



**HAL**  
open science

# La RWHT+P pour un codage multirésolution sans perte avancé

Olivier Déforges, Marie Babel, Jean Motsch

► **To cite this version:**

Olivier Déforges, Marie Babel, Jean Motsch. La RWHT+P pour un codage multirésolution sans perte avancé. Nov 2006, pp.4. hal-00132957

**HAL Id: hal-00132957**

**<https://hal.science/hal-00132957>**

Submitted on 23 Feb 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# La RWHT+P pour un codage multirésolution sans perte avancé

Olivier Déforges<sup>1</sup>

Marie Babel<sup>1</sup>

Jean Motsch<sup>2</sup>

IETR UMR CNRS 6164 - Groupe Image et Télédétection

<sup>1</sup> INSA de Rennes

<sup>2</sup> LESTP/CREC Saint-Cyr, Écoles de Coëtquidan

{odeforge,mbabel}@insa-rennes.fr, jean.motsch@st-cyr.terre.defense.gouv.fr

## Résumé

*Cet article présente une méthode complète de codage sans perte multirésolution, à forte scalabilité sémantique. En particulier, une forme réversible de la classique Transformée de Walsh Hadamard (RWHT) est tout d'abord introduite comme alternative aux transformées sans perte standard. Une représentation pyramidale et des schémas de décomposition reposant sur cette transformée sont ensuite proposés pour un codage multirésolution réversible. Des améliorations significatives y sont apportées en ajoutant deux concepts supplémentaires : la "résolution localement adaptative" à travers une représentation quadtree, et une étape de prédiction. Les résultats expérimentaux montrent en final que la méthode RWHT+P proposée aboutit à d'excellentes performances comparées à celles de l'état de l'art.*

## Mots clefs

Scalabilité, codage réversible, représentation pyramidale.

## 1 Introduction

Les nouvelles générations de codecs d'images doivent bien sûr être efficaces en termes de performances de compression, mais aussi fournir des fonctionnalités avancées telles que la scalabilité, le contrôle du débit, l'encodage de régions d'intérêt, ou encore la description de scènes. De plus, un schéma de codage unique, capable de compresser depuis les bas débits jusqu'au sans perte, constituerait une solution souhaitable pour des usages variés. Dans [1], nous avons introduit la méthode LAR (Locally Adaptive Resolution) comme codage irréversible efficace à bas et moyens débits. Cette technique repose sur un schéma global d'encodage à deux couches : une couche spatiale et une couche spectrale. Le premier codec spatial fournit une image principale à bas débit, alors que le codec spectral encode la texture locale. La qualité des images compressées par le LAR a été évaluée par un autre laboratoire, et reconnue meilleure que celle des images obtenues par Jpeg-2000 [1]. La méthode LAR repose en partie sur une décomposition en taille de blocs variable de type quadtree, estimée à partir de l'activité locale. Cette structure parti-

culière a permis une extension du schéma simple à une représentation en régions construite à partir des images fortement compressées, et autorisant ensuite un codage par région d'intérêt [2]. Récemment, nous avons également proposé une version modifiée du codec permettant du codage sans perte tout en augmentant son caractère scalable [3] : les couches spatiales et spectrales initiales y ont été substituées par deux décompositions multirésolution de type quadtree, à partir d'une solution modifiée de la transformée en S. Ce papier présente une alternative à cette méthode quant au noyau de décomposition. Cette nouvelle génération de codec LAR, appelée "RWHT+P", surpasse les performances précédentes à la fois pour du codage à bas débit et du sans perte. Nous nous limiterons toutefois ici au dernier cas. Plus précisément, la première partie de l'article s'attachera à introduire une technique permettant de rendre la transformée de Walsh-Hadamard (WHT), avec un noyau  $2 \times 2$ , réversible. Le reste du papier proposera un schéma de codage sans perte complet avec une scalabilité avancée en résolution.

En codage sans perte, la compression et décompression de données source doivent résulter dans l'exacte récupération de tout le signal d'origine. Le codage des images sans perte est nécessaire dans les applications où la dégradation des données n'est pas tolérée. C'est le cas par exemple pour des applications dans le domaine médical, en télédétection, ou encore pour l'archivage d'images et de vidéos à qualité studio. L'état de l'art en codage sans perte peut être grossièrement divisé en deux approches : les méthodes prédictives dans le domaine spatial, avec des codeurs populaires tels que CALIC [4], et les méthodes basées transformation généralement reposant sur la théorie des ondelettes. Le principal intérêt des codecs fondés ondelettes reside dans le fait qu'ils proposent un codage scalable, avec la possibilité d'une représentation multirésolution de l'image. Les transformées sans perte ont la particularité d'une correspondance non équivoque d'entiers à entiers, au contraire de l'essentiel des transformées dites avec pertes. La plupart des méthodes utilisent pour cela le concept "d'arrondi" [5]. La WHT est une technique très connue pour la compression des images et du signal. Beaucoup de méthodes multirésolution, utilisant cette transformation sur des blocs

$2 \times 2$ , ont été proposées dans la littérature. A des fins de compression réversible, une version modifiée de la WHT 1D a été introduite par P. Lux [6], puis popularisée par Said [5] et connue sous le nom de transformée en ‘S’, ou ‘transformée en ondelettes entière de Haar’ [7]. La transformée en S est actuellement reconnue comme une des meilleures bases d’ondelettes entières parmi celles existantes pour le codage réversible [8]. Afin d’améliorer la compression, une étape de prédiction a été ajoutée à la transformée elle-même, conduisant à la technique très connue de méthode ‘S+P’, qui fut plus tard généralisée à travers le concept de ‘lifting scheme’ [9].

Le paragraphe 2 introduit l’adaptation réalisée sur la transformée non réversible  $WHT_{2 \times 2}$  pour une forme sans perte, appelée la  $RWHT$ . Le paragraphe 3 présente un schéma pyramidal de compression sans perte fondé sur la  $RWHT$ . Ce schéma est ensuite amélioré à travers l’ajout de deux fonctionnalités supplémentaires : une décomposition quadtree et une phase de prédiction/interpolation. Finalement, nous concluons dans le paragraphe 4.

## 2 La transformée RWHT

De manière à retrouver les données en entrée à partir du vecteur transformé, la transformée  $WHT_{2 \times 2}$  initiale a été transformée dans sa normalisation pour aboutir à la transformée en S.

$$WHT_{2 \times 2} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad S = \begin{bmatrix} \frac{1}{2} & 1 \\ \frac{1}{2} & -1 \end{bmatrix}. \quad (1)$$

La correspondance non équivoque d’entiers à entiers est ensuite possible grâce aux opérations duales d’arrondi pendant les transformations directes et inverses. Une transformation 2D peut être réalisée en appliquant successivement la transformée suivant les deux directions. Cependant, ce noyau est moins efficace à des fins de codage avec pertes, dès lors qu’il augmente la dynamique des coefficients de hautes fréquences.

Nous introduisons ainsi une technique de transformation 2D réversible utilisant directement la matrice de transformation formelle de la  $WHT_{2 \times 2}$ .

Soit  $U_{2 \times 2}$  le bloc d’entrée avec :

$$U_{2 \times 2} = \begin{bmatrix} u_0 & u_1 \\ u_2 & u_3 \end{bmatrix}. \quad (2)$$

Le bloc transformé  $Z_{2 \times 2}$  est alors défini par :

$$\begin{aligned} Z_{2 \times 2} &= WHT_{2 \times 2}(U_{2 \times 2}) \\ &= W_{2 \times 2} U_{2 \times 2} W_{2 \times 2} \\ &= \begin{bmatrix} z_0 & z_1 \\ z_2 & z_3 \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} u_0 + u_1 + u_2 + u_3 & u_0 + u_1 - u_2 - u_3 \\ u_0 - u_1 + u_2 - u_3 & u_0 - u_1 - u_2 + u_3 \end{bmatrix}. \end{aligned} \quad (3)$$

Soit  $\hat{Z}_{2 \times 2}$  le bloc arrondi de  $Z_{2 \times 2}$  tel que :

$$\begin{aligned} \hat{Z}_{2 \times 2} &= Round(Z_{2 \times 2}) = \begin{bmatrix} \hat{z}_0 & \hat{z}_1 \\ \hat{z}_2 & \hat{z}_3 \end{bmatrix} \\ &= \begin{bmatrix} Round_{z_0}[z_0] & Round_{z_1}[z_1] \\ Round_{z_2}[z_2] & Round_{z_3}[z_3] \end{bmatrix}. \end{aligned} \quad (4)$$

$Round_{z_i}[\cdot]$  correspond à l’opération d’arrondi appliquée sur  $z_i$  : arrondi inférieur ( $\lfloor \cdot \rfloor$ ) ou supérieur ( $\lceil \cdot \rceil$ ).

La transformée inverse est identique à la transformée directe. Notons  $\tilde{U}_{2 \times 2}$  le bloc inverse transformé de  $\hat{Z}_{2 \times 2}$ , et  $\hat{U}_{2 \times 2}$  le bloc arrondi de  $\tilde{U}_{2 \times 2}$ . Définir une transformée réversible implique que  $\hat{U}_{2 \times 2} = U_{2 \times 2}$  malgré les opérations d’arrondi. Pour y parvenir directement dans l’espace 2D, nous proposons une méthode de contrôle des valeurs arrondies fondée sur la fonction de parité  $P(\cdot)$  telle que :

$$P(x) = \begin{cases} o & \text{si } x \text{ impair} \\ e & \text{si } x \text{ pair} \end{cases}, \quad x \in \mathbb{N}. \quad (5)$$

En posant  $z_0 = \lfloor z_0 \rfloor + \frac{\epsilon}{2}$ ,  $\epsilon \in \{0, 1\}$ , et lorsqu’il est substitué dans l’équation (3),  $Z_{2 \times 2}$  peut être exprimé selon :

$$Z_{2 \times 2} = \frac{1}{2} \begin{bmatrix} 2 \lfloor z_0 \rfloor + \epsilon & 2(\lfloor z_0 \rfloor - u_2 - u_3) + \epsilon \\ 2(\lfloor z_0 \rfloor - u_1 - u_3) + \epsilon & 2(\lfloor z_0 \rfloor - u_1 - u_2) + \epsilon \end{bmatrix} \quad (6)$$

A cette étape, deux cas demeurent possibles.

**Somme paire :** Si  $P(\sum_{i=0}^3 u_i) = e$ , alors  $\epsilon = 0$  et  $\hat{Z}_{2 \times 2} = Z_{2 \times 2}$ . Ceci implique des valeurs entières reconstruites :

$$\tilde{u}_0 = \frac{1}{2}(4 \lfloor z_0 \rfloor - 2(u_1 + u_2 + u_3)) = \frac{1}{2}(2u_0) = u_0 \quad (7)$$

et  $\hat{u}_0 = u_0$ .

**Somme impaire :** Si  $P(\sum_{i=0}^3 u_i) = o$ , le problème de l’arrondi de  $Z_{2 \times 2}$  est déplacé à celui d’arrondir  $\epsilon/2$  pour chaque coefficient. Soit  $\Delta_i \in \{0, 1\}$  l’arrondi de  $\epsilon/2$  pour  $z_i$  ( $\Delta_i = Round_{z_i}[\frac{\epsilon}{2}] = \frac{\epsilon}{2} + \frac{\epsilon_i}{2}$ ,  $\epsilon_i \in \{-1, +1\}$ ).

$$\begin{aligned} \hat{Z}_{2 \times 2} &= \begin{bmatrix} \lfloor z_0 \rfloor + \Delta_0 & \lfloor z_0 \rfloor - u_2 - u_3 + \Delta_1 \\ \lfloor z_0 \rfloor - u_1 - u_3 + \Delta_2 & \lfloor z_0 \rfloor - u_1 - u_2 + \Delta_3 \end{bmatrix} \\ &= Z_{2 \times 2} + \frac{1}{2} \begin{bmatrix} \epsilon_0 & \epsilon_1 \\ \epsilon_2 & \epsilon_3 \end{bmatrix}. \end{aligned} \quad (8)$$

Ainsi les coefficients reconstruits s’écrivent :

$$\tilde{U}_{2 \times 2} = \frac{1}{2} \begin{bmatrix} 2(u_0 - \epsilon) + (\Delta_0 + \Delta_1 + \Delta_2 + \Delta_3) \\ 2u_2 + (\Delta_0 - \Delta_1 + \Delta_2 - \Delta_3) \\ 2u_1 + (\Delta_0 + \Delta_1 - \Delta_2 - \Delta_3) \\ 2u_3 + (\Delta_0 - \Delta_1 - \Delta_2 + \Delta_3) \end{bmatrix}. \quad (9)$$

Dès lors, la reconstruction exacte implique :

$$\begin{cases} \Delta_0 + \Delta_1 + \Delta_2 + \Delta_3 = 2\epsilon = 2 \\ \Delta_0 + \Delta_2 = \Delta_1 + \Delta_3 \\ \Delta_0 + \Delta_1 = \Delta_2 + \Delta_3 \\ \Delta_0 + \Delta_3 = \Delta_1 + \Delta_2 \end{cases} \quad (10)$$

Clairement, le système d'équations sur les valeurs de  $\Delta_i$  ne peut être résolu. De ce fait, aucun arrondi systématique, comme pour la transformée en S, ne permet une transformation réversible.

L'alternative réside dans un contrôle des opérations d'arrondi, de sorte que le procédé de décodage soit à même de distinguer des valeurs reconstruites entières ou non entières. Fixer  $\{\Delta_i\}$  tel que  $P(\sum_{i=0}^3 \Delta_i) = o$  aboutit uniquement à des valeurs réelles pour les coefficients  $\tilde{u}_i$ . Si nous imposons que  $\sum_{i=0}^3 \Delta_i = 1$ , alors

$$\begin{aligned} \Delta_0 + \Delta_1 + \Delta_2 + \Delta_3 = 1 &\Rightarrow 4\frac{\epsilon}{2} + \frac{\epsilon_1}{2} + \frac{\epsilon_2}{2} + \frac{\epsilon_3}{2} + \frac{\epsilon_3}{2} = 1 \\ &\Rightarrow \epsilon_0 + \epsilon_1 + \epsilon_2 + \epsilon_3 = -2. \end{aligned} \quad (11)$$

Par exemple, l'ensemble  $\{\epsilon_0 = 1, \epsilon_1 = \epsilon_2 = \epsilon_3 = -1\}$  est une solution pour la condition dans (11). Avec un tel choix, la transformation inverse est finalement réalisée en deux phases :

1. calcul de  $\tilde{U}_{2 \times 2} = WHT(\hat{Z}_{2 \times 2})$ .
2. si  $\tilde{u}_i$  réel, alors calcul d'un nouveau  $\tilde{U}_{2 \times 2}$  tel que :

$$\tilde{U}_{2 \times 2} = WHT \left( \hat{Z}_{2 \times 2} - \frac{1}{2} \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix} \right). \quad (12)$$

On vérifie aisément que  $\tilde{U}_{2 \times 2} = U_{2 \times 2}$  dans tous les cas.

### 3 Codage sans perte par la pyramide RWHT+P

*Notations :*

$I(i, j)$  dénote le pixel dans une image  $I$  avec les coordonnées  $(i, j)$ ,  $I(\mathbf{b}^N(i, j))$  le bloc  $\mathbf{b}^N(i, j)$  dans  $I$  incluant l'ensemble des pixels  $\{I(N.i, N.j), \dots, I(N.i + N - 1, N.j + N - 1)\}$ .

#### 3.1 La pyramide RWHT

Nous introduisons la pyramide  $\{Y_l\}_{l=0}^{L_{max}}$  comme la représentation multirésolution d'une image  $I$  de taille  $N_x \times N_y$ , où  $L_{max}$  est le niveau haut de la pyramide et  $l = 0$  le niveau pleine résolution. Comme pour le cas de la  $WHT_{2 \times 2}$  classique, la pyramide est construite itérativement en regroupant quatre blocs pour former un bloc moyen au niveau supérieur :

$$\begin{cases} l = 0, & Y_0(i, j) = I(i, j); \\ l > 0, & Y_l(i, j) = \left[ \frac{1}{4} \sum_{k=0}^1 \sum_{m=0}^1 Y_{l-1}(2x+k, 2y+m) \right]. \end{cases} \quad (13)$$

avec  $0 \leq i \leq N_x^l$ ,  $0 \leq j \leq N_y^l$ , où  $N_x^l = N_x/2^l$  et  $N_y^l = N_y/2^l$ .

La décomposition *top-down* de la pyramide consiste à encoder le bloc transformé  $Z_l(\mathbf{b}^2(i, j))$  par la *RWHT* de chaque bloc d'entrée  $Y_l(\mathbf{b}^2(i, j))$ . De (3) et (13), nous obtenons :

$$\begin{aligned} Y_{l+1}(i, j) &= \left\lfloor \frac{z_{0l}(2i, 2j)}{2} \right\rfloor \\ &\Rightarrow z_{0l}(2i, 2j) = 2 \times Y_{l+1}(i, j) + \epsilon_{z_{0l}(2i, 2j)}, \end{aligned} \quad (14)$$

avec  $\epsilon_{z_{0l}(2i, 2j)} \in \{0, 1\}$ .

Ainsi, la composante *DC* de chaque bloc est reconstruite sans ambiguïté depuis le niveau supérieur plus un bit additionnel. Ce bit est ici codé séparément des autres coefficients. Si nous notons  $\hat{Z}_l(\mathbf{b}^2(i, j))$  le bloc *WHT* transformé de  $Y_l(\mathbf{b}^2(i, j))$  avec ce seul bit comme composante *DC* ( $\hat{z}_{0l} = \epsilon_{z_{0l}}$ ), alors la reconstruction à partir du niveau supérieur du bloc *WHT* courant est donnée par :

$$\begin{aligned} \tilde{Y}_l(\mathbf{b}^2(i, j)) &= EXP(Y_{l+1}(i, j)) + \tilde{Y}_l(\mathbf{b}^2(i, j)) \\ \text{avec } \tilde{Y}_l(\mathbf{b}^2(i, j)) &= WHT_{2 \times 2}^{-1} \left( \hat{Z}_l(\mathbf{b}^2(i, j)) \right). \end{aligned} \quad (15)$$

La fonction *EXP* duplique simplement une valeur d'un nœud de l'arbre à ses quatre fils.

A cette étape, nous possédons une représentation pyramidale et un encodage classiques fondés sur une transformation par  $WHT_{2 \times 2}$ , mais avec l'exception d'une possible décomposition sans perte. Le tableau 1 donne les valeurs entropiques d'ordre zéro pour une compression réversible avec à la fois la transformée de S et la *RWHT* proposée. Le niveau supérieur a été codé pour les deux cas par un simple *MICD*. Les résultats démontrent que la méthode proposée améliore la compression tout en généralisant le noyau *2D* non réversible de la *WHT* à une version réversible.

#### 3.2 Décomposition quadtree

Nos précédents travaux ont été consacrés à l'élaboration d'un schéma de codage fondé sur une représentation à taille de blocs variable, efficace en termes de compression à la fois pour les hauts et les très bas débits [1]. L'idée est ici de montrer que ce concept, appliqué dans un contexte de codage sans perte, apporte des améliorations significatives au schéma pyramidal original.

Une partition quadtree suppose la décomposition de l'image entière en blocs de taille  $N \times N$ , avec  $N = 2^k$ , et  $k \in \mathbb{N}^+$ . La représentation pyramidale précédente induit une décomposition dyadique, ordinairement associée à une partition multiniveaux quadtree  $QP^{[2^{L_{max}} \dots 2^l]}$ , où le niveau  $l$  de la pyramide spécifie également la résolution la plus fine. Plus généralement, nous considérons une partition quadtree globale de l'image. Ainsi,  $QP^{[N_{max} \dots N_{min}]}$  définit les tailles de blocs autorisées, et le paramètre  $N_l \in [N_{max} \dots N_{min}]$  donne la limite supérieure des tailles de blocs à décomposer au niveau  $l$  de la pyramide. A titre

Image	Entropie (bpp)						
	Raw	S	RWHT	CALIC	S+P	RWHT+P	RWHT+P Qd
Barbara2	7.51	5.45	5.47	4.93	5.04	5.06	<b>4.89</b>
Hotel	7.57	5.11	5.09	<b>4.57</b>	4.97	4.83	4.60
Lena	7.44	4.77	4.75	4.33	4.33	4.30	<b>4.19</b>
Gold	7.60	5.08	5.06	4.65	4.73	4.73	<b>4.63</b>
Peppers	7.57	4.89	4.87	4.58	4.67	4.54	<b>4.43</b>
us	4.84	3.65	3.64	3.60	3.78	3.78	<b>3.26</b>
tools	7.52	5.95	5.95	5.53	5.73	5.71	<b>5.50</b>
Average	<b>7.15</b>	4.99	4.97	4.60	4.75	4.71	<b>4.49</b>

TAB. 1 – Comparaison des approches proposées avec l'état de l'art. Entropie du premier ordre (bit/pixels).

d'exemple, une partition globale  $QP^{[32...2]}$  conduit à l'encodage de la seule représentation utilisant les blocs de taille 32 à 2, tandis que la notation  $N_0 = 4$  implique la décomposition au niveau 0 des blocs de taille 4 et 2.

Enfin, le paramètre  $L_{min}$  indique le dernier niveau à encoder. Ainsi, pour tous les niveaux inférieurs à  $L_{min}$ , la valeur de l'ensemble des nœuds de la pyramide résulte simplement d'une phase de duplication.

La partition de l'image est réalisée en fonction de l'activité locale, estimée par un gradient morphologique (différence entre les valeurs minimales et maximales) calculé sur chaque bloc. Ainsi, une première phase de décomposition de la pyramide s'attache à raffiner uniquement les petits blocs situés sur les contours, selon l'expression :

$$\tilde{Y}_l(\mathbf{b}^2(i, j)) = \begin{cases} EXP(Y_{l+1}(i, j)) + \tilde{Y}_l(\mathbf{b}^2(i, j)), & \text{si } \mathbf{b}^2(i, j) \notin QP^{[N_{max}...N_l]} \\ \text{et } l \geq L_{min} \\ EXP(Y_{l+1}(i, j)) \text{ sinon} \end{cases} \quad (16)$$

avec  $l < L_{max}$ .  $\tilde{Y}_l(\mathbf{b}^2(i, j))$  représente le bloc reconstruit du bloc original  $Y_l(\mathbf{b}^2(i, j))$ .

La figure 1 illustre les étapes de codage relatives à ce modèle (codeur  $C_l$ ).

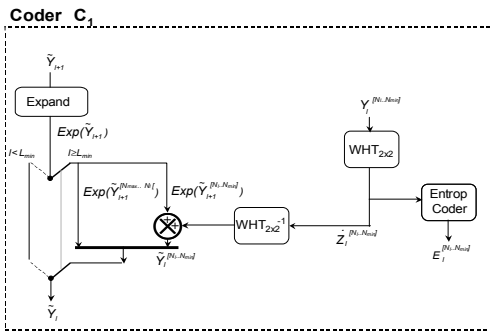


FIG. 1 – Simple pyramidal coder

La deuxième descente de la pyramide consiste en la décomposition de tous les blocs du niveau courant qui n'ont pas été traités lors de la première phase : l'information de texture locale est ainsi encodée.

La décomposition quadtree utilisée ici possède plusieurs avantages :

1. le nombre de niveau de décomposition est doublé ( $2 \times L_{max}$ ), augmentant la scalabilité du schéma,
2. des images de bonne qualité sont disponibles à bas débit,
3. l'approche agit comme une **modélisation de contexte** objective, décorrélant naturellement les lois entropiques des erreurs de prédiction : entropie haute pour la première descente, entropie basse pour la seconde.

### 3.3 Pyramide RWHT et prédiction

Si la prédiction et l'interpolation constituent deux fonctions relativement proches dans le domaine spatial, elles poursuivent cependant deux objectifs différents. La première vise à optimiser la compression en limitant l'erreur de prédiction, alors que la seconde augmente la qualité et la résolution de l'image. Un bon prédicteur n'est pas nécessairement un bon interpolateur, et vice-versa. Les deux fonctions s'avèrent cependant utiles à notre schéma de codage. En particulier, à un niveau décomposition donné, la première descente requiert à la fois une phase de prédiction, pour l'encodage des blocs décomposés, et une phase d'interpolation pour le lissage des zones homogènes (blocs non décomposés). De ce fait, nous proposons une méthode unifiée pour les deux fonctions via la définition d'un procédé d'estimation unique.

Dans ce qui suit, nous notons  $\tilde{Y}_l(\mathbf{b}^2(i, j))$  le bloc reconstruit du bloc original  $Y_l(\mathbf{b}^2(i, j))$ . L'estimation consiste en la reconstruction linéaire des valeurs inconnues à partir de leur valeur moyenne de bloc. L'information inter et intra niveau est alors exploitée dans un contexte 2D selon :

Initialisation :

$$\tilde{Y}_l(\mathbf{b}^2(i, j)) = \tilde{Y}_{l+1}(i, j), \forall (i, j) \in \tilde{Y}_{l+1}$$

Estimation :

$$\begin{aligned} \tilde{Y}_l(2i+k, 2j+m) = & \tilde{Y}_{l+1}(i, j) \\ & + \beta_m \left( \tilde{Y}_l(2i+k, 2j-1+3m) - \tilde{Y}_{l+1}(i, j) \right) \\ & + \beta_k \left( \tilde{Y}_l(2i-1+3k, 2j+m) - \tilde{Y}_{l+1}(i, j) \right), \\ & (k, m) \in \{0, 1\}^2, \end{aligned} \quad (17)$$

avec  $\beta_m$  et  $\beta_k$  les poids appliqués au gradient local.

Sans quantification, les valeurs de voisinage diffèrent selon la configuration, et correspondent

- soit à une valeur exactement reconstruite (position déjà traitée au niveau courant par un codage exact),
- soit à une valeur moyenne de bloc (position non traitée au niveau courant),
- soit à une valeur interpolée (position déjà traitée mais non encodée).

Dans ce dernier cas, il existe une inter-dépendance des données, dans la mesure où la valeur du voisinage a été partiellement calculée à partir de la valeur moyenne du bloc courant. Ceci implique, pour deux positions adjacentes de blocs, une relation entre les coefficients  $\beta$ . Si l'on considère, pour deux positions  $(2i, 2j)$  et  $(2i-1, 2j)$ , uniquement les relations horizontales, l'expression 17 devient

$$\begin{cases} \check{Y}_l(2i, 2j) = \\ \check{Y}_{l+1}(i, j) + \beta_0 \left( \check{Y}_l(2i-1, 2j) - \check{Y}_{l+1}(i, j) \right) \\ \check{Y}_l(2i-1, 2j) = \\ \check{Y}_{l+1}(i-1, j) + \beta_1 \left( \check{Y}_l(2i, 2j) - \check{Y}_{l+1}(i-1, j) \right) \\ \Rightarrow \check{Y}_l(2i, 2j) = \\ \check{Y}_{l+1}(i, j) + \beta_0 \left( \check{Y}_{l+1}(i, j) - \check{Y}_{l+1}(i-1, j) \right) (\beta_1 - 1). \end{cases} \quad (18)$$

Si un gradient symétrique est de plus imposé de telle sorte que

$$\check{Y}_l(2i-1, 2j) - \check{Y}_{l+1}(i-1, j) = - \left( \check{Y}_l(2i, 2j) - \check{Y}_{l+1}(i, j) \right), \quad (19)$$

alors nous obtenons la relation suivante :

$$\beta_0 = \frac{\beta_1}{1 - \beta_1}, \quad \beta_1 \in [0, 0.5]. \quad (20)$$

L'effet de l'estimation se calibre via la valeur de  $\beta_1$  :

- pour  $\beta_1 = 0$ ,  $\check{Y}_l(2i, 2j) = \check{Y}_{l+1}(i, j)$  : l'estimation s'avère sans effet (le bloc est reconstruit par sa valeur moyenne),
- pour  $\beta_1 = 0.25$ ,  $\check{Y}_l(2i, 2j) - \check{Y}_{l+1}(i, j) = \check{Y}_{l+1}(i-1, j) - \check{Y}_l(2i-1, 2j)$  : la pente est régulière entre les deux points interpolés (lissage de l'image),
- pour  $\beta_1 = 0.5$ ,  $\check{Y}_l(2i, 2j) = \check{Y}_l(2i-1, 2j)$  : les points reconstruits adjacents sont identiques (accentuation des contours).

En fait, nos expérimentations ont montré que le mode lissage ( $\beta_1 = 0.25$ ) conduit à la meilleure prédiction.

La figure 2 donne le nouveau schéma de codage incluant la phase d'estimation. Le codeur  $C_2$  s'appuie uniquement sur des relations intra-niveaux, et est adapté à une reconstruction progressive de l'image (le rehaussement de la résolution d'obtient directement à partir de l'interpolation de l'image issue du niveau précédent). Le codeur  $C_3$  tire aussi parti des valeurs reconstruites au niveau courant, conduisant naturellement à des performances de compression supérieures.

Les résultats de la compression sans perte par la méthode proposée sont regroupés dans le tableau 1, et comparés à ceux obtenus par les codeurs de l'état de l'art CALIC (non scalable) et S+P (scalable). Le choix de S+P, plutôt que tout

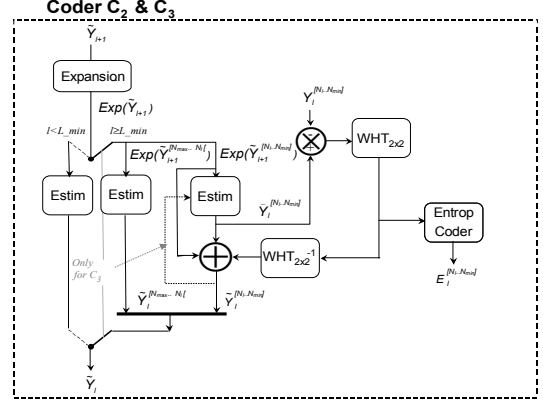


FIG. 2 – Pyramidal coder with prediction step

autre noyau d'ondelettes entières, a été motivé par deux états de fait : d'une part, cette solution demeure l'une des meilleures, et d'autre part, un codeur est disponible et permet de réaliser des expérimentations sans mettre en œuvre la couche de codage entropique.

La configuration "RWHT+P" correspond au mode  $C_3$  du codeur sans partition (décomposition en une seule descente). "RWHT+P & Qd" implique une partition  $QP^{[64...2]}$ , avec  $N_0 = 2$  et  $N_l = 2^l$ . Nous remarquons immédiatement que la séparation des lois entropiques pour les symboles à encoder, suivant la décomposition quadtree, compense largement le coût de sa structure : les résultats de codage de cette configuration dépassent largement ceux de S+P et de CALIC.

Afin d'illustrer la **scalabilité sémantique** de notre approche, la figure 3 montre des images intermédiaires obtenues lors du processus de décomposition pyramidale. Pour six niveaux de décomposition, l'encodage sans perte de l'image nécessite onze flux successifs ( $1 + 2 \times 5$ ). Il est à noter que la distorsion visuelle est essentiellement due à un effet de flou, moins perturbant pour l'observateur que les effets de blocs ou de rebonds. Les images reconstruites à la fin de la première passe sont caractérisées par des contours globalement conservés et des zones homogènes lissées.

## 4 Conclusion

La transformée en S a été initialement conçue afin d'introduire la notion de réversibilité dans la transformée Walsh-Hadamard classique  $WHT_{2 \times 2}$ . La première partie de l'article s'est attachée à démontrer que, moyennant des opérations d'arrondi suivant un critère de parité, le noyau de la  $WHT_{2 \times 2}$  possède à lui seul cette propriété. La décomposition pyramidale RWHT qui en résulte présente des performances meilleures que celles de la pyramide en S.

Deux innovations majeures ont de plus été exposées, contribuant à une décorrélation supplémentaire de l'information, ainsi qu'à des améliorations significatives du schéma original, à savoir : une décomposition pyramidale conditionnelle au contenu de l'image, et une phase de

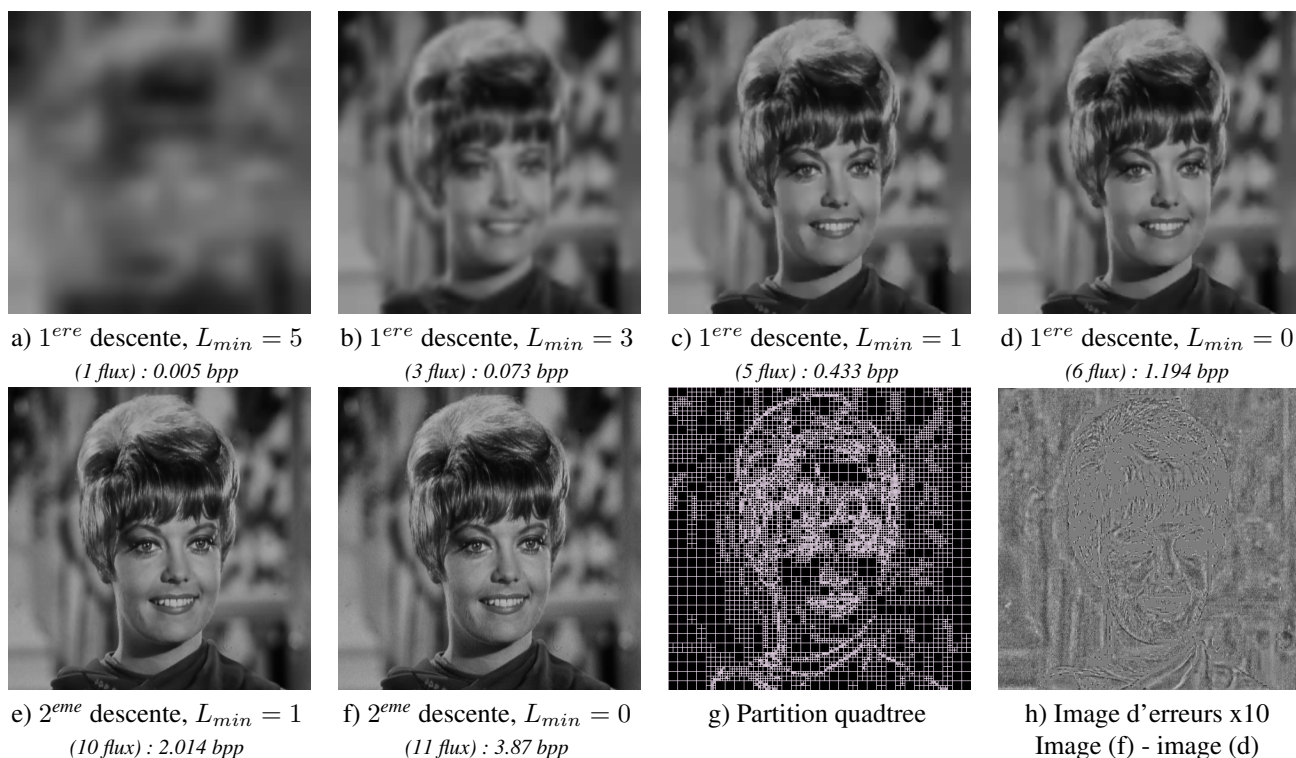


FIG. 3 – Codage scalable sans perte sur “Zelda”, partition quadtree associée  $QP^{[64\dots 2]}$

prédiction. Le schéma scalable global surpasse, en termes de compression sans perte, à la fois S+P et CALIC. En outre, il réalise une représentation multirésolution localement adaptée de l’image, permettant la reconstruction à très bas débit d’images de bonne qualité visuelle.

Cette méthode de codage a aussi prouvé son efficacité dans le contexte de la compression avec pertes à bas débit. Ainsi, il est possible d’introduire une étape de quantification de l’erreur ajustée au contenu : quantification fine sur les gros blocs (l’œil humain s’avère plus sensible aux variations de luminance sur les zones uniformes), quantification grossière sur les petits blocs (l’œil humain est moins sensible aux variations sur les contours).

Une application directe de cette méthode consiste en la définition d’un système d’archivage des images à haute résolution issues du musée du Louvre. La bibliothèque numérique permettra l’accès à différentes qualités d’images. Ces travaux sont supportés par le ministère français de la recherche dans le cadre du projet ANR “TSAR”.

## Références

[1] O. Deforges et J. Ronsin. Region of interest coding for low bit-rate image transmission. Dans *ICME*, volume 1, pages 107–110, July 2000.

[2] O. Deforges et J. Ronsin. Supervised segmentation at low bit rates for region representation and color image

compression. Dans *ICME*, volume 1, pages 665–668, 2002.

[3] M. Babel, O. Deforges, et J. Ronsin. Interleaved s+p pyramidal decomposition with refined prediction model. Dans *ICIP*, volume 2, pages 750–753, 2005.

[4] X. Wu et N. Memon and. A Context-based, Adaptive, Lossless/Nearly-Lossless Coding Scheme for Continuous-Tone Images (CALIC). *International Standards Organization working document, ISO/IEC SC29/WG 1/N256*, 1995.

[5] A. Said et W. Pearlman. Reversible image compression via multiresolution representation and predictive coding. Dans *Visual Communication and Image Processing*, volume 209, pages 664–674. SPIE, Novembre 1993.

[6] P. Lux. A novel set of closed orthogonal functions for picture coding. *Archiv für Elektronik und Übertragungstechnik*, 31(7) :267–274, 1977.

[7] K. Komatsu et K. Sezaki. Lossless 2d discrete walsh-hadamard transform. *Proc. IEEE ICASSP*, May 2001.

[8] G. C. K. Abhayaratne. *Lossless and Nearly Lossless Digital Video Coding*. Thèse de doctorat, Univ. of Bath, 2002.

[9] Wim Sweldens. The Lifting Scheme : A Construction of Second Generation Wavelets. *SIAM Journal on Mathematical Analysis*, 29(2) :511–546, 1998.