0.5041, 0.5054, -0.1192, -0.1081]^\top$ | $[0.4994, 0.4994, 0, 0]^\top$ |
| | $\mathbf{p}_0^{(LH)}$ | $[0.4983, 0.5039, -0.1494, -0.1542]^\top$ | $[0.4997, 0.4997, 0, 0]^\top$ |
| | $\mathbf{u}_0$ | $[0.2897, 0.2855, 0.2839, 0.2771, 0.0906, 0.0753, 0.0827, 0.0849]^\top$ | $[0.3150, 0.3150, 0.3132, 0.3132, 0.1013, 0.1013, 0.1013, 0.1013]^\top$ |
| straw | $\mathbf{p}_0^{(HH)}$ | $[-0.2410, 0.0855, 0.0676, -0.2505, 0.3446, 0.3360, 0.3160, 0.3289]^\top$ | $[-0.2293, -0.2293, -0.2293, -0.2293, 0.4769, 0.4769, 0.4808, 0.4808]^\top$ |
| | $\mathbf{p}_0^{(HL)}$ | $[0.5206, 0.5298, -0.0763, -0.0659]^\top$ | $[0.4808, 0.4808, 0, 0]^\top$ |
| | $\mathbf{p}_0^{(LH)}$ | $[0.5568, 0.4904, -0.1051, -0.1554]^\top$ | $[0.4769, 0.4769, 0, 0]^\top$ |
| | $\mathbf{u}_0$ | $[0.1990, 0.1881, 0.1744, 0.1949, 0.2118, -0.0928, -0.0949, 0.2110]^\top$ | $[0.2874, 0.2874, 0.2901, 0.2901, 0.1013, 0.1013, 0.1013, 0.1013]^\top$ |

Table 6.3: Performance of the lossless decompositions in terms of bitrate (bpp) using JPEG2000.

| Image | NSLS(2,2) | NSLS(2,2) OPT1 | NSLS(2,2)-OPT2-GM | NSLS(2,2)-OPT2-PM1 |
|---|---|---|---|---|
| spot5-1 | 3.94 | 3.87 | 3.88 | 3.85 |
| spot5-2 | 4.17 | 4.09 | 4.09 | 4.06 |
| spot5-3 | 4.15 | 4.07 | 4.08 | 4.04 |
| lena | 4.31 | 4.31 | 4.31 | 4.31 |
| straw | 6.35 | 6.34 | 6.34 | 6.34 |
| airport | 5.49 | 5.48 | 5.48 | 5.48 |
| earthquake | 6.48 | 6.30 | 6.33 | 6.30 |
| castle | 3.70 | 3.66 | 3.66 | 3.66 |
| Average | 4.82 | 4.76 | 4.77 | 4.75 |

The second part of the experiments is concerned with stereo images. More precisely, we also evaluate our optimization method for coding residual images obtained after disparity-



(a): Original image (b): PSNR=28.42 dB, SSIM=0.734

(c): PSNR=28.45 dB, SSIM=0.738 (d): PSNR=28.80 dB, SSIM=0.741

Figure 6.7: Zoom applied on the reconstructed "airport" image at 0.35 bpp using: (b) NSLS(2,2) (c) NSLS(2,2)-OPT1 (d) NSLS(2,2)-OPT2-PM1.

compensation between the right and the left views. Note that this stereo image coding scheme was designated by scheme B in the previous chapters. Figures 6.10 and 6.11 illustrate the evolution of the PSNR versus the bitrate of the residual image generated respectively from the "shrub" and "spot5-3" stereo image pairs. An improvement of 0.1-0.25 dB is obtained by optimizing the update filter. The gain becomes more important (up to 0.6 dB) when compared with the non-optimized NSLS(2,2) wavelet transform. Furthermore, we

(a): Original image

(b): PSNR=31.81 dB, SSIM=0.860

(c): PSNR=32.49 dB, SSIM=0.872

(d): PSNR=32.92 dB, SSIM=0.877

Figure 6.8: The reconstructed "lena" image at 0.25 bpp using: (b) NSLS(2,2) (c) NSLS(2,2)-OPT1 (d) NSLS(2,2)-OPT2-PM1.

show that using the theoretical filter coefficient expressions, NSLS(2,2)-OPT2-PM2 yields performances similar to the NSLS(2,2)-OPT2-PM1 scheme. An additional gain of 0.3 dB is achieved by using NSLS(4,2)-OPT2-PM1. Figure 6.12 displays the reconstructed images of the residual one generated from the "pentagon" stereo pair. It can be observed that optimizing both the prediction and update filters leads to an improvement of 0.22 dB (resp. 0.04) in PSNR (resp. SSIM) compared with the decomposition in which only the

(a): Original image

(b): PSNR=30.64 dB, SSIM=0.733

(c): PSNR=30.99 dB, SSIM=0.741

(d): PSNR=31.37 dB, SSIM=0.748

Figure 6.9: The reconstructed "elaine" image at 0.2 bpp using: (b) NSLS(2,2) (c) NSLS(2,2)-OPT1 (d) NSLS(2,2)-OPT2-PM1.

prediction filters are optimized.

All these results, obtained with monocular images and residual images of stereo pairs, confirm the effectiveness of our method in terms of reconstruction quality.

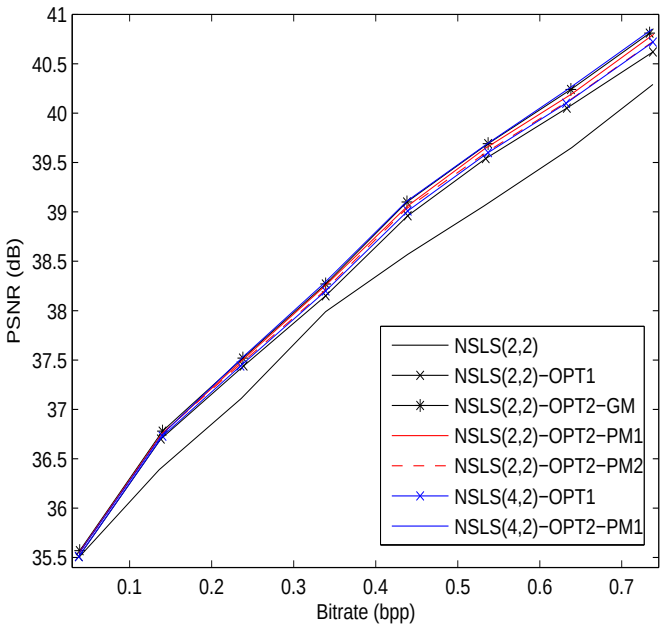



Figure 6.10: PSNR (in dB) versus the bitrate (bpp) after JPEG2000 progressive encoding for the "shrub" residual image.

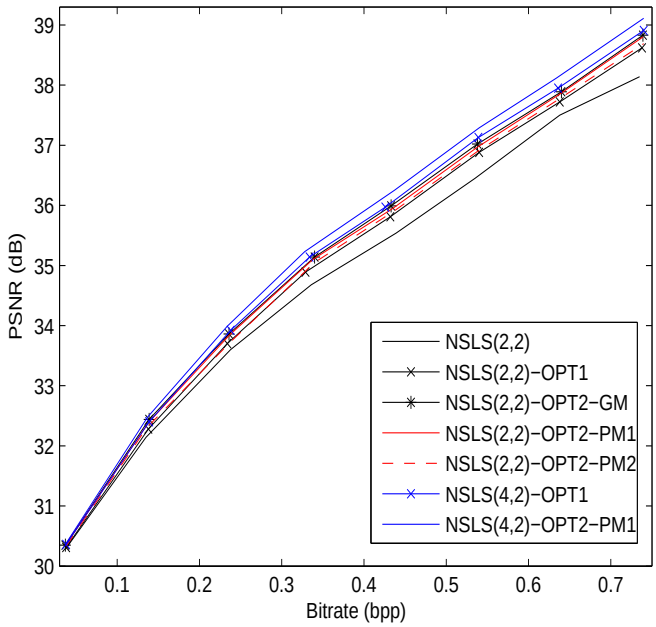
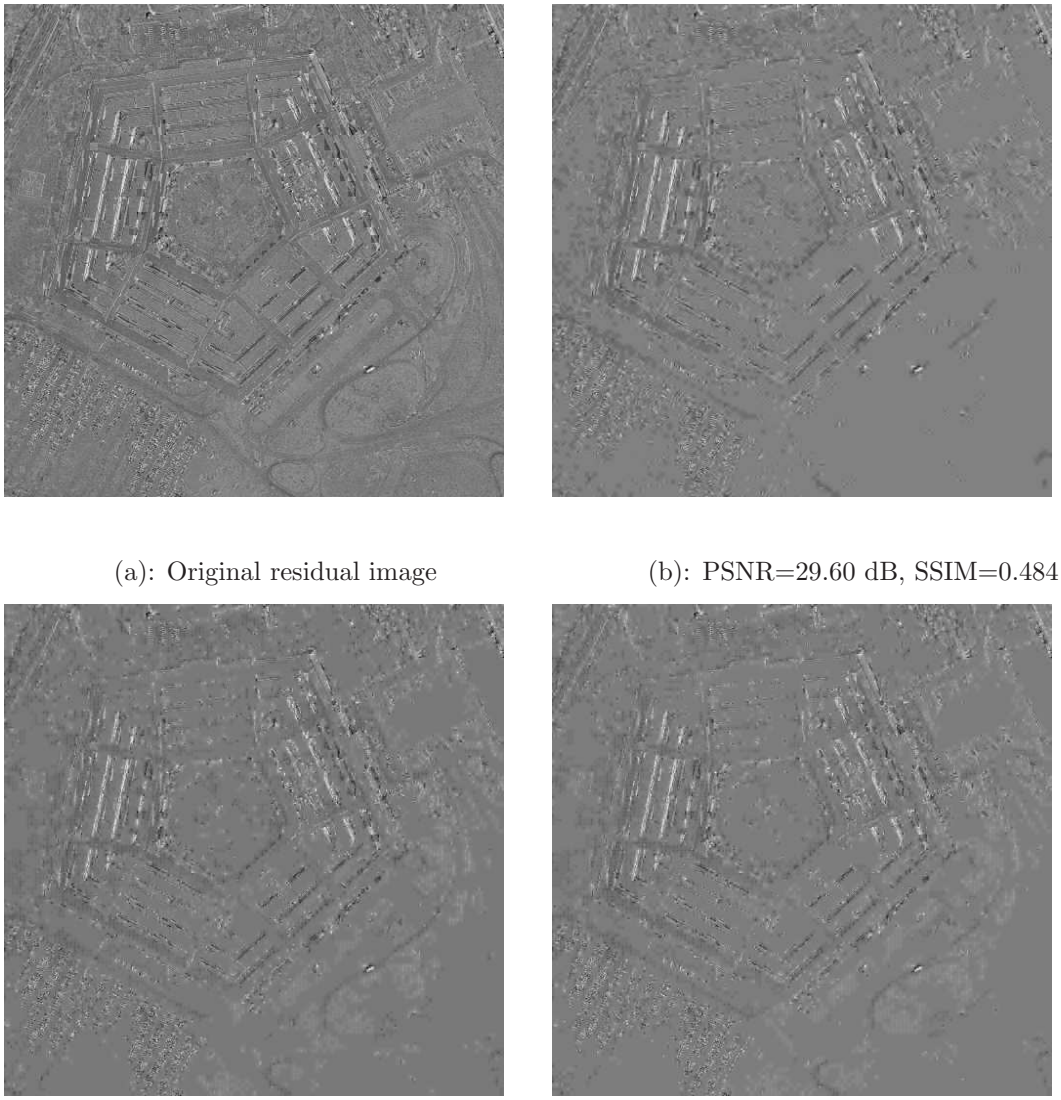



Figure 6.11: PSNR (in dB) versus the bitrate (bpp) after JPEG2000 progressive encoding for the "spot5-3" residual image.



(a): Original residual image (b): PSNR=29.60 dB, SSIM=0.484

(c): PSNR=29.65 dB, SSIM=0.515 (d): PSNR=29.87 dB, SSIM=0.541

Figure 6.12: The reconstructed "pentagon" residual image at 0.31 bpp using: (b) NSLS(2,2) (c) NSLS(2,2)-OPT1 (d) NSLS(2,2)-OPT2-PM1.

## 6.7 Conclusion

In this chapter, we have exploited the flexibility offered by non separable lifting schemes in the optimization of their operators. A new criterion has been presented for the optimization of the update filter in this context. The proposed method adapts the filters to the contents of the input image while ensuring perfect reconstruction. A theoretical analysis in terms of the chosen adaptation criterion was conducted in order to show the benefits that

can be drawn from this optimization method. Furthermore, this study provides closed form expressions of the optimal filter coefficients which, due to their simplicity, can be exploited in the implementation process. Experimental results, carried out on still images and residual images of stereo pairs have illustrated the good performance in terms of bitrate and quality of reconstruction when optimizing *both* the prediction and the update filters.

# Chapter 7

# Sparse optimization criteria for still and stereo image coding

## 7.1 Introduction

In this chapter, we investigate techniques for optimizing a sparsity promoting criterion, which can be used for the design of all the operators involved in a two-dimensional non separable lifting scheme. Unlike the previous work presented in chapter 6 [Kaaniche *et al.*, 2011] where the prediction filters have been *separately* optimized by minimizing an $\ell_2$ criterion, we propose here to perform the optimization process by minimizing an $\ell_1$ criterion. Furthermore, the main contribution of this work is to *jointly* optimize the prediction filters by minimizing a weighted $\ell_1$ criterion.

The remainder of this chapter is organized as follows. In Section 7.2, the motivation for using an $\ell_1$ criterion in lifting optimal design problems is firstly discussed. Then, the iterative algorithm for minimizing such a criterion is described in Section 7.3. In Section 7.4, we present a weighted $\ell_1$ criterion which aims at minimizing the global prediction error. In Section 7.5, we propose to *jointly* optimize the prediction filters by using an algorithm that alternates between optimizing all the filters and redefining the weights. Finally, in Section 7.6, experimental results are given and some conclusions are drawn in Section 7.7.

## 7.2 From $\ell_2$ minimization to $\ell_1$ minimization

Generally, wavelet coefficient statistics are exploited in order to increase the compression efficiency of images [LoPresto *et al.*, 1997]. More precisely, detail wavelet coefficients are often viewed as realizations of a continuous zero-mean random variable $X$ whose probability density function $f$ is given by a Generalized Gaussian Distribution (GGD) [Mallat, 1989; Payan, Antonini, 2005]:

$$\forall x \in \mathbb{R}, \quad f(x;\alpha,\beta) = \frac{\beta}{2\alpha\Gamma(\frac{1}{\beta})} e^{-\left(\frac{|x|}{\alpha}\right)^{\beta}} \tag{7.1}$$

where $\Gamma(z) = \int_0^{+\infty} t^{z-1}e^{-t}dt$ is the Gamma function, $\alpha > 0$ is the scale parameter, and $\beta > 0$ is the shape parameter. We should note that in the particular case when $\beta = 2$ (resp. $\beta = 1$), the GGD corresponds to the Gaussian distribution (resp. the Laplace one). The parameters $\alpha$ and $\beta$ can be easily estimated from the empirical moments $\chi = \mathsf{E}[|X|]$ and $\psi = \mathsf{E}[|X|^2]$. More precisely, by using the following relationship

$$\frac{\psi}{\chi^2} = \frac{\Gamma(\frac{1}{\beta})\Gamma(\frac{3}{\beta})}{\Gamma^2(\frac{2}{\beta})} \tag{7.2}$$

we can use a look-up table for different values of $\beta$ and determine the optimal value from the pair $(\chi,\psi)$ [Sharifi, Leron-Garcia, 1995]. Once we have obtained $\beta$, $\alpha$ can be deduced by using Eq. (7.2). Alternatively, maximum likelihood estimates, based on the Newton-Raphson iterative algorithm [Kay, 1993], can be used [Do, Vetterli, 2002].

Let us now adopt this GGD model for the probability of the detail coefficients generated by a lifting structure. More precisely, at each resolution level $j$ and orientation $o$ ($o \in \{HL, LH, HH\}$), the wavelet coefficients $x_{j+1}^{(o)}(m,n)$ are viewed as a realization of random variable $X_{j+1}^{(o)}$ with probability distribution given by a GGD function of parameters $\alpha_{j+1}^{(o)}$ and $\beta_{j+1}^{(o)}$. Thus, this class of distributions allows us to simply express the differential entropy $h$ of the variable $X_{j+1}^{(o)}$ as follows [Gish, Pierce, 1969; Benazza-Benyahia *et al.*, 2007]:

$$h(X_{j+1}^{(o)}) = \left(\frac{1}{M_{j+1}N_{j+1}(\alpha_{j+1}^{(o)})^{\beta_{j+1}^{(o)}}\ln(2)}\right) \sum_{m=1}^{M_{j+1}}\sum_{n=1}^{N_{j+1}} \left|x_{j+1}^{(o)}(m,n)\right|^{\beta_{j+1}^{(o)}} - \log_2\left(\frac{\beta_{j+1}^{(o)}}{2\alpha_{j+1}^{(o)}\Gamma(\frac{1}{\beta_{j+1}^{(o)}})}\right) \tag{7.3}$$

where $(M_{j+1}, N_{j+1})$ corresponds to the dimensions of $x_{j+1}^{(o)}$.

Let $\left(\bar{x}_{j+1}^{(o)}(m,n)\right)_{\substack{1\le m\le M_{j+1}\\ 1\le n\le N_{j+1}}}$ be the outputs of a uniform quantizer with quantization step $q$

driven with the real-valued coefficients $\left(x_{j+1}^{(o)}(m,n)\right)_{\substack{1\leq m\leq M_{j+1}\\1\leq n\leq N_{j+1}}}$. The coefficients $\bar{x}_{j+1}^{(o)}(m,n)$ can be viewed as a realization of a random variable $\overline{X}_{j+1}^{(o)}$ taking its values in $\{\ldots,-2q,-q,0,q,2q,\ldots\}$. At high resolution, it was proved in [Gish, Pierce, 1969] that the following relation holds between the discrete entropy $H$ of $\overline{X}_{j+1}^{(o)}$ and the differential entropy $h$ of $X_{j+1}^{(o)}$:

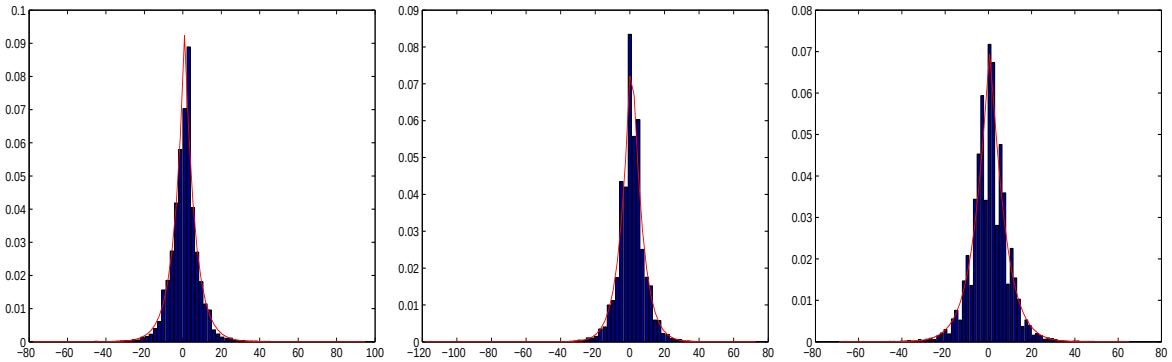
$$H(\overline{X}_{j+1}^{(o)}) \approx h(X_{j+1}^{(o)}) - \log_2(q). \tag{7.4}$$

Thus, from Eq. (7.3), we can see [Petrisor *et al.*, 2007] that the entropy $H$ of $\overline{X}_{j+1}^{(o)}$ is (up to a ln(2) dividing factor and an additive constant) approximatively equal to:

$$\left(\frac{1}{\alpha_{j+1}^{(o)}}\right)^{\beta_{j+1}^{(o)}} \sum_{m=1}^{M_{j+1}} \sum_{n=1}^{N_{j+1}} \left|x_{j+1}^{(o)}(m,n)\right|^{\beta_{j+1}^{(o)}}. \tag{7.5}$$

Therefore, we conclude that the minimization of the entropy of the detail wavelet coefficients is equivalent to the minimization of their $\ell_{\beta_{j+1}^{(o)}}$-norm.

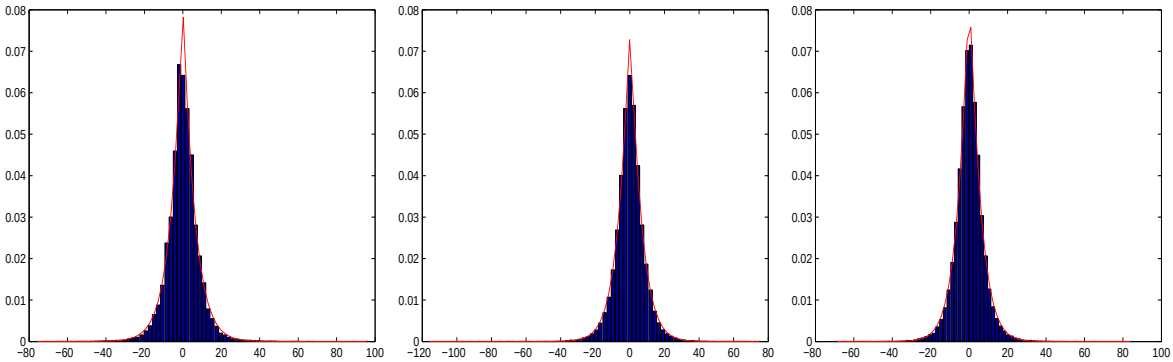
Based on these results, we have analyzed the wavelet coefficients generated from the previous NSLS(2,2) decomposition described in Section 6.6. Figure 7.1 (resp. 7.2) shows the distribution of each detail subband for the "einst" image when the prediction filters are not optimized (resp. they are optimized by minimizing the $\ell_2$-norm of the detail coefficients). The maximum likelihood technique is used to estimate the $\beta$ parameter.



(a) (b) (c)

Figure 7.1: The generalized Gaussian distribution of the: (b) horizontal detail subband $x_1^{(HL)}$ ($\beta_1^{(HL)} = 1.01$), (c): vertical detail subband $x_1^{(LH)}$ ($\beta_1^{(LH)} = 1.12$), (d): diagonal detail subband $x_1^{(HH)}$ ($\beta_1^{(HH)} = 1.16$). The detail coefficients are generated from the non-optimized NSLS(2,2) decomposition when applied to the "einst" image.

It is important to note that the shape parameters of the resulting detail subbands are closer



(a) (b) (c)

Figure 7.2: The generalized Gaussian distribution of the: (b) horizontal detail subband $x_1^{(HL)}$ ($\beta_1^{(HL)} = 1.07$), (c): vertical detail subband $x_1^{(LH)}$ ($\beta_1^{(LH)} = 1.14$), (d): diagonal detail subband $x_1^{(HH)}$ ($\beta_1^{(HH)} = 1.15$). The detail coefficients of the "einst" image are optimized by minimizing their $\ell_2$-norm.

to $\beta = 1$ than to $\beta = 2$. Therefore, this observation suggests that minimizing the $\ell_1$-norm may be more efficient than the $\ell_2$ minimization technique.

An additional advantage of $\ell_1$ minimization techniques is that they yield a sparse representation of the signal [Cai *et al.*, 2009]. Sparse representation approaches have been attracted a considerable attention over the last few years in the signal processing literature such as in the Compressed Sensing field [Donoho, 2006; Mun, Fowler, 2009], image deblurring [Yang *et al.*, 2009] and image compression [Fowler *et al.*, 2007; Masmoudi *et al.*, 2010]. For instance, the common wisdom in data compression community holds that the rate-distortion performance of coding is directly tied to the energy of the signal being compressed. On the other hand, in practical implementations of compression systems, the *sparsity* of a signal, where a portion of the signal samples are zero, has a great impact on the ultimate rate-distortion performance. For example, embedded wavelet-based image coders can spend a majority of their bit budget coding the significance map needed to locate non-zero coefficients within the wavelet domain. Here, we should note that there are relatively few works that have focused on the design of sparse decompositions for coding purposes. Indeed, geometric wavelet transforms such as curvelets [Candes, Donoho, 2004] and contourlets [Do, Vetterli, 2005] have been proposed to provide sparse representations of the images. One difficulty of such transforms is their redundancy: they produce a number of coefficients that is larger than the number of pixels in the original image. This

can be a main obstacle for achieving efficient coding schemes. To control this redundancy, a mixed contourlet and wavelet transform is proposed in [Chappelier *et al.*, 2004] where contourlet is used at fine scales and wavelet transform is employed at coarse scales. Later, bandelet transforms that aim at developing sparse geometric representation of the images have been introduced and studied in the context of image coding and image denoising [Pennec, Mallat, 2005]. Unlike contourlets and curvelets which are fixed transforms, bandelet transforms require an edge detection stage followed by an adaptive decomposition. Furthermore, the directional selectivity of 2D complex dual-tree discrete wavelet transforms [Kingsbury, 2001] has been exploited in the context of image [Fowler *et al.*, 2007] and video coding [Boettcher, Fowler, 2007]. Since such a transform is redundant, Fowler *et al.* apply a noise-shaping process [Reeves, Kingsbury, 2002] to increase the sparsity of the wavelet coefficients. Recently, a sparse block-based transform that exploits regularity along directional image singularities was proposed in [Sezer *et al.*, 2008]. This work classifies image blocks and optimizes transforms for each class by using a set of training images. In [Drémeau *et al.*, 2010b; Drémeau *et al.*, 2010a], the authors build a library of directional DCT bases using a bintree block segmentation. Then, the selection of the optimal basis from this set is made according to a rate-distortion criterion by employing dynamic programming.

However, while the minimization of an $\ell_1$ criterion is often considered in the aforementioned domains, it is worth pointing out that the use of such a criterion for lifting operator design has not been previously investigated. Note that the optimization of the prediction operator in a lifting scheme based on an entropy criterion has already been investigated in [Benazza-Benyahia *et al.*, 2007; Solé, Salembier, 2007]. In [Benazza-Benyahia *et al.*, 2007], the approach is limited to a quincunx structure and the optimization is performed in an empirical manner. In [Solé, Salembier, 2007], a generalized lifting scheme is considered. The prediction step, viewed as a mapping function, is optimized by minimizing the detail signal energy. The authors show that the resulting mapping function also minimizes the output entropy.

## 7.3 $\ell_1$ minimization method

Due to the aforementioned advantages of $\ell_1$ minimization techniques, it appears interesting to study the benefits of using such techniques in lifting filters design. For this purpose,

instead of minimizing the $\ell_2$-norm of the detail coefficients $x_{j+1}^{(o)}$ [Kaaniche *et al.*, 2011], we propose in this section to optimize the prediction filters by minimizing the following $\ell_1$ criterion:

$$\forall\, o \in \{HL, LH, HH\} \quad \text{and} \quad i \in \{1,2,3\},$$
$$\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(o)}) = \sum_{m=1}^{M_j}\sum_{n=1}^{N_j}\left|x_{i,j}(m,n) - (\mathbf{p}_j^{(o)})^\top \tilde{\mathbf{x}}_j^{(o)}(m,n)\right| \tag{7.6}$$

where $x_{i,j}(m,n)$ is the $(i+1)^{th}$ polyphase component to be predicted, $\tilde{\mathbf{x}}_j^{(o)}(m,n)$ is the reference vector containing the samples used in the prediction step, $\mathbf{p}_j^{(o)}$ is the prediction operator to be optimized ($L$ will subsequently designate its length), and $[M_j, N_j]$ corresponds to the dimensions of the image $x_j$ divided by 2.

Although the criterion in (7.6) is convex, a major difficulty that arises in solving this problem stems from the fact that the function to be minimized is not differentiable. For this purpose, a class of efficient proximal optimization algorithms has been proposed to solve (7.6) [Combettes, Pesquet, 2010]. In our context, we have employed the Douglas-Rachford algorithm which is known as one of the simplest and most efficient optimization algorithms [Eckstein, Bertsekas, 1992].

### Douglas-Rachford algorithm

For minimizing the $\ell_1$ criterion, we will resort to the concept of proximity operators [Moreau, 1965], which has been recognized as a promising tool in the recent convex optimization literature [Chaux *et al.*, 2007; Combettes, Wajs, 2005]. For this reason, before describing the algorithm, we need to address the necessary background on convex analysis and proximity operators [Hiriart-Urruty, Lemaréchal, 1993; Rockafellar, 1970]. The main definitions which will be useful to understand the remaining of this chapter are summarized briefly below:

- $\mathbb{R}^K$ is the usual $K$-dimensional Euclidean space with norm $\|.\|$.
- The distance function to a nonempty set $C \subset \mathbb{R}^K$ is defined by

$$\forall \mathbf{x} \in \mathbb{R}^K, \quad d_C(\mathbf{x}) = \inf_{\mathbf{y}\in C} \|\mathbf{x}-\mathbf{y}\|.$$

- The projection of $\mathbf{x} \in \mathbb{R}^K$ onto a nonempty closed convex set $C \subset \mathbb{R}^K$ is the unique point $P_C(\mathbf{x}) \in C$ such that $d_C(\mathbf{x}) = \|\mathbf{x} - P_C(\mathbf{x})\|$.

- The indicator function of $C$ is given by

$$\forall \mathbf{x} \in \mathbb{R}^K, \quad \imath_C(\mathbf{x}) = \begin{cases} 0 & \text{if} \quad \mathbf{x} \in C, \\ +\infty & \text{otherwise.} \end{cases} \tag{7.7}$$

- $\Gamma_0(\mathbb{R}^K)$ is the class of functions from $\mathbb{R}^K$ to $]-\infty, +\infty]$ which are lower semi-continuous, convex, and not identically equal to $+\infty$.

- The proximity operator of $f \in \Gamma_0(\mathbb{R}^K)$ is $\text{prox}_f : \mathbb{R}^K \to \mathbb{R}^K : \mathbf{x} \mapsto \arg\min_{\mathbf{y}\in\mathbb{R}^K} f(\mathbf{y}) + \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2$. It is important to note that the proximity operator generalizes the notion of a projection operator into a closed convex set $C$ in the sense that $\text{prox}_{\imath_C} = P_C$, and it moreover possesses most of its attractive properties [Moreau, 1965] that make it particularly well-suited for designing iterative minimization algorithms.

Now, we will describe the Douglas-Rachford algorithm for solving our minimization problem (7.6). We recall here that we are interested in optimizing the prediction filters by minimizing the $\ell_1$-norm of the difference between the current pixel $x_{i,j}$ and its predicted value. In this context, $\mathbf{x}_{i,j} = \Big(x_{i,j}(m,n)\Big)_{\substack{1\le m\le M_j \\ 1\le n\le N_j}}$ can be viewed as an element of the Euclidean space $\mathbb{R}^{K_j}$ where $K_j = M_j \times N_j$.
The minimization problem (7.6) can be rewritten as:

$$\begin{aligned} &\forall\ o \in \{HL, LH, HH\} \quad \text{and} \quad i \in \{1,2,3\}, \\ &\min_{\mathbf{z}_j^{(o)} \in V} \sum_{m=1}^{M_j} \sum_{n=1}^{N_j} \Big| x_{i,j}(m,n) - z_j^{(o)}(m,n) \Big| \end{aligned} \tag{7.8}$$

where $V$ is the vector space defined as

$$\begin{aligned} V = \{\mathbf{z}_j^{(o)} = \Big(z_j^{(o)}(m,n)\Big)_{\substack{1\le m\le M_j \\ 1\le n\le N_j}} \mid \exists\ \mathbf{p}_j^{(o)} \in \mathbb{R}^L, \forall\ (m,n) \in \{1,...,M_j\} \times \{1,...,N_j\}, \\ z_j^{(o)}(m,n) = (\mathbf{p}_j^{(o)})^\top \tilde{\mathbf{x}}_j^{(o)}(m,n)\}. \end{aligned}$$

Based on the definition of the indicator function, Problem (7.8) is equivalent to the following minimization problem:

$$\begin{aligned} &\forall\ o \in \{HL, LH, HH\} \quad \text{and} \quad i \in \{1,2,3\}, \\ &\min_{\mathbf{z}_j^{(o)} \in \mathbb{R}^{K_j}} \sum_{m=1}^{M_j} \sum_{n=1}^{N_j} \Big| x_{i,j}(m,n) - z_j^{(o)}(m,n) \Big| + \imath_V(\mathbf{z}_j^{(o)}). \end{aligned} \tag{7.9}$$

Therefore, Problem (7.9) can be viewed as a minimization of a sum of two functions $f_1$ and $f_2$ defined by:

$$f_1(\mathbf{z}_j^{(o)}) = \|\mathbf{x}_{i,j} - \mathbf{z}_j^{(o)}\|_{\ell_1} = \sum_{m=1}^{M_j}\sum_{n=1}^{N_j}\left|x_{i,j}(m,n) - z_j^{(o)}(m,n)\right| \tag{7.10}$$

$$f_2(\mathbf{z}_j^{(o)}) = \imath_V(\mathbf{z}_j^{(o)}). \tag{7.11}$$

In this case, the Douglas-Rachford algorithm can be applied to provide an appealing numerical solution to Problem (7.9). The solution is obtained by the following iterative algorithm:

$$\begin{aligned}
&\text{Set } \mathbf{t}_{j,0}^{(o)} \in \mathbb{R}^{K_j}, \gamma > 0, \lambda \in ]0,2[, \text{and,}\\
&\quad \text{for } k = 0,1,2,...\\
&\qquad \mathbf{z}_{j,k}^{(o)} = \text{prox}_{\gamma f_2}\mathbf{t}_{j,k}^{(o)}\\
&\qquad \mathbf{t}_{j,k+1}^{(o)} = \mathbf{t}_{j,k}^{(o)} + \lambda\Big(\text{prox}_{\gamma f_1}(2\mathbf{z}_{j,k}^{(o)} - \mathbf{t}_{j,k}^{(o)}) - \mathbf{z}_{j,k}^{(o)}\Big).
\end{aligned} \tag{7.12}$$

An important feature of this algorithm is that it proceeds by splitting, in the sense that the functions $f_1$ and $f_2$ are dealt with in separate steps: in the first step, only the function $f_2$ is required to obtain $\mathbf{z}_{j,k}^{(o)}$ and, in the second step, only the function $f_1$ is required to obtain $\mathbf{t}_{j,k+1}^{(o)}$. Furthermore, it can be seen that the algorithm requires to compute two proximity operators $\text{prox}_{\gamma f_1}$ and $\text{prox}_{\gamma f_2}$ at each iteration. One can find in [Combettes, Pesquet, 2010] closed-form expression of the proximity operator of various functions in $\Gamma_0(\mathbb{R})$. In our case, the proximity operator of $\gamma f_1$ is given by:

$$\forall\ \mathbf{t}_{j,k}^{(o)} \in \mathbb{R}^{K_j}, \quad \text{prox}_{\gamma f_1}(\mathbf{t}_{j,k}^{(o)}) = \Big(\pi_{j,k}^{(o)}(m,n)\Big)_{\substack{1\le m\le M_j\\ 1\le n\le N_j}} \tag{7.13}$$

where

$$\pi_{j,k}^{(o)}(m,n) = \text{soft}_{[-\gamma,\gamma]}\left(t_{j,k}^{(o)}(m,n) - x_{i,j}(m,n)\right) + x_{i,j}(m,n) \tag{7.14}$$

where

$$\forall\ \alpha \in \mathbb{R}, \quad \text{soft}_{[-\gamma,\gamma]}(\alpha) = \begin{cases} \text{sign}(\alpha)(|\alpha| - \gamma) & \text{if} \quad |\alpha| > \gamma \\ 0 & \text{otherwise.} \end{cases} \tag{7.15}$$

Concerning $\gamma f_2$, it is easy to check that its proximity operator is expressed as:

$$\begin{aligned}
\forall\ \mathbf{t}_{j,k}^{(o)} \in \mathbb{R}^{K_j}, \quad \text{prox}_{\gamma f_2}(\mathbf{t}_{j,k}^{(o)}) &= P_V(\mathbf{t}_{j,k}^{(o)})\\
&= \Big(\hat{z}_{j,k}^{(o)}(m,n)\Big)_{\substack{1\le m\le M_j\\ 1\le n\le N_j}} = \Big((\mathbf{p}_{j,k}^{(o)})^\top \tilde{\mathbf{x}}_j^{(o)}(m,n)\Big)_{\substack{1\le m\le M_j\\ 1\le n\le N_j}}
\end{aligned} \tag{7.16}$$

where $\mathbf{p}_{j,k}^{(o)} = \left( \sum_{m,n} \tilde{\mathbf{x}}_j^{(o)}(m,n)(\tilde{\mathbf{x}}_j^{(o)}(m,n))^\top \right)^{-1} \sum_{m,n} \tilde{\mathbf{x}}_j^{(o)}(m,n) t_{j,k}^{(o)}(m,n)$.

Finally, it is important to note that it has been shown (see [Combettes, Pesquet, 2007] and references therein) that every sequence $\mathbf{z}_{j,k}^{(o)}$ generated by the Douglas-Rachford algorithm (7.12) converges to a solution to problem (7.9) provided that the parameters $\gamma$ and $\lambda$ are fixed as indicated.

Once the different terms involved in the iterative algorithm (7.12) are defined, this one can be applied to optimize all the prediction filters.

## 7.4 Global prediction error minimization technique

### 7.4.1 Motivation

Up to now, each prediction filter $\mathbf{p}_j^{(o)}$ ($o \in \{HL, LH, HH\}$) has been separately optimized by minimizing the $\ell_1$-norm of the corresponding detail signal $x_{j+1}^{(o)}$ which seems appropriate to determine $\mathbf{p}_j^{(LH)}$ and $\mathbf{p}_j^{(HL)}$. However, it can be noticed from Fig. 7.3 that the diagonal detail signal $x_{j+1}^{(HH)}$ is also used through the second and the third prediction steps to compute the vertical and the horizontal detail signals respectively. Therefore, the solution $\mathbf{p}_j^{(HH)}$ resulting from the previous optimization method may be suboptimal.

As a result, we propose to optimize the prediction filter $\mathbf{p}_j^{(HH)}$ by minimizing a global prediction error as we describe in detail in the next section.

### 7.4.2 Optimization of the prediction filter $\mathbf{p}_j^{(HH)}$

More precisely, instead of minimizing the $\ell_1$-norm of $x_{j+1}^{(HH)}$, the filter $\mathbf{p}_j^{(HH)}$ will be optimized by minimizing the sum of the $\ell_1$-norm of the three detail subbands $x_{j+1}^{(o)}$. To this respect, we will consider the minimization of the following weighted $\ell_1$ criterion:

$$\mathcal{J}_{w\ell_1}(\mathbf{p}_j^{(HH)}) = \sum_{o \in \{HL,LH,HH\}} \sum_{m,n} \sqrt{w_j^{(o)}} \left| x_{j+1}^{(o)}(m,n) \right| \tag{7.17}$$

where $w_j^{(o)}$ is a strictly positive weighting term.

Before focusing on the method used to minimize the proposed criterion, we should first express $\mathcal{J}_{w\ell_1}$ as a function of the filter $\mathbf{p}_j^{(HH)}$ to be optimized.

Let $\left( x_{i,j}^{(1)}(m,n) \right)_{i \in \{0,1,2,3\}}$ be the four outputs obtained from $\left( x_{i,j}(m,n) \right)_{i \in \{0,1,2,3\}}$ following the first prediction step (see Fig. 7.3). Although $x_{i,j}^{(1)}(m,n) = x_{i,j}(m,n)$ for all $i \in \{0,1,2\}$,

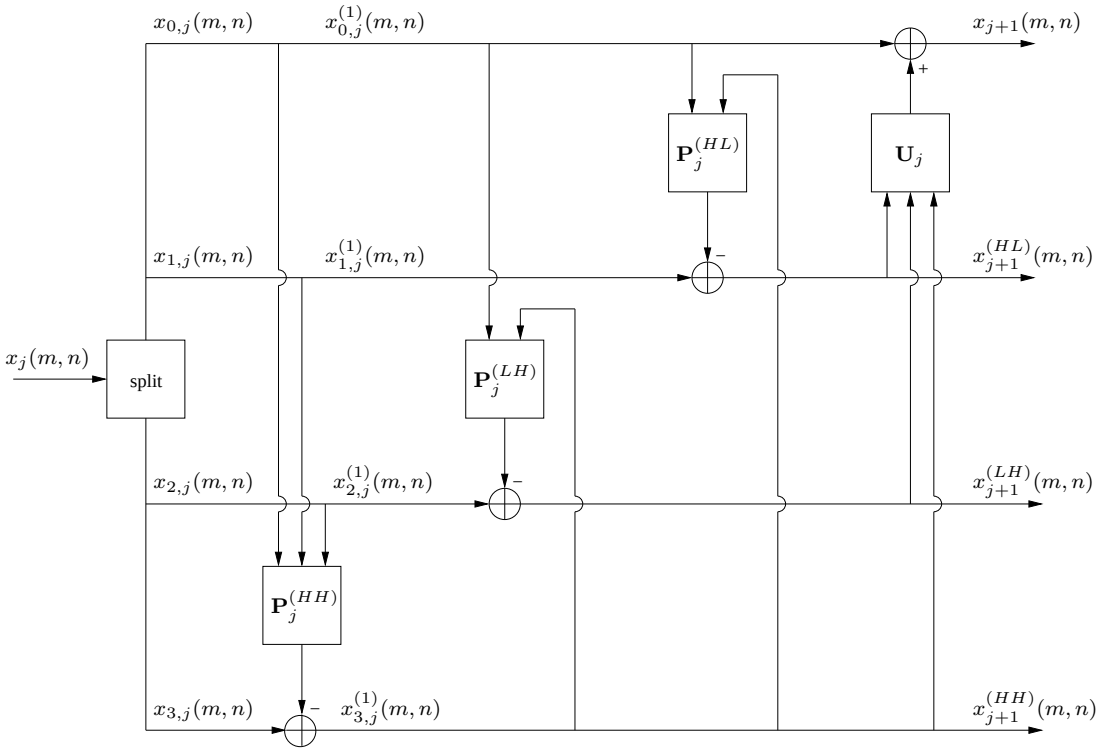



Figure 7.3: NSLS decomposition structure.

the use of the superscript will make the presentation below easier. Thus, $x^{(o)}_{j+1}$ can be expressed as:

$$\begin{aligned} x^{(o)}_{j+1}(m,n) &= \sum_{i\in\{0,1,2,3\}} \sum_{k,l} h^{(o,1)}_{i,j}(k,l) x^{(1)}_{i,j}(m-k,n-l) \\ &= \sum_{i\in\{0,1,2\}} \sum_{k,l} h^{(o,1)}_{i,j}(k,l) x^{(1)}_{i,j}(m-k,n-l) + \sum_{k,l} h^{(o,1)}_{3,j}(k,l) x^{(1)}_{3,j}(m-k,n-l) \end{aligned} \tag{7.18}$$

where $h^{(o,1)}_{i,j}$ is a filter which depends on the prediction coefficients of $\mathbf{p}^{(LH)}_j$ and $\mathbf{p}^{(HL)}_j$. Knowing

$$x^{(1)}_{3,j}(m,n) = x_{3,j}(m,n) - (\mathbf{p}^{(HH)}_j)^\top \tilde{\mathbf{x}}^{(HH)}_j(m,n) \tag{7.19}$$

where $\tilde{\mathbf{x}}^{(HH)}_j(m,n) = \Big(x_{i,j}(m-r,n-s)\Big)_{\substack{(r,s)\in\mathcal{P}^{(HH)}_j \\ i\in\{0,1,2\}}}$ ($\mathcal{P}^{(HH)}_j$ is the support of the predictor $\mathbf{p}^{(HH)}_j$), we thus obtain, after some simple calculations,

$$\forall\ o \in \{HH, LH, HL\}, \quad x^{(o)}_{j+1}(m,n) = y^{(o,1)}_j(m,n) - (\mathbf{p}^{(HH)}_j)^\top \mathbf{x}^{(o,1)}_j(m,n) \tag{7.20}$$

where

$$y_j^{(o,1)}(m,n) = \sum_{i\in\{0,1,2\}} \sum_{k,l} h_{i,j}^{(o,1)}(k,l) x_{i,j}^{(1)}(m-k,n-l) + \sum_{k,l} h_{3,j}^{(o,1)}(k,l) x_{3,j}(m-k,n-l), \tag{7.21}$$

$$\mathbf{x}_j^{(o,1)}(m,n) = \Big( \sum_{k,l} h_{3,j}^{(o,1)}(k,l) x_{i,j}(m-k-r,n-l-s) \Big)_{\substack{(r,s)\in\mathcal{P}_j^{(HH)} \\ i\in\{0,1,2\}}}. \tag{7.22}$$

Consequently, the proposed weighted $\ell_1$ criterion (Eq. 7.17) can be expressed as:

$$\mathcal{J}_{w\ell_1}(\mathbf{p}_j^{(HH)}) = \sum_{o\in\{HL,LH,HH\}} \sum_{m,n} \sqrt{w_j^{(o)}} \Big| y_j^{(o,1)}(m,n) - (\mathbf{p}_j^{(HH)})^\top \mathbf{x}_j^{(o,1)}(m,n) \Big|. \tag{7.23}$$

It is worth noting that in practice, the determination of $y_j^{(o,1)}(m,n)$ and $\mathbf{x}_j^{(o,1)}(m,n)$ does not require to find the explicit expressions of $h_{i,j}^{(o,1)}$ and these signals can be determined numerically as follows:

- The first term (resp. the second one) in the expression of $y_j^{(o,1)}(m,n)$ (Eq. (7.21)) can be found by computing $x_{j+1}^{(o)}(m,n)$ from the components $\Big(x_{i,j}^{(1)}(m,n)\Big)_{i\in\{0,1,2,3\}}$ while setting $x_{3,j}^{(1)}(m,n) = 0$ (resp. while setting $x_{i,j}^{(1)}(m,n) = 0$ for $i \in \{0,1,2\}$ and $x_{3,j}^{(1)}(m,n) = x_{3,j}(m,n)$).
- The vector $\mathbf{x}_j^{(o,1)}(m,n)$ (Eq. (7.22)) can be found as follows. For each $i \in \{0,1,2\}$, the computation of its component $\sum_{k,l} h_{3,j}^{(o,1)}(k,l) x_{i,j}(m-k,n-l)$ requires to compute $x_{j+1}^{(o)}(m,n)$ by setting $x_{3,j}^{(1)}(m,n) = x_{i,j}(m,n)$ and $x_{i',j}^{(1)}(m,n) = 0$ for $i' \in \{0,1,2\}$. We perform this operation for the different shift values $(r,s)$ (as it can be seen in Eq. 7.22).

Once the different terms involved in the proposed weighted criterion (Eq. (7.23)) are defined (the weighting values $w_j^{(o)}$ are supposed to be known), we will focus now on its minimization. Indeed, unlike the previous criterion (Eq. (7.6)) which consists only of an $\ell_1$ term, the proposed criterion is a sum of three $\ell_1$ terms. To minimize such a criterion (7.23), one can still use the Douglas-Rachford algorithm through a formulation in a product space [Combettes, Pesquet, 2010].

### Douglas-Rachford algorithm in a product space

Consider the $\ell_1$ minimization problem:

$$\min_{\mathbf{p}_j^{(HH)}} \sum_{o\in\{HL,LH,HH\}} \sum_{m,n} \sqrt{w_j^{(o)}} \Big| y_j^{(o,1)}(m,n) - (\mathbf{p}_j^{(HH)})^\top \mathbf{x}_j^{(o,1)}(m,n) \Big| \tag{7.24}$$

where $w_j^{(o)}$ are positive weights.

Since the Douglas-Rachford algorithm described hereabove is designed for the sum of two functions, we can attempt to reformulate (7.24) under this form in the 3-fold product space $\mathbb{H}_j$

$$\mathbb{H}_j = \mathbb{R}^{K_j} \times \mathbb{R}^{K_j} \times \mathbb{R}^{K_j} \tag{7.25}$$

If we define the vector subspace $U$ as

$$U = \Big\{ \mathbf{Z}_j = \begin{pmatrix} \mathbf{z}_j^{(HH,1)} \\ \mathbf{z}_j^{(LH,1)} \\ \mathbf{z}_j^{(HL,1)} \end{pmatrix} \in \mathbb{H}_j \mid \exists \ \mathbf{p}_j^{(HH)} \in \mathbb{R}^L, \ \forall\, o \in \{HH, LH, HL\},$$
$$\forall\ (m,n) \in \{1,2,...,M_j\} \times \{1,2,...,N_j\},\ z_j^{(o,1)}(m,n) = (\mathbf{p}_j^{(HH)})^\top \mathbf{x}_j^{(o,1)}(m,n) \Big\}$$

$$= \Big\{ \mathbf{Z}_j = \begin{pmatrix} \mathbf{z}_j^{(HH,1)} \\ \mathbf{z}_j^{(LH,1)} \\ \mathbf{z}_j^{(HL,1)} \end{pmatrix} \in \mathbb{H}_j \mid \exists \ \mathbf{p}_j^{(HH)} \in \mathbb{R}^L, \ \forall\ (m,n) \in \{1,2,...,M_j\} \times \{1,2,...,N_j\},$$
$$\mathbf{Z}_j(m,n) = \mathbf{X}_j(m,n)^\top \mathbf{p}_j^{(HH)} \text{ with } \mathbf{X}_j(m,n) = \Big(\mathbf{x}_j^{(HH,1)}(m,n), \mathbf{x}_j^{(LH,1)}(m,n), \mathbf{x}_j^{(HL,1)}(m,n)\Big) \ \Big\}, \tag{7.26}$$

the minimization problem (7.24) is equivalent to

$$\min_{\mathbf{z}_j \in \mathbb{H}_j} f_1(\mathbf{z}_j) + f_2(\mathbf{z}_j) \tag{7.27}$$

where

$$f_1(\mathbf{z}_j) = \sum_{o \in \{HL,LH,HH\}} \sum_{m,n} \sqrt{w_j^{(o)}} \Big| y_j^{(o,1)}(m,n) - z_j^{(o,1)}(m,n) \Big|$$
$$f_2(\mathbf{z}_j) = \imath_U(\mathbf{z}_j). \tag{7.28}$$

We are thus back to a problem involving two functions in a larger space $\mathbb{H}_j$. So, the Douglas-Rachford algorithm can be applied to solve our minimization problem:

$$\begin{aligned} &\text{Set } \mathbf{t}_{j,0} \in \mathbb{H}_j, \gamma > 0, \lambda \in ]0,2[, \text{and,} \\ &\quad \text{for } k = 0,1,2,... \\ &\qquad \mathbf{z}_{j,k} = \text{prox}_{\gamma f_2} \mathbf{t}_{j,k} \\ &\qquad \mathbf{t}_{j,k+1} = \mathbf{t}_{j,k} + \lambda \Big( \text{prox}_{\gamma f_1}(2\mathbf{z}_{j,k} - \mathbf{t}_{j,k}) - \mathbf{z}_{j,k} \Big). \end{aligned} \tag{7.29}$$

Note that the above algorithm requires to compute the proximity operators of 2 new functions $\gamma f_1$ and $\gamma f_2$. Concerning the proximity operator of $\gamma f_1$, we have

$$\forall\ \mathbf{t}_j = \begin{pmatrix} \mathbf{t}_j^{(HH,1)} \\ \mathbf{t}_j^{(LH,1)} \\ \mathbf{t}_j^{(HL,1)} \end{pmatrix} \in \mathbb{H}_j,\ \text{prox}_{\gamma f_1}(\mathbf{t}_{j,k}) = \begin{pmatrix} \text{soft}_{\left[-\gamma\sqrt{w_j^{(HH)}},\gamma\sqrt{w_j^{(HH)}}\right]}(\mathbf{t}_{j,k}^{(HH,1)}) \\ \text{soft}_{\left[-\gamma\sqrt{w_j^{(LH)}},\gamma\sqrt{w_j^{(LH)}}\right]}(\mathbf{t}_{j,k}^{(LH,1)}) \\ \text{soft}_{\left[-\gamma\sqrt{w_j^{(HL)}},\gamma\sqrt{w_j^{(HL)}}\right]}(\mathbf{t}_{j,k}^{(HL,1)}) \end{pmatrix} \tag{7.30}$$

where

$$\forall\ o\ \in \{HH, LH, HL\},$$
$$\text{soft}_{\left[-\gamma\sqrt{w_j^{(o)}},\gamma\sqrt{w_j^{(o)}}\right]}(\mathbf{t}_{j,k}^{(o,1)}) = \left(\text{soft}_{\left[-\gamma\sqrt{w_j^{(o)}},\gamma\sqrt{w_j^{(o)}}\right]}(t_{j,k}^{(o,1)}(m,n))\right)_{\substack{1\le m\le M_j \\ 1\le n\le N_j}}. \tag{7.31}$$

Concerning $\gamma f_2$, its proximity operator is given by:

$$\begin{aligned} \text{prox}_{\gamma f_2}(\mathbf{t}_{j,k}) &= P_U(\mathbf{t}_{j,k}) \\ &= \left(\hat{\mathbf{Z}}_{j,k}(m,n)\right)_{\substack{1\le m\le M_j \\ 1\le n\le N_j}} = \left(\mathbf{X}_j(m,n)^\top \mathbf{p}_{j,k}^{(HH)}\right)_{\substack{1\le m\le M_j \\ 1\le n\le N_j}} \end{aligned} \tag{7.32}$$

where

$$\mathbf{p}_{j,k}^{(HH)} = \left(\sum_m \sum_n \mathbf{X}_j(m,n)\mathbf{X}_j(m,n)^\top\right)^{-1} \sum_{m,n} \mathbf{X}_j(m,n)\mathbf{t}_{j,k}(m,n).$$

Finally, once the prediction filter $\mathbf{p}_j^{(HH)}$ is optimized and fixed, it can be noticed that the other prediction filters $\mathbf{p}_j^{(HL)}$ and $\mathbf{p}_j^{(LH)}$ can be separately optimized by minimizing $\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(HL)})$ and $\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(LH)})$ as explained in Section 7.3. This is justified by the fact the inputs of the filter $\mathbf{p}_j^{(HL)}$ (resp. $\mathbf{p}_j^{(LH)}$) are independent of the output of the filter $\mathbf{p}_j^{(LH)}$ (resp. $\mathbf{p}_j^{(HL)}$).

## 7.5 Joint optimization method

### 7.5.1 Motivation

From Equations (7.21) and (7.22), it can be observed that $y_j^{(o,1)}$ and $\mathbf{x}_j^{(o,1)}$, which are used to optimize $\mathbf{p}_j^{(HH)}$, depend on the coefficients of the prediction filters $\mathbf{p}_j^{(HL)}$ and $\mathbf{p}_j^{(LH)}$. On the other hand, since $\mathbf{p}_j^{(HL)}$ and $\mathbf{p}_j^{(LH)}$ use $x_{j+1}^{(HH)}$ as a reference signal in the second and the third prediction steps, their optimal values will depend on the optimal prediction filter

$\mathbf{p}_j^{(HH)}$. Thus, we conclude that the optimization of the filters $(\mathbf{p}_j^{(HL)}, \mathbf{p}_j^{(LH)})$ depends on the optimization of the filter $\mathbf{p}_j^{(HH)}$ and vice-versa.
A *joint* optimization method can therefore be proposed which iteratively optimizes the prediction filters $\mathbf{p}_j^{(HH)}$, $\mathbf{p}_j^{(HL)}$ and $\mathbf{p}_j^{(LH)}$.

### 7.5.2 Proposed algorithms

While the optimization of the prediction filters $\mathbf{p}_j^{(HL)}$ and $\mathbf{p}_j^{(LH)}$ is simple, the optimization of the prediction filter $\mathbf{p}_j^{(HH)}$ is less obvious. Indeed, if we examine the criterion $\mathcal{J}_{w\ell_1}$, the immediate question that arises is: which values of the weighting parameters will produce the sparsest decomposition?

A simple solution consists of setting all the weights $w_j^{(o)}$ to one. Then, we are considering the particular case of the unweighted $\ell_1$ criterion which simply represents the sum of the $\ell_1$-norm of the three details subbands $x_{j+1}^{(o)}$. In this case, the joint optimization problem is solved by applying the following simple iterative algorithm at each resolution level $j$.

#### First proposed algorithm

① Initialize the iteration number $it$ to 0.

- Optimize separately the three prediction filters as explained in section 7.3. The resulting filters will be denoted respectively by $\mathbf{p}_j^{(HH,it)}$, $\mathbf{p}_j^{(LH,it)}$ and $\mathbf{p}_j^{(HL,it)}$.
- Compute the resulting global unweighted prediction error (i.e. the sum of the $\ell_1$-norm of the three resulting details subbands).

② for $it = 1, 2, 3, ...$

- Set $\mathbf{p}_j^{(LH)} = \mathbf{p}_j^{(LH,it-1)}$, $\mathbf{p}_j^{(HL)} = \mathbf{p}_j^{(HL,it-1)}$, and optimize $\mathbf{P}_j^{(HH)}$ by minimizing $\mathcal{J}_{w\ell_1}(\mathbf{p}_j^{(HH)})$ (while setting $w_j^{(o)} = 1$). Let $\mathbf{p}_j^{(HH,it)}$ be the new optimal filter.
- Set $\mathbf{p}_j^{(HH)} = \mathbf{p}_j^{(HH,it)}$, and optimize $\mathbf{P}_j^{(LH)}$ by minimizing $\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(LH)})$. Let $\mathbf{p}_j^{(LH,it)}$ be the new optimal filter.
- Set $\mathbf{p}_j^{(HH)} = \mathbf{p}_j^{(HH,it)}$, and optimize $\mathbf{P}_j^{(HL)}$ by minimizing $\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(HL)})$. Let $\mathbf{p}_j^{(HL,it)}$ be the new optimal filter.

Once the prediction filters are optimized, the update filter is finally optimized by minimizing the difference between the approximation signal and the decimated version of the output

of an ideal low-pass filter (as explained in Section 6.3.2).
Table 7.1 shows the global unweighted prediction error ($w_j^{(o)} = 1$), obtained at a resolution level $j$, with respect to the iteration number of the algorithm. In this case, it can be noticed that this error decreases from an iteration $it$ to $it + 1$. However, in practice, once all the filters are optimized and the decomposition is performed, the different generated wavelet subbands are weighted before the entropy encoding as discussed at the end of Section 3.2.1. Therefore, the resulting filters are suboptimal in the sense that they do not take into account the weighting procedure. For this reason, we have noticed in our experiments that this optimization technique does not improve the coding performance.

A more judicious choice of $w_j^{(o)}$ consists of setting their values to the weights of the detail subbands $x_{j+1}^{(o)}$ as given by Eq. (3.23). As mentioned at the end of Section 3.2.1, these weights depend on the filters used for the reconstruction process [Usevitch, 1996]. Thus, the main difficulty in this case is that our criterion uses weights which also depend on the prediction filters. To solve this problem, we propose an alternative iterative algorithm that alternates between optimizing all the filters and redefining the weights. This algorithm, which is performed for each resolution level $j$, is as follows.

**Second proposed algorithm**

① Initialize the iteration number $it$ to 0.

- Optimize separately the three prediction filters as explained in section 7.3. The resulting filters will be denoted respectively by $\mathbf{p}_j^{(HH,it)}$, $\mathbf{p}_j^{(LH,it)}$ and $\mathbf{p}_j^{(HL,it)}$.
- **Optimize the update filter (as proposed in Section 6.3.2)).**
- **Compute the weights $w_j^{(o,it)}$ of each detail subband (using Eq. (3.23)) as well as the resulting global weighted prediction error (i.e. the weighted sum of the $\ell_1$-norm of the three resulting detail subbands).**

② for $it = 1, 2, 3, ...$

- Set $\mathbf{p}_j^{(LH)} = \mathbf{p}_j^{(LH,it-1)}$, $\mathbf{p}_j^{(HL)} = \mathbf{p}_j^{(HL,it-1)}$, and optimize $\mathbf{P}_j^{(HH)}$ by minimizing $\mathcal{J}_{w\ell_1}(\mathbf{p}_j^{(HH)})$. Let $\mathbf{p}_j^{(HH,it)}$ be the new optimal filter.
- Set $\mathbf{p}_j^{(HH)} = \mathbf{p}_j^{(HH,it)}$, and optimize $\mathbf{P}_j^{(LH)}$ by minimizing $\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(LH)})$. Let $\mathbf{p}_j^{(LH,it)}$ be the new optimal filter.
- Set $\mathbf{p}_j^{(HH)} = \mathbf{p}_j^{(HH,it)}$, and optimize $\mathbf{P}_j^{(HL)}$ by minimizing $\mathcal{J}_{\ell_1}(\mathbf{p}_j^{(HL)})$. Let $\mathbf{p}_j^{(HL,it)}$

be the new optimal filter.

- **Optimize the update filter (as proposed in Section 6.3.2)).**
- **Compute the new weights $w_j^{(o,it)}$ (using Eq. (3.23)) as well as the resulting global weighted error.**

③ Select the best optimal filters from the iteration giving the minimum criterion value.

It is worth pointing out that step 3 has been introduced at the end of this algorithm because the global weighted prediction error does not necessary decrease from an iteration $it$ to $it+1$. This can be explained by the fact that the weights are updated after the optimization process of all the filters at each iteration of the algorithm.

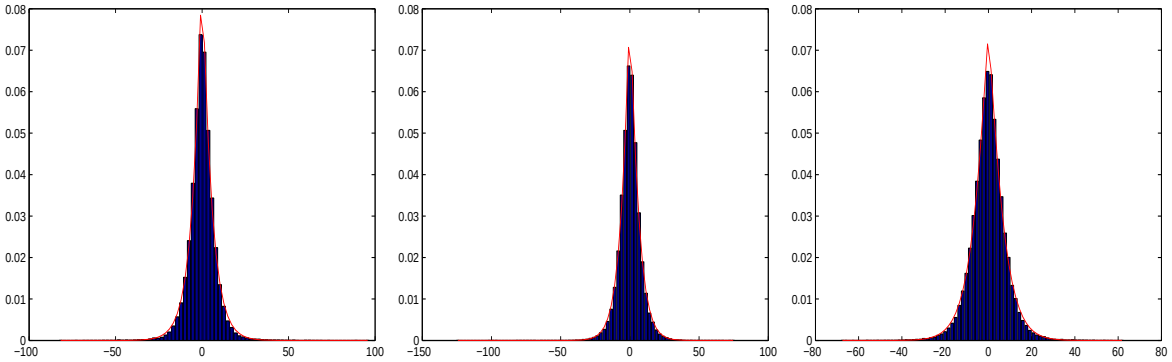
Table 7.1: Evolution of the global prediction error obtained at each resolution level $j$ for the "lena" and "einst" images.

| Image | Filter | | $it=0$ | $it=1$ | $it=2$ | $it=3$ | $it=4$ | $it=5$ | $it=6$ | $it=7$ |
|---|---|---|---|---|---|---|---|---|---|---|
| einst | Global | $j$=1 | 15.864 | 15.820 | 15.769 | 15.658 | 15.554 | **15.522** | 15.585 | 15.647 |
| | weighted | $j$=2 | 28.191 | 27.797 | **27.451** | 27.466 | 27.601 | 27.804 | 27.827 | 27.908 |
| | error | $j$=3 | 64.336 | 62.963 | **62.556** | 62.971 | 63.568 | 64.042 | 64.863 | 64.996 |
| | global | $j$=1 | 17.143 | 17.133 | 17.102 | 17.040 | 16.971 | 16.930 | 16.919 | **16.918** |
| | unweighted | $j$=2 | 21.168 | 21.017 | 20.840 | 20.747 | 20.725 | 20.723 | 20.722 | **20.722** |
| | error | $j$=3 | 26.218 | 25.924 | 25.739 | 25.639 | 25.617 | 25.615 | 25.613 | **25.611** |
| lena | Global | $j$=1 | 9.191 | 9.081 | **9.066** | 9.216 | 9.352 | 9.400 | 9.423 | 9.421 |
| | weighted | $j$=2 | 23.360 | 23.163 | **23.073** | 23.342 | 23.627 | 23.647 | 23.707 | 23.691 |
| | error | $j$=3 | 71.178 | 69.928 | **69.071** | 69.266 | 71.284 | 71.429 | 72.383 | 73.268 |
| | global | $j$=1 | 9.679 | 9.617 | 9.563 | 9.535 | 9.525 | 9.523 | 9.523 | **9.522** |
| | unweighted | $j$=2 | 16.948 | 16.811 | 16.691 | 16.627 | 16.615 | 16.613 | 16.613 | **16.612** |
| | error | $j$=3 | 27.898 | 27.645 | 27.404 | 27.308 | 27.291 | 27.281 | 27.275 | **27.272** |

Table 7.1 shows also the global weighted prediction error, obtained at a resolution level $j$, with respect to the iteration number of the algorithm. It can be noticed that the decrease of the weighted error is mainly achieved during the early iterations. As a result, we have chosen the iteration number equal to 5.

Using this optimization method to design the prediction filters of the NSLS(2,2), Figure 7.4 shows the distribution of the resulting detail subbands for the "einst" image. This figure

confirms that the shape parameters of these distributions remain closer to one than to two.



(a) (b) (c)

Figure 7.4: The generalized Gaussian distribution of the: (b) horizontal detail subband $x_1^{(HL)}$ ($\beta_1^{(HL)} = 1.02$), (c): vertical detail subband $x_1^{(LH)}$ ($\beta_1^{(LH)} = 1.12$), (d): diagonal detail subband $x_1^{(HH)}$ ($\beta_1^{(HH)} = 1.17$). The detail coefficients of the "einst" image are optimized by minimizing the weighted $\ell_1$ criterion.

## 7.6 Experimental results

Simulations were carried out on two kinds of still images originally quantized over 8 bpp which are either single views[1] or stereoscopic ones[2]. The gain related to the optimization of the NSLS operators, using different minimization criteria, was evaluated in these contexts. Since we are interested in the optimization of all the operators involved in the considered NSLS (which corresponds to the 2D scheme of any 1D P-U lifting structure), we will mainly consider the example of the 5/3 wavelet transform. In order to show the benefits of the proposed $\ell_1$ optimization criterion, we provide the results for the following decompositions carried out over three resolution levels:

• The first one is the lifting scheme corresponding to the 5/3 transform, which is known as the (2,2) integer wavelet transform [Calderbank *et al.*, 1998]. In the following, this method will be designated by NSLS(2,2).

• The second variant corresponds to our previous method presented recently in [Kaaniche *et al.*, 2011]. We recall that this method consists of optimizing the prediction filters of

[1] taken from the URL http://sipi.usc.edu/database

[2] http://vasc.ri.cmu.edu/idb/html/stereo/index.html

the NSLS by minimizing the $\ell_2$-norm of the detail coefficients. Moreover, the update filter is optimized by minimizing the difference between the approximation signal and the decimated version of the output of an ideal low-pass filter. We emphasize here that the prediction filters are *separately* optimized. This method will be denoted by NSLS(2,2)-OPT-L2.

• The third method modifies the optimization stage of the prediction filters by using the $\ell_1$-norm instead of the $\ell_2$-norm. In what follows, this method will be designated by NSLS(2,2)-OPT-L1.

• The fourth method consists of *jointly* optimizing the prediction filters by using the proposed weighted $\ell_1$ minimization technique. This optimization method will be designated by NSLS(2,2)-OPT-WL1.

Figures 7.5 and 7.6 show the scalability in quality of the reconstruction procedure by providing the variations of the PSNR versus the bitrate for the images "castle" and "einst" using JPEG2000 as codec. These plots show that NSLS(2,2)-OPT-L2 outperforms NSLS(2,2) by 0.1-0.5 dB. An improvement of 0.1-0.3 dB is obtained by using the $\ell_1$ minimization technique instead of the $\ell_2$ one. Finally, the *joint* optimization technique (NSLS(2,2)-OPT-WL1) outperforms the *separate* optimization technique (NSLS(2,2)-OPT-L1) and improves the PSNR by 0.1-0.2 dB. The gain becomes more important (up to 0.55 dB) when compared to NSLS(2,2)-OPT-L2.

Figures 7.7 and 7.8 display the reconstructed images of "lena" and "einst". In addition to PSNR and SSIM metrics, the quality of the reconstructed images are also compared in terms of VSNR (Visual Signal-to-Noise ratio) which was found to be an efficient metric for quantifying the visual fidelity of natural images [Chandler, Hemami, 2007]: it is based on physical luminances and visual angle (rather than on digital pixel values and pixel-based dimensions) to accommodate different viewing conditions. It can be observed that the weighted $\ell_1$ minimization technique significantly improves the visual quality of reconstruction. The difference in VSNR (resp. PSNR) between NSLS(2,2)-OPT-L2 and NSLS(2,2)-OPT-WL1 ranges from 0.35 dB to 0.6 dB (resp. 0.25 dB to 0.3 dB).

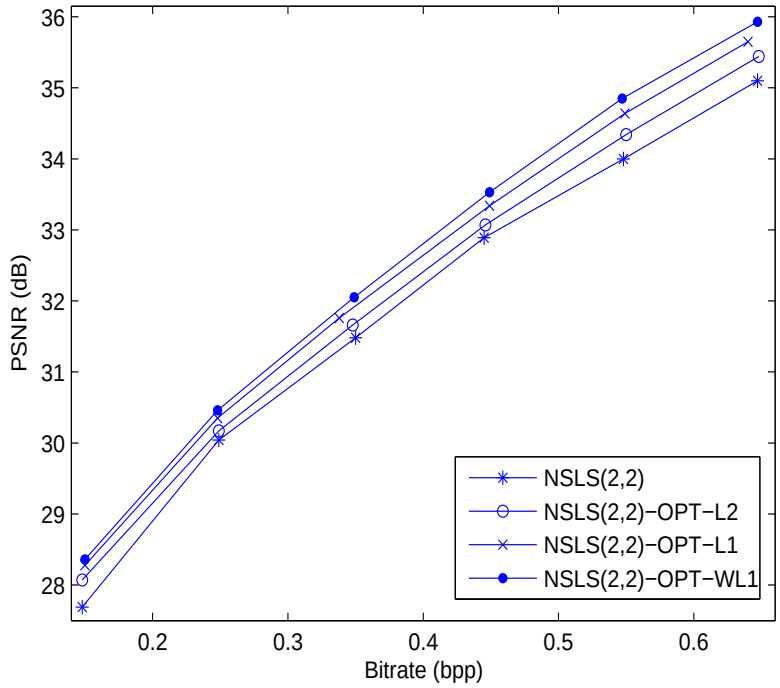



Figure 7.5: PSNR (in dB) versus the bitrate (bpp) after JPEG2000 progressive encoding for the "castle" image.

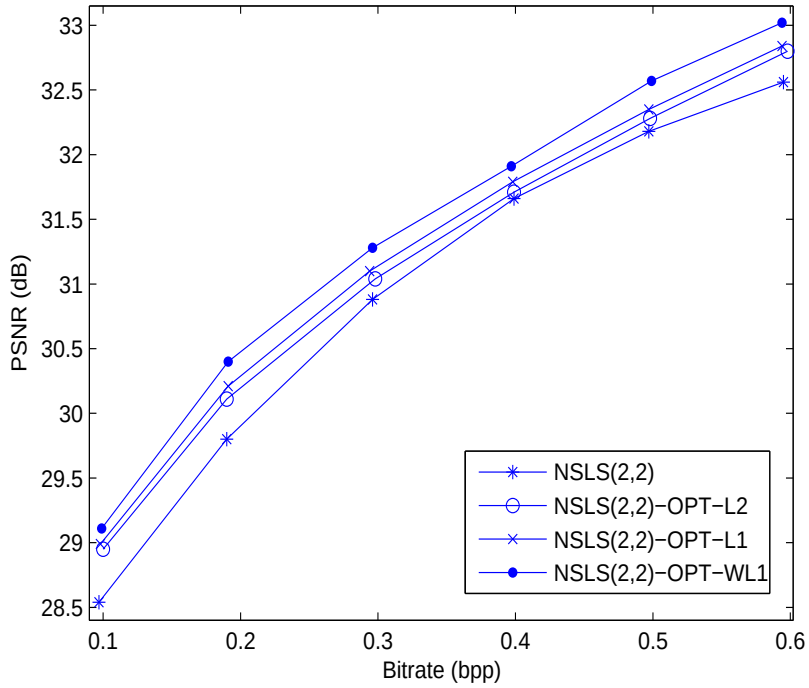



Figure 7.6: PSNR (in dB) versus the bitrate (bpp) after JPEG2000 progressive encoding for the "einst" image.

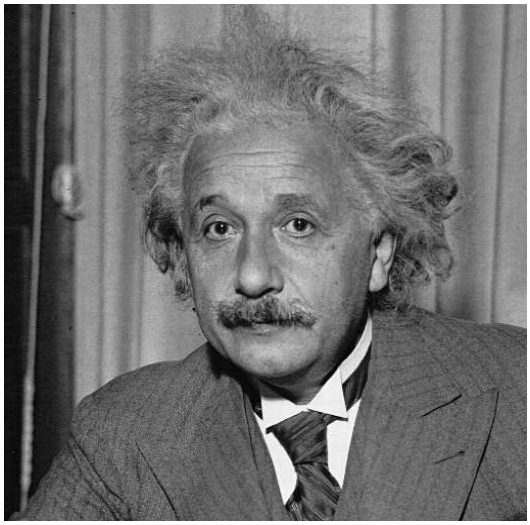
(a): Original image

(b): PSNR=30.44 dB, SSIM=0.844, VSNR=22.96 dB

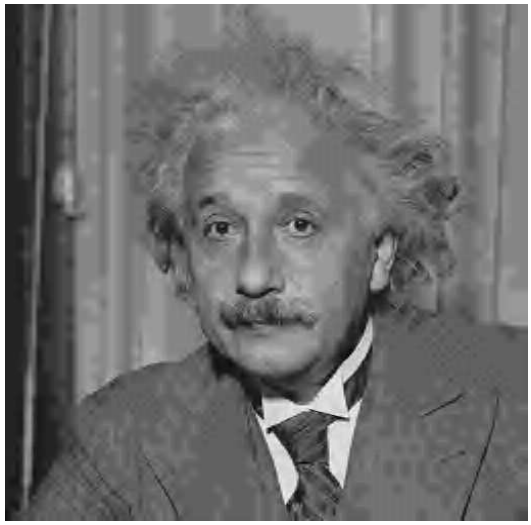
(c): PSNR=30.93 dB, SSIM=0.845, VSNR=23.46 dB

(d): PSNR=31.24 dB, SSIM=0.851, VSNR=24.06 dB

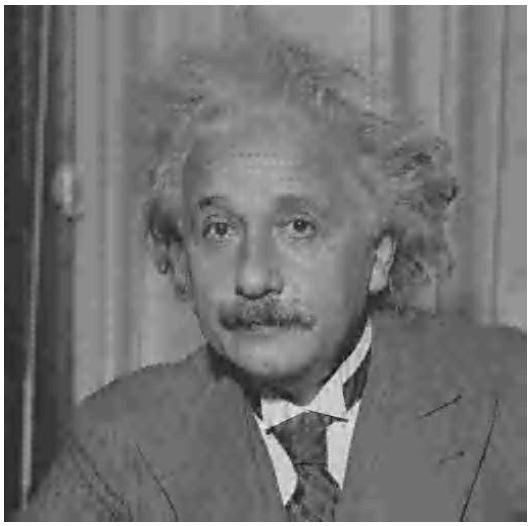
Figure 7.7: Reconstructed "lena" image at 0.15 bpp using: (b) NSLS(2,2) (c) NSLS(2,2)-OPT-L2 (d) NSLS(2,2)-OPT2-WL1.



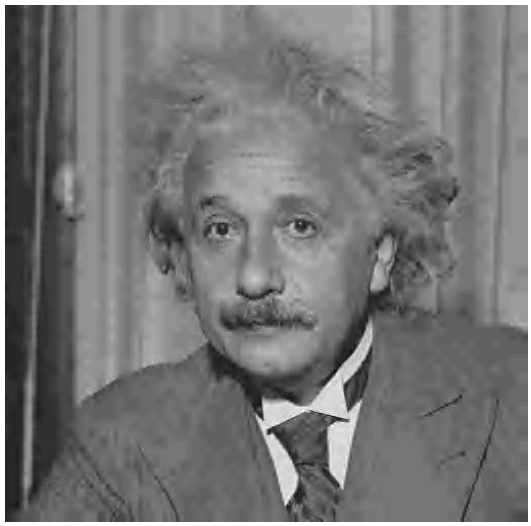
(a): Original image



(b): PSNR=28.55 dB, SSIM=0.648, VSNR=17.82 dB



(c): PSNR=28.94 dB, SSIM=0.649, VSNR=18.24 dB



(d): PSNR=29.11 dB, SSIM=0.654, VSNR=18.59 dB

Figure 7.8: Reconstructed "einst" image at 0.1 bpp using: (b) NSLS(2,2) (c) NSLS(2,2)-OPT-L2 (d) NSLS(2,2)-OPT2-WL1.

The performance of these optimization methods are also compared in stereo image coding context. More precisely, each method is applied to encode the reference image and the residual one. Figures 7.9 and 7.10 illustrate the evolution of the *average* PSNR (as defined in chapter 4) versus the *average* bitrate for the stereo image pairs "shrub" and "pentagon". An improvement of 0.05-0.4 dB is obtained by *jointly* optimizing the prediction filters.

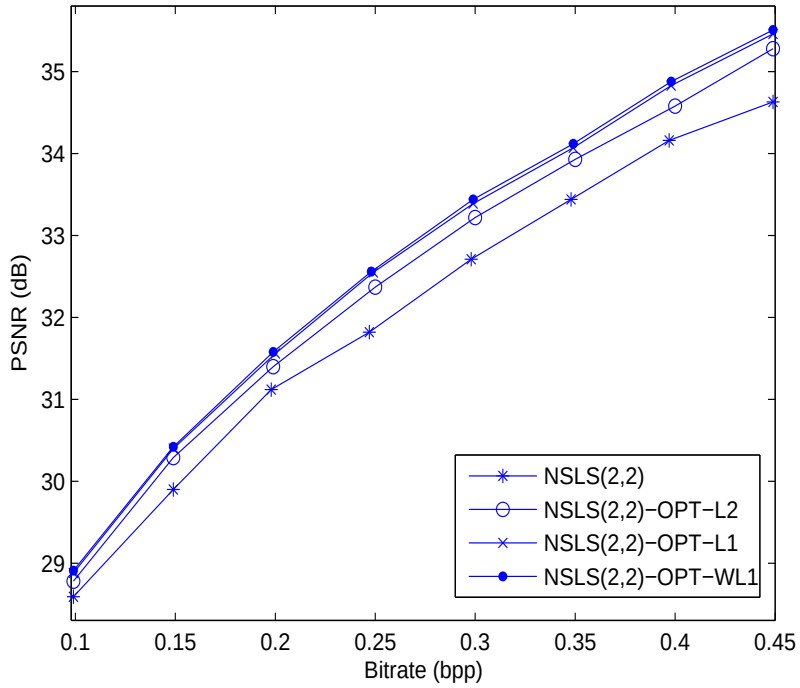



Figure 7.9: PSNR (in dB) versus the bitrate (bpp) after JPEG2000 progressive encoding for the "shrub" stereo images.

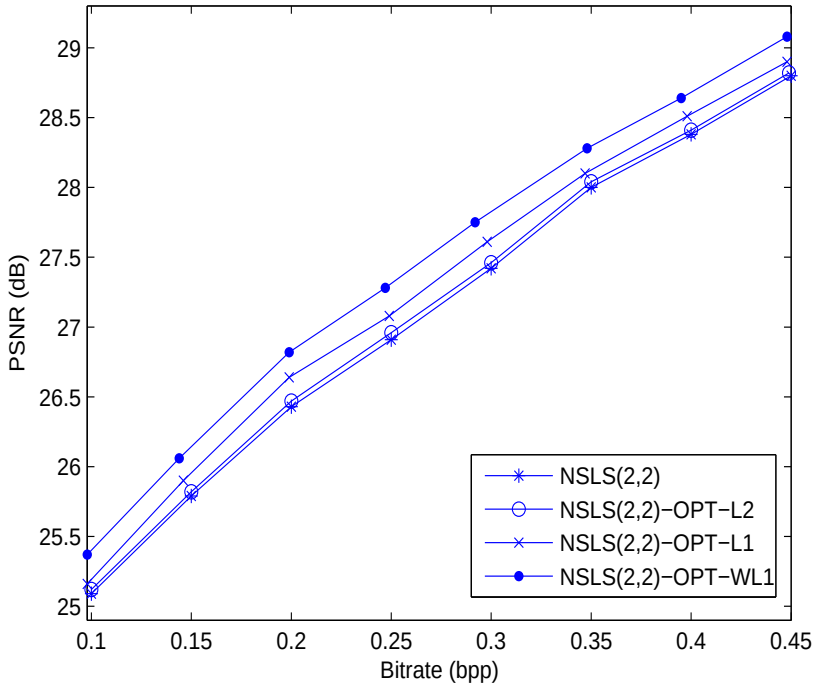



Figure 7.10: PSNR (in dB) versus the bitrate (bpp) after JPEG2000 progressive encoding for the "pentagon" stereo images.

Figure 7.11 displays the reconstructed target image of the "pentagon" stereo pair. It can be observed that the proposed *joint* optimization method leads to an improvement of 0.35 dB (resp. 0.016) in VSNR (resp. SSIM) compared with the decomposition in which the prediction filters are optimized *separately*.

(a): Original image

(b): PSNR=26.44 dB, SSIM=0.693, VSNR=12.17 dB

(c): PSNR=26.56 dB, SSIM=0.691, VSNR=12.49 dB

(d): PSNR=26.90 dB, SSIM=0.697, VSNR=13.06 dB

Figure 7.11: Reconstructed target image for the "pentagon" stereo images at 0.15 bpp using: (b) NSLS(2,2) (c) NSLS(2,2)-OPT-L2 (d) NSLS(2,2)-OPT2-WL1.

For completeness sake, the performance of the proposed method (NSLS(2,2)-OPT-WL1)

has also been compared with the 9/7 transform retained for the lossy mode of JPEG2000 standard. Table 7.2 shows the performance of the latter methods in terms of PSNR, SSIM and VSNR. Since the human eye cannot always distinguish the subjective image quality at middle and high bitrate, the results were restricted to the lower bitrate values.

Table 7.2: Performance of the proposed method vs the 9/7 transform

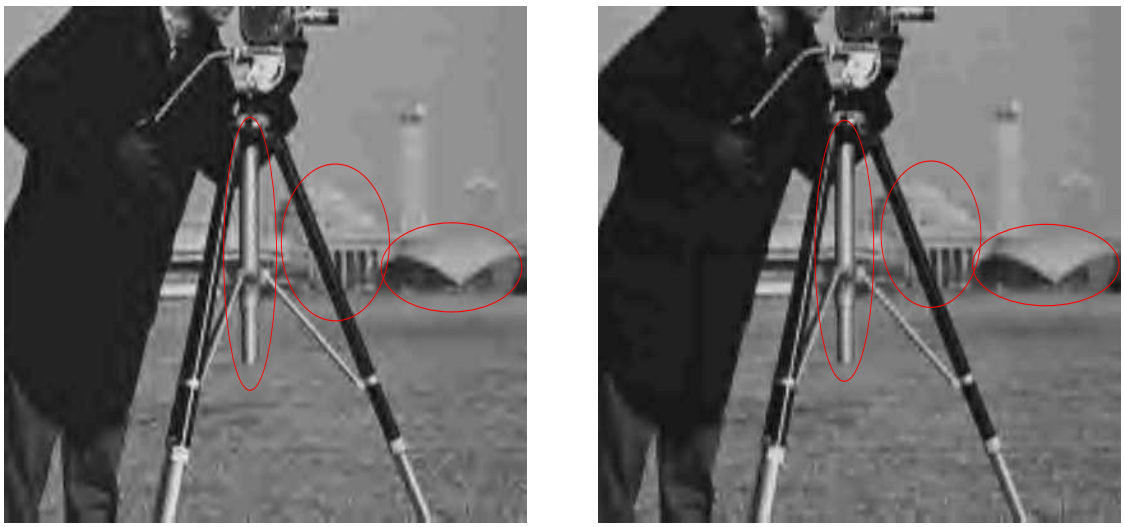
| | | 0.05 bpp | | 0.1 bpp | | 0.15 bpp | | 0.2 bpp | |
|---|---|---|---|---|---|---|---|---|---|
| | | NSLS(2,2)-OPT-WL1 | 9/7 | NSLS(2,2)-OPT-WL1 | 9/7 | NSLS(2,2)-OPT-WL1 | 9/7 | NSLS(2,2)-OPT-WL1 | 9/7 |
| elaine | PSNR | **27.85** | 27.06 | **30.25** | 30.22 | 31.23 | **31.31** | 31.76 | **31.92** |
| | SSIM | **0.669** | 0.646 | **0.716** | 0.713 | **0.739** | 0.738 | 0.754 | **0.756** |
| | VSNR | **18.44** | 16.84 | **23.10** | 22.78 | **25.60** | 25.41 | 27.28 | **27.42** |
| castle | PSNR | **25.10** | 24.69 | **27.08** | 26.96 | 28.36 | **28.41** | 29.51 | **29.58** |
| | SSIM | **0.725** | 0.701 | **0.790** | 0.775 | **0.825** | 0.819 | **0.855** | 0.851 |
| | VSNR | **17.54** | 16.46 | **21.55** | 20.86 | **23.74** | 23.17 | **25.80** | 25.32 |
| einst | PSNR | **27.51** | 27.50 | 29.12 | **29.18** | 29.92 | **30.06** | 30.50 | **30.70** |
| | SSIM | **0.603** | 0.596 | **0.653** | 0.652 | 0.687 | 0.687 | 0.710 | **0.715** |
| | VSNR | **15.33** | 15.17 | **18.62** | 18.48 | 20.37 | **20.40** | 21.59 | **21.94** |
| lena | PSNR | **26.70** | 26.33 | **29.59** | 29.50 | 31.22 | **31.42** | 32.70 | **32.90** |
| | SSIM | **0.747** | 0.725 | **0.818** | 0.807 | **0.850** | 0.850 | 0.871 | **0.873** |
| | VSNR | **15.95** | 15.18 | **20.56** | 20.03 | **24.05** | 23.79 | 26.12 | **26.15** |
| cameraman | PSNR | **26.51** | 26.34 | 29.81 | **30.28** | 31.84 | **32.63** | 33.61 | **34.44** |
| | SSIM | **0.783** | 0.767 | **0.847** | 0.839 | 0.887 | **0.892** | 0.914 | **0.915** |
| | VSNR | **16.74** | 16.14 | **21.73** | 21.61 | 24.94 | **25.70** | 27.75 | **28.34** |
| boat | PSNR | **24.65** | 24.55 | 26.82 | **26.86** | 28.43 | **28.54** | 29.52 | **29.74** |
| | SSIM | **0.675** | 0.661 | **0.753** | 0.746 | **0.806** | 0.802 | **0.837** | 0.836 |
| | VSNR | **13.41** | 13.03 | **17.14** | 16.89 | **20.24** | 19.76 | **22.19** | 21.89 |

While the proposed method is less performant in terms of PSNR than the 9/7 transform for some images, better results are obtained in terms of perceptual quality. For instance, Figures 7.12-7.16 illustrate some reconstructed images. It can be observed that the proposed method (NSLS(2,2)-OPT-WL1) achieves a gain of about 0.12-1.6 dB (resp. 0.004-0.023)

in terms of VSNR (resp. SSIM).

(a): PSNR=28.54 dB, SSIM=0.802, VSNR=19.76 dB (b): PSNR=28.43 dB, SSIM=0.806, VSNR=20.24 dB

Figure 7.12: Zoom applied on the reconstructed "boat" image at 0.1 bpp using: (a) 9/7 transform ; (b) NSLS(2,2)-OPT-WL1 .



(a): PSNR=30.28 dB, SSIM=0.839, VSNR=21.61 dB (b): PSNR=29.81 dB, SSIM=0.847, VSNR=21.73 dB

Figure 7.13: Zoom applied on the reconstructed "cameraman" image at 0.1 bpp using: (a) 9/7 transform; (b) NSLS(2,2)-OPT-WL1 .

(a): PSNR=26.33 dB, SSIM=0.725, VSNR=15.18 dB (b): PSNR=26.70 dB, SSIM=0.747, VSNR=15.95 dB

Figure 7.14: Zoom applied on the reconstructed "lena" image at 0.05 bpp using: (a) 9/7 transform ; (b) NSLS(2,2)-OPT-WL1 .

(a): PSNR=29.50 dB, SSIM=0.807, VSNR=20.03 dB (b): PSNR=29.59 dB, SSIM=0.818, VSNR=20.56 dB

Figure 7.15: Zoom applied on the reconstructed "lena" image at 0.1 bpp using: (a) 9/7 transform ; (b) NSLS(2,2)-OPT-WL1.

(a): PSNR=27.06 dB, SSIM=0.646, VSNR=16.84 dB (b): PSNR=27.85 dB, SSIM=0.669, VSNR=18.44 dB

Figure 7.16: Zoom applied on the reconstructed "elaine" image at 0.05 bpp using: (a) 9/7 transform ; (b) NSLS(2,2)-OPT-WL1.

(a): PSNR=30.22 dB, SSIM=0.713, VSNR=22.78 dB (b): PSNR=30.25 dB, SSIM=0.716, VSNR=23.10 dB

Figure 7.17: Zoom applied on the reconstructed "elaine" image at 0.1 bpp using: (a) 9/7 transform ; (b) NSLS(2,2)-OPT-WL1.

Furthermore, Figures 7.18 and 7.19 display the reconstructed target image for the stereo image pairs "shrub" and "pentagon". While NSLS(2,2)-OPT-WL1 and 9/7 transform show

similar visual quality for the "pentagon" pair, the proposed method leads to better quality of reconstruction than the 9/7 transform for the "shrub" stereo images.

(a): PSNR=28.68 dB, SSIM=0.682, VSNR=19.27 dB (b): PSNR=28.76 dB, SSIM=0.698, VSNR=19.63 dB

Figure 7.18: Zoom applied on the reconstructed target image for the "shrub" stereo images at 0.1 bpp using: (a) 9/7 transform ; (b) NSLS(2,2)-OPT-WL1.

(a): PSNR=27.11 dB, SSIM=0.706, VSNR=13.12 dB (b): PSNR=26.90 dB, SSIM=0.697, VSNR=13.06 dB

Figure 7.19: Zoom applied on the reconstructed target image for the "pentagone" stereo images at 0.15 bpp using: (a) 9/7 transform ; (b) NSLS(2,2)-OPT-WL1.

## 7.7 Conclusion

In this chapter, we have studied different optimization techniques for the design of filters in a NSLS structure. A new criterion has been presented for the optimization of the prediction filters in this context. The idea consists of *jointly* optimizing these filters by minimizing iteratively a weighted $\ell_1$ criterion. Experimental results carried out on still images and stereo images pair have illustrated the benefits which can be drawn from the proposed optimization technique.

# Chapter 8

# Conclusion and future work

In the context of an increased interest for stereoscopic images in many application fields such as entertainment and remote sensing, this thesis has been mainly devoted to the design and analysis of new and efficient coding schemes for stereo images. Furthermore, another goal of this work was to design adaptive decompositions well adapted to the characteristics of the images. In the following, we summarize the main contributions of this work while mentioning some possible future extensions at the end of each paragraph.

The first contribution of this research work is the development of new methods, based on Vector Lifting Schemes (VLS), for stereo image coding purpose. More precisely, unlike the most commonly used approaches which consist of encoding a reference image (say the left one) and a residual one as well as a disparity map, the proposed schemes use a joint multiscale decomposition which has the advantage of producing two compact multiresolution representations of the left and the right views. In this context, we have provided two examples of structures denoted by VLS-I and VLS-II. The first scheme (i.e. VLS-I) relies on a basic prediction-update structure. Despite its simplicity, one drawback of VLS-I is that it generates an update leakage effect in the sense that the information coming from the left image, which is used for the prediction of the right one, is also used, through the update step, to compute the approximation coefficients of the right image. To alleviate this shortcoming, we have proposed an alternative scheme given by a predict-update-predict structure (i.e. VLS-II). A theoretical analysis in terms of prediction error variance was conducted in order to show the benefits of the underlying VLS structure. Experimental results have shown that VLS based decompositions significantly reduce the problem of

blocking artifacts which appears at low bitrates with conventional residual image based coding methods.
The proposed method still requires further investigations. Indeed, the encoding performance of the VLS can be improved by better taking into account the effect of occlusions. More precisely, a hybrid coding scheme can be designed where the occluded blocks (resp. non occluded ones) are encoded using an intra (resp. inter) prediction step in the previous VLS structures. Such method will require to transmit a side information corresponding to the block type, that may limit its performance especially at low bitrates. Thus, it will be interesting to use a technique that allows the decoder to detect the occluded blocks without any overhead cost.
Furthermore, the proposed VLS structure can be extended to a multiview/video coding framework. Note that rate-distortion techniques developed in video coding could also be used in conjunction with our schemes.

The second contribution of this work was to integrate a dense disparity estimation technique in joint stereo image coding schemes. Unlike conventional methods that employ a block-based disparity estimation technique, our method uses a dense and *smooth* disparity map. However, in a coding application context, the problem of coding the resulting dense disparity is a challenging issue because of the high bitrate needed to store/transmit such a field. For this reason, we have proposed to resort to a quadtree decomposition and entropy encoding that achieve a good tradeoff between the accuracy of the disparity representation and the coding cost. Experimental results, carried out on stereo and multiview video sequences, have shown the improvements achieved by this approach.
Considering an alternative of segmenting the dense disparity map requires further work on the improvement of the coding efficiency in terms of bitrate and quality of the reconstructed image. Furthermore, the smoothness of the disparity field can be exploited to develop a fully scalable stereo image coder. In other words, instead of using a DPCM technique to encode the disparity map, a progressive coding technique can also be employed. Indeed, as it has been recently investigated in scalable video coding, the disparity representation in a scalable strategy can help to improve the encoding performance at low bitrates.

The third contribution of this research work has been focused on the optimization of all the operators involved in lifting schemes (LS). As mentioned before, an important limitation

of conventional separable LS is that they may not appear very efficient to cope with the two dimensional characteristics of contours which are neither horizontal nor vertical. For this reason, we have proposed to perform the optimization techniques on a two-dimensional non separable lifting scheme (NSLS). The considered structure is composed of three prediction lifting steps followed by an update lifting step. Concerning the predictions filters, each one is optimized by minimizing the detail signal variance, which is a standard criterion. The optimization of the update filter constitutes the main contribution of this part. More precisely, we have designed an update filter that aims at reducing the aliasing effects. Thus, it is optimized by minimizing the difference between its output and the output of an ideal low-pass filter. Moreover, we have shown that the proposed optimization method leads to the same optimal update filter when the optimization is performed either before or after the decimation step. Experimental results have firstly indicated that such a method improves the encoding performance compared with structures where only the prediction filters are optimized. Secondly, compared with the state-of-the-art update optimization method based on the minimization of the reconstruction error, the proposed technique performs similarly in terms of quality of reconstruction. However, we have shown that our method presents the advantage of reducing significantly the complexity of the optimization algorithm. Furthermore, an additional advantage of our work was to derive theoretical expressions of the optimal filter coefficients based on the correlation factors of the input image. This may be interesting in the implementation process of our optimization method. While the considered autoregressive model is separable, it will be interesting to extend the theoretical analysis by using a more sophisticated model which takes into account the directional structures in the image. Furthermore, a local adaptive approach, where the filters are optimized for different regions of the image, could be investigated.

As a continuation of this work, the last contribution of this thesis has been devoted to investigating sparse optimization criteria to improve the encoding performance. For this purpose, we have firstly introduced the motivation of using an $\ell_1$ criteria instead of $\ell_2$ ones. Then, unlike the previous optimization techniques where each prediction filter was separately optimized by minimizing the $\ell_2$-norm of each detail subband, we have proposed to design these filters by minimizing the $\ell_1$-norm of each detail subband. Based on the fact that the output of a prediction filter can be used as an input for another prediction

filter, we have proposed to optimize such a filter by minimizing a global prediction error which is a *weighted* sum of the $\ell_1$-norms of the three resulting detail subbands. Moreover, we have seen that the optimization of a prediction filter depends on that of the remaining filters and vice-versa. To solve this problem, we have proposed an iterative algorithm that *jointly* optimizes the different prediction filters. Our simulations, carried out on still and residual images, have shown the benefits of using the $\ell_1$ minimization technique and the proposed joint optimization method. These results motivate us to investigate further ideas in this direction.

A first important area of research may consider the extensions of the proposed optimization methods to other lifting structures. Indeed, up to now, we have focused on the optimization of the filters involved in a one stage lifting scheme (i.e a P-P-P-U non separable structure as depicted in Figure 7.3). Recently, it was shown in [Iwahashi, Kiya, 2010] that the 9/7 transform can be put in a non separable way by using a cascade of the previous structure yielding a P-P-P-U-P-P-P-U structure. Thus, it will be interesting to extend this optimization technique to lifting schemes with more than one stage like the 9/7 transform and the proposed VLS-II decomposition.

While the considered non separable structure consists of 4 branches containing one approximation and three detail subbands, further improvements may be achieved by adding other channels that take into account the variety of directions in images. However, an important limitation of such a structure is the introduction of redundancy which may affect the compression efficiency. To reduce this problem, coding techniques should be investigated as performed in [Reeves, Kingsbury, 2002; Fowler *et al.*, 2007].

Another important area of research is to investigate more efficient optimization methods for lifting design problems. As discussed in Section 7.2, we optimized in this work some sparse criteria which are related to the bitrate of the quantized signal. However, it will be interesting to minimize such criteria subject to some rate-distortion constraints.

# Bibliography

Abhayaratne, G. C. K. (2004). 2D wavelet transforms with a spatially adaptive 2D low-pass filter. *In the 6th Nordic Signal Processing Symposium*, Espoo, Finland.

Abhayaratne, G. C. K., Piella, G., Heijmans, H., Pesquet-Popescu, B. (2003). Adaptive integer to integer wavelet transforms using update lifting. *In Wavelet Applications in Signal and Image Processing X, SPIE*, vol. 5207, pages 813–824, San Diego, CA, USA.

Ahlvers, U., Zoelzer, U., Rechmeier, S. (2003). FFT-based disparity estimation for stereo image coding. *In IEEE International Conference on Image Processing*, vol. 1, pages 761–764, Barcelona, Spain.

Alkanhal, M., Turaga, D., Chen, T. (1999). Correlation based search algorithms for motion estimation. *In Picture Coding Symposium*, pages 99–102, Portland.

Alvarez, L., Deriche, R., Sanchez, J., Weickert, J. (2002). Dense disparity map estimation respecting image discontinuities: A PDE and scale-space based approach. *Journal of Visual Communication and Image Representation*, 13:3–21.

Anantrasitichai, N., Canagarajah, C. N., Bull, D. R. (2005). Multi-view image coding with wavelet lifting and in-band disparity compensation. *In IEEE International Conference on Image Processing*, pages 33–36, Genoa, Italy.

Arnavut, Z., Narumalani, S. (1996). Application of permutations to lossless compression of multispectral thematic mapper images. *Optical Engineering*, 35:3442–3448.

Aydinoglu, H., Hayes, M. (1995). Stereo image coding. *In International Symposium on Circuits and Systems*, vol. I, pages 247–250, Washington, USA.

Bekkouche, H., Barret, M., Oksman, J. (2008). Adapted generalized lifting schemes for scalable lossless image coding. *Signal Processing*, 88(11):2790–2803.

Belzer, B., Lina, J.-M., Villasenor, J. (1995). Complex, linear-phase filters for efficient image coding. *IEEE Transactions on Signal Processing*, 43(10):2425–2427.

Benazza-Benyahia, A., Pesquet, J.-C. (2002). Vector-lifting schemes based on sorting techniques for lossless compression of multispectral images. *In SPIE Conference on Mathematics of Data/Image Coding, Compression and Encryption IV, with Applications*, vol. 4793, pages 42–51, Seattle, WA, USA.

Benazza-Benyahia, A., Pesquet, J.-C., Hamdi, M. (2002). Vector lifting schemes for lossless coding and progressive archival of multispectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 40(9):2011–2024.

Benazza-Benyahia, A., Pesquet, J.-C., Hattay, J., Masmoudi, H. (2007). Block-based adaptive vector lifting schemes for multichannel image coding. *EURASIP International Journal of Image and Video Processing*, 2007(1):10 pages.

Boettcher, J. B., Fowler, J. E. (2007). Video coding using a complex wavelet transform and set partitioning. *IEEE Signal Processing Letters*, 14(9):633–636.

Boufama, B., Jin, K. (2002). Towards a fast and reliable dense matching algorithm. *In International Conference on Vision Interface*, pages 178–185, Calgary, Canada.

Boulgouris, N. V., Strintzis, M. G. (1999). Reversible multiresolution image coding based on adaptive lifting. *In IEEE International conference on Image Processing*, vol. 3, pages 546–550, Kobe, Japan.

Boulgouris, N. V., Strintzis, M. G. (2002). A family of wavelet-based stereo image coders. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(10):898–903.

Boulgouris, N. V., Tzovaras, D., Strintzis, M. G. (2001). Lossless image compression based on optimal prediction, adaptive lifting, and conditional arithmetic coding. *IEEE Transaction on Image Processing*, 10(1):1–14.

Bovik, A. C. (2000). *Handbook of Image and Video Processing.* Academic Press, San Diego, CA.

Boykov, Y., Veksler, O., Zabih., R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

Cagnazzo, M., Castaldo, F., André, T., Antonini, M., Barlaud, M. (2007). Optimal motion estimation for wavelet motion compensated video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(7):11.

Cai, T. T., Xu, G., Zhang, J. (2009). On recovery of sparse signal via $\ell_1$ minimization. *IEEE Transactions on Information Theory*, 55(7):3388–3397.

Calderbank, A. R., Daubechies, I., Sweldens, W., Yeo, B.-L. (1998). Wavelet transforms that map integers to integers. *Applied and Computational Harmonic Analysis*, 5(3):332–369.

Candes, J. E., Donoho, L. D. (2004). New tight frames of curvelets and optimal representations of objects with piecewise $C^2$ singularities. *Communications on pure and applied mathematics*, 57(2):219–266.

Chabrier, S., Emile, B., Rosenberger, C. (2005). Texture detection for image analysis. *In International Conference on Advances in Pattern Recognition*, vol. 3687, pages 455–463, Berlin.

Chandler, D. M., Hemami, S. S. (2007). VSNR: A wavelet-based visual signal-to-noise ratio for natural images. *IEEE Transactions on Image Processing*, 16(9):2284–2298.

Chang, C.-L., Girod, B. (2007). Direction adaptive discrete wavelet transform for image compression. *IEEE Transactions on Image Processing*, 16(5):1289–1302.

Chappelier, V., Guillemot, C. (2006). Oriented wavelet transform for image compression and denoising. *IEEE Transactions on Image Processing*, 15(10):2892–2903.

Chappelier, V., Guillemot, C., Marinkovic, S. (2004). Image coding with iterated contourlet and wavelet transforms. *In International Conference on Image Processing*, Singapore.

Chaux, C., Combettes, P., Pesquet, J.-C., Wajs, V. (2007). A variational formulation for framebased inverse problems. *Inverse Problems*, 23(4):1495–1518.

Chaux, C., Duval, L., Pesquet, J.-C. (2005). 2D dual-tree M-band wavelet decomposition. *In IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pages 537–540, Philadelphia, USA.

Chen, Y., Wang, Y.-K., K.Ugur, Hannuksela, M.-M., Lainema, J., Gabbouj, M. (2009). The emerging MVC standard for 3D video services. *EURASIP Journal on Advances in Signal Processing*, 2009(8):1–13.

Chokchaitam, S. (2004). A non-separable two-dimensional LWT for an image compression and its theoretical analysis. *Thammasat International Journal of Science and Technology*, 9(1):35–43.

Claypoole, R. L., Davis, G., Sweldens, W., Baraniuk, R. G. (1997). Nonlinear wavelet transforms for image coding. *In the 31st Asilomar Conference on Signals, Systems and Computers*, vol. 1, pages 662–667.

Claypoole, R. L., Davis, G. M., Sweldens, W., Baraniuk, R. G. (2003). Nonlinear wavelet transforms for image coding via lifting. *IEEE Transactions on Image Processing*, 12(12):1449–1459.

Cohen, A., Daubechies, I., Feauveau, J. C. (1992). Biorthogonal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 45(5):485–560.

Combettes, P. L. (1993). The foundations of set theoretic estimation. *Proceedings of the IEEE*, 81(2):182–208.

Combettes, P. L. (2003). A block-iterative surrogate constraint splitting method for quadratic signal recovery. *IEEE Transactions on Signal Processing*, 51(7):1771–1782.

Combettes, P. L., Pesquet, J.-C. (2004). Image restoration subject to a total variation constraint. *IEEE Transactions on Image Processing*, 13(9):1213–1222.

Combettes, P. L., Pesquet, J.-C. (2007). A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Journal Selected Topics Signal Processing*, 1:564–574.

Combettes, P. L., Pesquet, J.-C. (2010). Proximal splitting methods in signal processing. *In* Bauschke, H. H., Burachik, R., Combettes, P. L., Elser, V., Luke,

D. R., Wolkowicz, H., éditeurs : *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer-Verlag, New York.

Combettes, P. L., Wajs, V. R. (2005). Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation*, 4(4):1168–1200.

Cover, T. M., Thomas, J. A. (1991). *Elements of information theory*. Wiley-Interscience, New York, USA.

Cutler, C. C. (1950). *Differential Quantization of Communication Signals*. first filling date June 29, 1950, U.S. patent number 2605361.

Darazi, R., Gouze, A., Macq, B. (2009). Adaptive lifting scheme-based method for joint coding 3D-stereo images with luminance correction and optimized prediction. *In IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 917–920, Taipei.

Daribo, I., Kaaniche, M., Miled, W., Cagnazzo, M., Pesquet-Popescu, B. (2009). Dense disparity estimation in multiview video coding. *In International Workshop on Multimedia Signal Processing*, pages 1–6, Rio de Janeiro, Brazil.

Daribo, I., Tillier, C., Pesquet-Popescu, B. (2008). Adaptive wavelet coding of the depth map for stereoscopic view synthesis. *In International Workshop on Multimedia Signal Processing*, pages 413–417, Cairns, Australia.

Daubechies, I. (1992). *Ten Lectures on Wavelets*, vol. 61. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Daubechies, I., Sweldens, W. (1998). Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4(3):247–269.

Deever, A., Hemami, S. S. (1998). Dense motion field reduction for motion estimation. *In Asilomar Conference on Signals, Systems and Computers*, vol. 2, pages 944–948, Pacific Grove, CA, USA.

Dhond, V. R., Aggarwal, J. K. (1989). Structure from stereo - a review. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1489–1510.

Ding, W., Wu, F., Wu, X., Li, S., Li, H. (2007). Adaptive directional lifting-based wavelet transform for image coding. *IEEE Transactions on Image Processing*, 10(2):416–427.

Do, M. N., Vetterli, M. (2002). Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. *IEEE Transactions on Image Processing*, 11(2):146–158.

Do, M. N., Vetterli, M. (2005). The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on Image Processing*, 14(12): 2091–2106.

Donoho, D. (2006). Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306.

Drémeau, A., Herzet, C., Guillemot, C., Fuchs, J.-J. (2010a). Anisotropic multi-scale sparse learned bases for image compression. *In SPIE, Image and Video Coding I*, vol. 7543, San Jose.

Drémeau, A., Herzet, C., Guillemot, C., Fuchs, J.-J. (2010b). Sparse optimization with directional DCT bases for image compression. *In IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1290–1293, Dallas.

Eckstein, J., Bertsekas, D. P. (1992). On the Douglas-Rachford splitting methods and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55:293–318.

Edward, J. D., Kashyap, R. L., Mitchell, O. R. (1979). Image data compression using autoregressive time series models. *Pattern Recognition*, 11(5-6):313–323.

Ellinas, J., Sangriotis, M. S. (2006). Morphological wavelet-based stereo image coders. *Journal of Visual Communication and Image Representation*, 17(4):686–700.

Faugeras, O. (1994). *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA.

Feldmann, I., Waizenegger, W., Atzpadin, N., Schreer, O. (2010). Real-time depth estimation for immersive 3D videoconferencing. *In 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video*, pages 1–4, Tampere.

Fowler, J. E. (2003). Embedded wavelet-based image compression: state of the art. *Information Technology*, 45(5):256–262.

Fowler, J. E., Boettcher, J. B., Pesquet-Popescu, B. (2007). Image coding using a complex dual-tree wavelet transform. *In the European Signal Processing Conference*, page 5 pages, Poznan, Poland.

Fowler, J. E., Pesquet-Popescu, B. (2007). An overview on wavelets in source coding, communications, and networks. *EURASIP Journal on Image and Video Processing*, 2007:27 pages.

Frajka, T., Zeger, K. (2003). Residual image coding for stereo image compression. *Optical Engineering*, 42(1):182–189.

Fusiello, A., Trucco, E., Verri, A. (2000). A compact algorithm for rectification of stereo pairs. *International Journal of Machine Vision and Applications*, 12(1):16–22.

G. W. Bothwell, E. G. Hansen, R. E. V., Miller, K. C. (2002). The multi-angle imaging spectroradiometer science data system, its products, tools, and performance. *IEEE Transactions on Geoscience and Remote Sensing*, 40(7):1467–1476.

Gerek, O. N., Çetin, A. E. (2000). Adaptive polyphase subband decomposition structures for image compression. *IEEE Transactions on Image Processing*, 9(10):1649–1660.

Gerek, O. N., Çetin, A. E. (2006). A 2D orientation-adaptive prediction filter in lifting structures for image coding. *IEEE Transactions on Image Processing*, 15(1):106–111.

Gersho, A., Gray, R. M. (1993). *Vector Quantization and Signal Compression*. MA: Kluwer.

Gish, H., Pierce, J. N. (1969). Asymptotically efficient quantizing. *IEEE Transactions on Information Theory*, 14(5):676–683.

Gouze, A., Antonini, M., Barlaud, M., Macq, B. (2001). Optimized lifting scheme for two-dimensional quincunx sampling images. *In IEEE International Conference on Image Processing*, vol. 2, pages 253–258, Thessaloniki, Greece.

Gouze, A., Antonini, M., Barlaud, M., Macq, B. (2004). Design of signal-adapted multidimensional lifting schemes for lossy coding. *IEEE Transactions on Image Processing*, 13(12):1589–1603.

Hampson, F. J., Pesquet, J.-C. (1996). A nonlinear subband decomposition with perfect reconstruction. *In IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pages 1523–1526, Atlanta, GA, USA.

Hampson, F. J., Pesquet, J.-C. (1998). M-band nonlinear subband decompositions with perfect reconstruction. *IEEE Transactions on Image Processing*, 7(11):1547–1560.

Hartley, R., Gupta, R. (1993). Computing matched-epipolar projections. *In IEEE Conference on Computer Vision and Pattern Recognition*, pages 549–555, New York, United States.

Hartley, R. I., P. Sturn, P. (1997). Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157.

Heijmans, H., Piella, G., Pesquet-Popescu, B. (2002). Building adaptive 2D wavelet decompositions by update lifting. *In IEEE International Conference on Image Processing*, vol. 1, pages 397–400, Rochester, New York, USA.

Heijmans, H. J. A. M., Pesquet-Popescu, B., Piella, G. (2005). Building nonredundant adaptive wavelets by update lifting. *Applied and Computational Harmonic Analysis*, 18:252–281.

Herpers, R., Verghese, G., Derpanis, K., McCready, R., MacLean, J., Levin, A., Topalovic, D., Wood, L., Jepson, A., Tsotsos, J. K. (1999). Detection and tracking of faces in real environments. *In Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, page 96.

Hiriart-Urruty, J.-B., Lemaréchal, C. (1993). *Convex analysis and minimization algorithms*. Springer-Verlag, Berlin, London.

Hsiang, S.-T., Woods, J. W. (2000). Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling. *In IEEE International Symposium on Circuits and Systems*, vol. 3, pages 662–665, Geneva, Switzerland.

Hua, Y., Liu, W. (1998). Generalized karhunen-loeve transform. *IEEE Signal Processing Letters*, 5(6):141–142.

ITU-T Recommentation H.264 & ISO/IEC 14496-10 AVC (2003). Advanced video coding for generic audio-visual services.

Iwahashi, M., Kiya, H. (2010). A new lifting structure of non separable 2D DWT with compatibility to JPEG 2000. *In IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1306–1309, Dallas, Texas, USA.

Jayant, N. S., Noll, P. (1984). *Digital Coding of Waveforms.* Englewood Cliffs, N.J: Prentice-Hall.

Kaaniche, M., Benazza-Benyahia, A., Pesquet-Popescu, B., Pesquet, J.-C. (2007). Lifting schemes for joint coding of stereoscopic pairs of satellite images. *In European Signal and Image Processing Conference*, pages 981–984, Poznan, Poland.

Kaaniche, M., Benazza-Benyahia, A., Pesquet-Popescu, B., Pesquet, J.-C. (2009a). Vector lifting schemes for stereo image coding. *IEEE Transactions on Image Processing*, 18(11):2463–2475.

Kaaniche, M., Benazza-Benyahia, A., Pesquet-Popescu, B., Pesquet, J.-C. (2011). Non separable lifting scheme with adaptive update step for still and stereo image coding. *Elsevier Signal Processing: Special issue on Advances in Multirate Filter Bank Structures and Multiscale Representations.*

Kaaniche, M., Pesquet, J.-C., Benazza-Benyahia, A., Pesquet-Popescu, B. (2010). Two-dimensional non separable adaptive lifting scheme for still and stereo image coding. *In IEEE International Conference on Acoustics, Speech and Signal Processing*, Dallas, Texas, USA.

Kaaniche, M., Pesquet-Popescu, W. M. B., Benazza-Benyahia, A., Pesquet, J.-C. (2009b). Dense disparity map representations for stereo image coding. *In IEEE International Conference on Image Processing*, pages 725–728, Cairo, Egypt.

Kanade, T., Okutomi, M. (1994). A stereo matching algorithm with an adaptive window : Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932.

Kay, S. M. (1993). *Fundamentals of statistical signal processing.* Englewood Cliffs, N.J: Prentice-Hall.

Kim, C., Lee, K., Choi, B., Lee, S. (2005). A dense stereo matching using two-pass dynamic programming with generalized ground control points. *In IEEE Conference Proceedings of Computer Vision and Pattern Recognition*, vol. 2, pages 1075–1082, San Diego, United States.

Kingsbury, N. G. (2001). Complex wavelets for shift invariant analysis and filtering of signals. *Journal of Applied Computational Harmonic Analysis*, 10:234–253.

Kolmogorov, V., Zabih, R. (2001). Computing visual correspondence with occlusions using graph cuts. *In IEEE International Conference on Computer Vision*, vol. 2, pages 508–515, Vancouver, BC, Canada.

Kosov, S., Thormählen, T., Seidel, H.-P. (2009). Accurate real-time disparity estimation with variational methods:. *In International Symposium on Visual Computing*, vol. 5875, pages 796–807, Las Vegas, United States.

Lau, R., Ince, S., Konrad, J. (2007). Compression of still multiview images for 3-D automultiscopic spatially-multiplexed displays. *In SPIE Conference on Stereoscopic Displays & Virtual reality Systems*, vol. 6490, pages 0O1–0O9, San-Jose, CA, USA.

Liu, Y., Ngan, K. N. (2008). Weighted adaptive lifting-based wavelet transform for image coding. *IEEE Transactions on Image Processing*, 17(4):500–511.

LoPresto, S. M., Ramchandran, K., Orchard, M. T. (1997). Image coding based on mixture modeling of wavelet coefficients and a fast estimation quantization framework. *In Data Compression Conference*, pages 221–230, Snowbird, USA.

Lukacs, M. E. (1986). Predictive coding of multi-viewpoint image sets. *In IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 11, pages 521–524, Tokyo, Japan.

Maalouf, A., Larabi, M.-C. (2010). Bandelet-based stereo image coding. *In IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 698–701, Dallas, Texas, United States.

Mallat, S. (1989). A theory for multiresolution signal decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693.

Mallat, S. (1998). *A wavelet tour of signal processing.* Academic Press, San Diego.

Mallat, S. (2009). Geometrical grouplets. *Applied and Computational Harmonic Analysis*, 26(2):161–180.

Malvar, H. S. (1992). *Signal Processing with Lapped Transforms.* Artech House, Norwood, MA, USA.

Marangaz, A. M., Buyuksalih, G., Buyuksalih, I., Serfercik, U. G., Akcin, H. (2005). Geometric evaluation automated DEM and orthoimage generation from along-track stereo ASTER images. *In International Conference on Recent Advances in Space Technologies*, pages 505–510, Istanbul, Turkey.

Marvell, L. M., Khayrallah, A. S., Boncelet, C. G. (1999). Robust source coding of images for very noisy channels. *IEEE Transactions on Signal Processing*, 47(4):1198–1202.

Masmoudi, K., Antonini, M., Kornprobst, P. (2010). Spike based neural codes: towards a novel bio-inspired still image coding scheme. Research Report RR-2010-01-FR, INRIA, France.

Medioni, G., Nevatia, R. (1985). Segment-based stereo matching. *Computer Vsion, Graphics and Image Processing*, 31:2–18.

Miled, W. (2008). *Mise en correspondance stéréoscopiques par approches variationnelles convexes; application à la détection d'obstacles routiers.* Phd, Université de Paris-Est, Marne-la-Vallée, France.

Miled, W., Pesquet, J.-C., Parent, M. (2006a). Dense disparity estimation from stereo images. *In International Symposium on Image/Video Communications*, Hammamet, Tunisia.

Miled, W., Pesquet, J. C., Parent, M. (2006b). Disparity map estimation using a total variation bound. *In the 3rd Canadian Conference on Computer and Robot Vision*, pages 48–55.

Miled, W., Pesquet, J.-C., Parent, M. (2009a). A convex optimization approach for depth estimation under illumination variation. *IEEE Transactions on Image Processing*, 18(4):813–830.

Miled, W., Pesquet-Popescu, B., Cherif., W. (2009b). A variational framework for simultaneous motion and disparity estimation in a sequence of stereo images. *In IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 741–744, Taipei, Taiwan.

Moellenhoff, M. S., Maier, M. W. (1998a). Characteristics of disparity-compensated stereo image pair residuals. *Signal Processing: Image Communications*, 14:49–55.

Moellenhoff, M. S., Maier, M. W. (1998b). Transform coding of stereo image residuals. *IEEE Transactions on Image Processing*, 7(6):804–812.

Moreau, J.-J. (1965). Proximité et dualité dans un espace hilbertien. *Bulletin de la Societé Mathématique de France*, 93:273–288.

Morvan, Y., Farin, D., de With, P. H. N. (2007). Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images. *In International Conference on Image Processing*, vol. 1, pages 205–208, San Antonio, TX, USA.

Mun, S., Fowler, J. E. (2009). Block compressed sensing of images using directional transforms. *In International Conference on Image Processing*, pages 3021–3024, Cairo, Egypt.

Otha, Y., Kanade, T. (1985). Stereo by intra and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2): 139–154.

Palfner, T., Mali, A., Muller, E. (2002). Optimized progressive coding of stereo image using discrete wavelet transform. *In World Multiconference on Systemics, Cybernetics and Informatics*, vol. 1, pages 35–39, Orlando, Florida, USA.

Papadimitriou, D. V., Dennis, T. J. (1996). Epipolar line estimation and rectification for stereo image pairs. *IEEE Transactions on Image Processing*, 5(4):672–677.

Parrilli, S., Cagnazzo, M., Pesquet-Popescu, B. (2008). Distortion evaluation in transform domain for adaptive lifting schemes. *In International Workshop on Multimedia Signal Processing*, Cairns, Queensland, Australia.

Pau, G., Pesquet-Popescu, B. (2005). Comparison of spatial m-band filter bank for t+2D video coding. *In Visual communications and image processing*, vol. 5960, Beijing, Chine.

Pau, G., Tillier, C., Pesquet-Popescu, B. (2004). Motion compensation and scalability in lifting-based video coding. *Special Issue on Subband/Wavelet Interframe Video Coding*, 19(7):577–600.

Payan, F., Antonini, M. (2005). An efficient bit allocation for compression normal meshes whith an error driven quantization. *Computer Aided Geometric Design, Special Issue On Geometric Mesh Processing*, 22(5):466–486.

Payan, F., Boulfani, Y., Antonini, M. (2005). Temporal lifting scheme for the compression of animated sequences of meshes. *In IEEE International Workshop VLBV*, Sardinia, Italy.

Pedersini, F., Sarti, A., Tubaro, S. (1999). Multicamera systems. *IEEE Signal Processing Magazine, Special Issue on Stereo and 3D Imaging*, 16(3):55–65.

Pennebaker, W. B., Mitchell, J. L. (1993). *JPEG Still Image Data Compression Standard*. Kluwer Academic Publishers, Norwell, MA, USA.

Pennec, E. L., Mallat, S. (2005). Sparse geometric image representations with bandelets. *IEEE Transactions on Image Processing*, 14(4):423–438.

Perez, M., Pagliari, C., Dennis, T. (2001). Statistical analyses of disparity maps and disparity compensated residuals in the presence of occlusions. *In IEEE International Conference on Image Processing*, vol. 3, pages 911–914, Thessaloniki, Greece.

Perkins, M. G. (1992). Data compression of stereo pairs. *IEEE Transactions on Communications*, 40(4):684–696.

Pesquet-Popescu, B. (1999). *Two-stage adaptive filter bank*. first filling date 1999/07/27, official filling number 99401919.8,European patent number EP1119911.

Pesquet-Popescu, B., Bottreau, V. (2001). Three-dimensional lifting schemes for motion compensated video compression. *In IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pages 1793–1796, Salt Lake City, UT.

Petrisor, T., Pesquet-Popescu, B., Pesquet, J.-C. (2007). A compressed sensing approach to frame-based multiple description coding. *In IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 709–712, Honolulu, HI.

Piella, G., Pesquet-Popescu, B., Heijmans, H. (2002). Adaptive update lifting with a decision rule based on derivative filters. *IEEE Signal Processing Letters*, 9(10):329–332.

Piella, G., Pesquet-Popescu, B., Heijmans, H., Pau, G. (2006). Combining seminorms in adaptive lifting schemes and applications to image analysis and compression. *Journal of Mathematical Imaging and Vision*, 25(2):203–226.

Poli, D., Zhang, L., Gruen, A. (2004). Spot-5 HRS stereo images orientation and automated dsm generation. *In International Archives of Photogrammetry and Remote Sensing*, vol. 35, pages 421–432, Istanbul, Turkey.

Q. Jiang, J. J. L., Hayes, M. H. (1999). A wavelet based stereo image coding algorithm. *In IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 6, pages 3157–3160, Phoenix, Arizona, USA.

Quellec, G., Lamard, M., Gazuguel, G., Cochener, B., Roux, C. (2010). Adaptive nonseparable wavelet transform via lifting and its application to content-based image retrieval. *IEEE Transactions on Image Processing*, 19(1):25–35.

Rao, K. R., Yip, P. (1990). *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, London.

Reeves, T. H., Kingsbury, N. G. (2002). Overcomplete image coding using iterative projection-based noise shaping. *In International Conference on Image Processing*, vol. 3, pages 597–600, Rochester, NY.

Rioul, O. (1993). Regular wavelets: A discrete time approach. *IEEE Transactions on Signal Processing*, 41(12):3572–3578.

Rockafellar, R. T. (1970). *Convex analysis*, vol. 28. Princeton University Press, N.J.

Rolon, J. C., Mendonca, E., Salembier, P. (2009). Generalized lifting with adaptive local PDF estimation for image coding. *In Picture Coding Symposium*, pages 173–176, Chicago, IL, USA.

Rolon, J. C., Salembier, P. (2007). Generalized lifting for sparse image representation and coding. *In Picture Coding Symposium*, page 4, Lisbon, Portugal.

Rolon, J. C., Salembier, P., Alameda, X. (2008). Image compression with generalized lifting and partial knowledge of the signal pdf. *In International Conference on Image Processing*, pages 129–132, San Diego, CA.

Rouse, D., Hemami, S. S. (2008). Understanding and simplifying the structural similarity metric. *In IEEE International Conference on Image Processing*, pages 1188–1191, San Diego, CA, USA.

Rudin, L. I., Osher, S., Stanley, F., Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268.

Said, A., Pearlman, W. (1996). A new fast and efficient image codec based on set partitionning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6:243–250.

Scharstein, D., Szeliski, R. (2002). A taxonomy and evaluation of dense two-frames stereo correspondance algorithms. *International Journal of Computer Vision*, 47(1):7–42.

Sezer, O. G., Harmanci, O., Guleryuz, O. G. (2008). Sparse orthonormal transforms for image compression. *In International Conference on Image Processing*, pages 149–152, San Diego, CA.

Shapiro, J. M. (1993). Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Signal Processing*, 41(12):3445–3462.

Sharifi, K., Leron-Garcia, A. (1995). Estimation of shape parameter for generalized gaussian distributions in subband decompositions of video. *IEEE Transactions on Circuits and Systems for Video Technology*, 5(1):52–56.

Slesareva, N., Bruhn, A., Weickert, J. (2005). Optic flow goes stereo: A variational method for estimating discontinuity-preserving dense disparity maps. *In the 27th DAGM Symposium*, vol. 3663, pages 33–40, Vienna, Austria.

Solé, J., Salembier, P. (2004). Adaptive discrete generalized lifting for lossless compression. *In International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pages 57–60, Montreal, Quebec, Canada.

Solé, J., Salembier, P. (2007). Generalized lifting prediction optimization applied to lossless image compression. *IEEE Signal Processing Letters*, 14(10):695–698.

Sun, C. (2002). Fast stereo matching using rectangular subregioning and 3D maximum surface techniques. *International Journal of Computer Vision*, 47(1):99–117.

Sun, Y.-K. (2004). A two-dimensional lifting scheme of integer wavelet transform for lossless image compression. *In International Conference on Image Processing*, vol. 1, pages 497–500, Singapore.

Sweldens, W. (1995). The lifting scheme: a new philosophy in biorthogonal wavelet construction. *In Wavelet Applications in Signal and Image Processing III, SPIE*, pages 68–79, San-Diego, CA, USA.

Sweldens, W. (1996). The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, 3(2):186–200.

Taubman, D. (1999). Adaptive, non-separable lifting transforms for image compression. *In IEEE International Conference on Image Processing*, vol. 3, pages 772–776, Kobe, Japan.

Taubman, D. (2000). High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7):1158–1170.

Taubman, D., Marcellin, M. (2001). *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, Norwell, MA, USA.

Tsutsui, K., Rokugawa, S., Nakagawa, H., Miyazaki, S., Cheng, C.-T., Shiraishi, T., Yang, S.-D. (2007). Detection and volume estimation of large-scale landslides based on elevation-change analysis using DEMs extracted from high-resolution satellite

stereo imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 45(6):1681–1696.

Tzovaras, D., Grammalidis, N., Strintzis, M. (1998). Disparity field and depth map coding for multiview 3D image generation. *Signal Processing: Image Communication*, 11(3):205–230.

Tzovaras, D., Strintzis, M. G. (1998). Motion and disparity field estimation using rate-distortion optimization. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(2):171–180.

U. Fecker, M. Barkowsky, A. K. (2007). Time-constant histogram matching for colour compensation of multi-view video sequences. *In Picture Coding Symposium*, Lisbon, Portugal.

Usevitch, B. (1996). Optimal bit allocation for biorthogonal wavelet coding. *In Data Compression Conference*, pages 387–395, Snowbird, USA.

Vaidyanathan, P. P. (1993). *Multirate Systems and Filter Banks*. Prentice-Hall, Englewood Cliffs, NJ.

Veksler, O. (2001). Stereo matching by compact windows via minimum ratio cycle. *In IEEE Conference Proceedings of International Conference on Computer Vision*, vol. 1, pages 540–547, Vancouver, BC, Canada.

Veksler, O. (2005). Stereo correspondence by dynamic programming on a tree. *In IEEE Conference Proceedings of Computer Vision and Pattern Recognition*, vol. 2, pages 384–390, San Diego, United States.

Waldowski, M. (1991). A new segmentation algorithm for videophone applications based on stereo image pairs. *IEEE Transactions on Communications*, 39(12):1856–1868.

Wallace, G. K. (1991). The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44.

Wang, S., Chen, H. (1999). An improved algorithm of motion compensation MPEG video compression. *In International Vehicle Electronics Conference*, vol. 1, pages 261–264, Changchun, China.

Wang, Z., Bovik, A. C., Sheikh, H. R., Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Woo, O., Ortega, A. (1997). Stereo image compression based on disparity field segmentation. *In SPIE Conference on Visual Communications and Image Processing*, vol. 3024, pages 391–402, San Jose, California.

Woo, O., Ortega, A. (2000). Overlapped block disparity compensation with adaptive windows for stereo image coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(2):194–200.

Xing, M., Hexin, C., Yan, Z. (2009). Stereo image coding method using stereo matching with difference based adaptive searching windows. *In IEEE International Workshop on Imaging Systems and Techniques*, pages 373–376, Shenzhen.

Xiong, R., Wu, F., Xu, J., Li, S., Zhang, Y.-Q. (2004). Barbell lifting wavelet transform for highly scalable video coding. *In Picture Coding Symposium*, San Francisco, CA, USA.

Yang, J., Zhang, Y., Yin, W. (2009). An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise. *SIAM Journal on Sciences Computing*, 31(4):2842–2865.

Yang, W., Ngi, N. K. (2004). MPEG-4 based stereoscopic video sequences encoder. *In International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, Montreal, Quebec, Canada.

Zhang, Z. (1998). Determining the epipolar geometry and its uncertainty: a review. *International Journal on Computer Vision*, 27(2):161–195.

Zhang, Z., Xu, G. (1997). A general expression of the fundamental matrix for both perspective and affine cameras. *In International Joint Conference on Artificial Intelligence*, vol. 2, pages 1502–1507, Nagoya, Japan.

Zitova, B., Flusser, J. (2003). Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000.