

**BADJI MOKHTAR ANNABAUNIVERSITY  
UNIVERSITE BADJIMOKHTAR ANNABA**



**Faculté: Sciences de l'Ingénierat  
Département: Electronique**

**Année : 2013**

**MEMOIRE**

**Présenté en vue de l'obtention du diplôme de MAGISTER**

**Intitulé**

**Compression des Images en Couleurs  
Fixes en Utilisant la DWT**

**Option : Traitement du Signal**

**Par:**

**BENYAHIA Mohamed**

**Directeur de Mémoire: DOGHMANE Noureddine Prof. Université Annaba**

**DEVANT Le JURY**

**Président : M. TOUMI Salah**

**Prof. Université Annaba**

**Examineur : M. FEZARI Mohamed**

**M.C(A) Université Annaba**

**M. BENOURET Mohamed**

**M.C (A) Université Annaba**

**M. BOUGHAZI Mohamed**

**M.C (A) Université Annaba**

# Remerciements

Je tiens tout d'abord à remercier l'ensemble des enseignants du département d'électronique de l'université Badji Mokhtar Annaba pour m'avoir donné l'occasion de terminer mes études.

Je remercie le professeur Nouredine DOGHMANE, mon directeur de mémoire, pour m'avoir dirigé dans ce travail et accepté d'être toujours présent durant toutes ces années. Ses conseils m'ont été très bénéfiques.

Je tiens à exprimer mes vives reconnaissances et tout mon respect envers le professeur Salah TOUMI pour avoir accepté d'être le président de mon jury de magister.

Egalement je remercie chaleureusement messieurs les membres du jury, en l'occurrence M. Mohamed FEZARI, M. Mohamed BENOURET et M. Mohamed BOUGHAZI pour avoir accepté de juger ce travail et d'être membres de ce jury.

---

الرقمية عملية منها تقليل حجمها.  
يترافق هذا التدهور. الهدف أساليب جديدة هو توفير  
بين عالية تدهور. بين تقنيات أساليب  
. DWT هي التغييرات العديد تقنيات  
والمعايير ( JPEG 2000 EZW SPIHT ... ) . SPIHT  
DWT بطريقة تدريجية. يهدف وتقييم هذه الخوارزمية  
الرمادية . ناحية مهتمون تغيير التقنيات  
تعقيدها الرياضية. وهذا يؤدي تنفيذ أجهزة

# Résumé

La compression des images numérique est devenue une opération inévitable afin de diminuer leurs tailles en vue d'une transmission et/ou stockage. Cette compression s'accompagne souvent d'une certaine dégradation. L'objectif visé pour proposer de nouvelles techniques de compression est d'assurer un compromis entre un taux de compression élevé et une dégradation moindre. Parmi les techniques de compression les plus utilisées nous avons les méthodes basées sur les transformations. La DWT est l'une des transformations les plus sollicitées dans plusieurs techniques et normes de compression (exemple : jpeg2000, EZW, SPIHT ...etc). la méthode de SPIHT, basée sur la DWT, opère d'une manière progressive. Notre travail de magister a pour objectif l'utilisation et l'évaluation des performances de cet algorithme sur des images en niveaux de gris et en couleur. D'un autre côté nous nous intéressons à modifier certaines techniques de compression locales dans le but de réduire leurs complexités mathématiques. Ceci pourra avoir comme conséquence une possible implémentation dans des réseaux de capteurs d'images sans fil.

# Abstract

The digital image coding has become an inevitable process in order to reduce their size for transmission and / or storage. This compression is often accompanied by some degradation. The aim to propose new compression techniques is to provide a compromise between a high compression ratio and less degradation. Among the compression techniques we have the methods based on transformations. The DWT is one of the transformations used in some compression techniques (eg JPEG2000, EZW, SPIHT ... etc). The method SPIHT, based on the DWT, operates in a progressive way. Our work has as aim: the using and the evaluation of performance of this algorithm on grayscale images and color images. On the other hand we are interested in changing some local compression techniques to reduce their mathematical complexities. This may result in a possible implementation in networks of wireless image sensors.

# Liste des Symboles

---

RIF	Response Impulse Filter
WSN	Wireless Sensor Network
PDA	Personal digital Assistant
RDC	Réseaux de Capteurs
DCT	Discrete Cosine Transform
JPEG	Joint Photographic Experts Group
S-JPEG	Square JPEG
JPEG2000	Joint Photographic Experts Group 2000
DWT	Discrete Wavelet Transform
SPIHT	Set Partitioning In Hierarchical Trees
MSB	Most Significant Bit
RC	Rapport de Compression
RMSE	Root-Mean-Square Error
SNR	Signal-to-Noise Ratio
PSNR	Peak Signal-to-Noise Ratio
LZ77	Lempel-Ziv-77
LZW	Lempel-Ziv-Welch
ARJ	Archived by Robert K. Jung
GIF	Graphics Interchange Format
PNG	Portable Network Graphics
QS	Quantification Scalaire
QV	Quantification Vectorielle
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
DWT	Discrete Wavelet Transform
QMFB	Quadrature Mirror Filters Bank
EZW	Embedded coding with Zerotree of Wavelet coefficients
POS	Positive
NEG	Négative
AZ	Arbre de Zéros
ZI	Zéro Isolé
LSB	Least Significant Bit
LCS	Liste des Coefficients Significatifs
LCN	Liste des Coefficients Non Significatifs
LEN	Liste des Ensembles Non Significatif
SPECK	Set-Partitioning Embedded Block Coder
CCD	Charge Coupled Device

---

# Liste des Figures

---

Figure 2.1	Schéma de principe de la compression par transformée	14
Figure 2.2.	Schéma de principe de la multirésolution.	19
Figure 2.3	Représentation des coefficients d'approximation et de détails	21
Figure 2.4	Représentation des plans de bits .....	22
Figure 2.5	Structure d'arbre des coefficients d'ondelette	23
Figure 2.6a	Courbe PSNR en fonction du débit obtenu par SPIHT pour l'image Pepper	34
Figure 2.6b	Courbe PSNR en fonction du débit obtenu par SPIHT pour l'image Barbara	34
Figure 2.6c	Courbe PSNR en fonction du débit obtenu par SPIHT pour l'image Boat	35
Figure 2.6d	Courbe PSNR en fonction du débit obtenu par SPIHT pour l'image Baboon	35
Figure 2.7	Résultats de compression de l'image Pepper par SPIHT	36
Figure 2.8	Résultats de compression de l'image Barbara par SPIHT	37
Figure 2.9	Résultats de compression de l'image Boat par SPIHT	38
Figure 2.10	Résultats de compression de l'image Baboon par SPIHT	38
Figure 2.11a	PSNR en fonction du débit obtenu par SPIHT pour l'image Pepper	39
Figure 2.11b	PSNR en fonction du débit obtenu par SPIHT pour l'image Barbar	41
Figure 2.11c	PSNR en fonction du débit obtenu par SPIHT pour l'image Boat	41
Figure 2.11d	PSNR en fonction du débit obtenu par SPIHT pour l'image Babbon	42
Figure 2.12	Résultats de compression de l'image Pepper couleur par SPIHT	42
Figure 2.13	Résultats de compression de l'image Boat couleur par SPIHT	43
Figure 3.1	Schéma d'un réseau de capteurs sans fil	44
Figure 3.2	Constitution d'un nœud de capteur sans fil	49
Figure 3.3	Nœud de capteur Mica2 et carte de capture	53
Figure 3.4	Architecture d'un réseau de capteurs d'images sans fil	54
Figure 3.5	Classification des méthodes de compression	58
Figure 4.1	Principe de base du codeur JPEG	61
Figure 4.2	Matrice de quantification du standard JPEG	64
Figure 4.3	Balayage zig-zag du standard JPEG	65
Figure 4.4	Consommation énergétique d'un capteur d'images exécutant le JPEG	65
Figure 4.5	Schéma de principe de la compression zonale	66
Figure 4.6	Forme carrée de la DCT zonale	70
Figure 4.7	Forme triangulaire de la DCT zonale	70
Figure 4.8	Les résultats de la DCT zonale sur l'image Barbara (zone carrée)	71
Figure 4.9	Les résultats de la DCT zonale sur l'image Barbara (zone triangulaire)	76
		76

---

# Liste des tableaux

---

Tableau 2.1	Les PSNR en fonction des débits obtenus par SPIHT	33
Tableau 2.2	PSNR en fonction des débits obtenus par SPIHT pour des images en couleurs.	40
Tableau 3.1	Exemples et caractéristiques de nœuds de capteurs sans fil	54
Tableau 4.1	Les PSNR en fonction des débits obtenus par une DCT zonale de forme carrée avec $\rho=6$ .	72
Tableau 4.2	Les PSNR en fonction des débits obtenus par une DCT zonale triangulaire avec $\rho=6$ .	73
Tableau 4.3	Les PSNR en fonction des débits obtenus par une DCT zonale de forme carrée avec $\rho=4$ .	73
Tableau 4.4	Les PSNR en fonction des débits obtenus par une DCT zonale triangulaire avec $\rho=4$ .	73
Tableau 4.5	Les PSNR en fonction des débits obtenus par une DCT zonale de forme carrée avec $\rho=2$ .	74
Tableau 4.6	Les PSNR en fonction des débits obtenus par une DCT zonale triangulaire avec $\rho=2$ .	74
Tableau 4.7	Les PSNR en fonction de $\rho$ pour une forme carrée et un débit de 0.5bpp	75
Tableau 4.8	Les PSNR en fonction de $\rho$ pour une forme triangulaire et un débit de 0.5bpp	75
Tableau 4.9	Rendement énergétique entre JPEG et JPEG zonale	78

# Sommaire

Introduction générale	01
Chapitre1 : Aperçu sur les méthodes de compression	04
1. Introduction	05
2. Les méthodes de compression	05
2.1 Compression sans pertes	05
2.2 Compression avec pertes	06
3. Mesures de performances des méthodes de compression	06
3.1 Rapport de compression et débit	06
3.2 Mesure de la qualité ou PSNR	07
4. Classification des méthodes de compression des images	08
4.1 Codage de type dictionnaire	08
4.2 Codage par prédiction	09
4.3 Codage statistique	09
4.4 Compression par transformée	10
4.5 Compression fractale	10
5. Conclusion	11
Chapitre 2 : Les méthodes de codage progressif	13
1. Schéma général de compression par transformations	14
1.1 Transformation de l'image	14
1.2 Quantification	14
1.3 Coage entropique	15
2. Transformation en cosinus discrète	15
2.1 La DCT 1-D	16
2.2 La DCT 2-D	16
3. Transformation en ondelettes discrète	17
3.1 Inversibilité	18
3.2 Condition d'admissibilité	18
3.3 Implémentation par banc de filtres	19
3.4 Implémentation par "lifting scheme"	21
4. Méthodes de codages progressifs	21
4.1 L'algorithme EZW: Embedded coding with Zerotree of Wavelet coefficients	23

4.2 SPIHT: Set Partitioning In Hierarchical Trees	25
4.3 SPECK: Set-Partitioning Embedded Block Coder	26
5. Algorithme de codage SPIHT	27
5.1 Méthode de codage	28
5.2 Méthode de décodage	30
6. Les images en couleurs	31
7. Résultats de compression	32
8. Conclusion	46
CHAPITRE 3 : Les réseaux de capteurs sans fil (WSN)	47
1. Introduction	48
2. Les réseaux de capteurs sans fil	49
3. Domaines d'applications des réseaux de capteurs sans fil	49
• Application militaire	50
• Applications environnementales	50
• Application à la surveillance	50
	51
• Applications médicales	51
	51
4. Constitution d'un nœud capteur sans fil	51
	51
4.1. Architecture d'un nœud capteur sans fil	52
	52
• Unité de Captage	53
	55
• Unité de traitement	55
	55
• Unité de Communication	55
• Unité de puissance	56
4.2. Prototypes de nœuds de capteurs	56
4.3. Plateformes logicielles	56
5. Consommation énergétique d'un nœud capteur	56
• Les collisions	56
• L'écoute à vide (idle listening)	56
• L'écoute abusive (overhearing)	57

• La transmission radio	57
6. Problématiques des réseaux de capteurs sans fil	57
• Energie	58
• Ressources limitées des nœuds	59
• Environnement de communication non contrôlable	59
7. Les réseaux de capteurs d'images sans fil	59
• Réseaux de capteurs d'images fixes	59
• Réseaux de capteurs de vidéo	59
8. Spécificités des réseaux de capteurs d'images	59
• Capture du signal	60
• Besoins de mémoire	61
• Traitement du signal	62
• Transmission d'image	63
9. Compression d'image dans les réseaux de capteurs sans fil	63
9.1. Introduction	64
9.2. Classification des algorithmes de compression d'image pour réseaux de capteurs	65
10. Conclusion	65
CHAPITRE 4 : Algorithme de Compression à faible consommation	66
1. Introduction	66
2. Algorithme de compression JPEG	66
• La transformée par DCT	67
• Quantification	67
• Codage en ZigZag	68
• Codage par RLE et Huffman	69
3. Consommation énergétique du JPEG	69
4. Temps d'exécution du JPEG	70
5. Modèle de la consommation énergétique du JPEG	70
5.1 Modèle énergétique de la DCT2D	71
5.2 Modèle énergétique de la quantification	77
5.3 Modèle énergétique du balayage Zigzag	77
6. Principe de l'algorithme JPEG Zonale	77
6.1. Méthode S-JPEG	78

6.2. Méthode T-JPEG	79
7. Comparaison des performances des deux formes	81
8. Calcul du gain énergétique entre JPEG et JPEG zonale	
Modèle énergétique de la DCT2D zonale	
Modèle énergétique de la quantification en prenant en considération la zone	
Modèle énergétique du balayage en Zigzag	
9 . CONCLUSION	
CONCLUSION GENERALE	
REFERENCES BIBLIOGRAPHIQUES	

## **INTRODUCTION GENERALE**

Depuis toujours l'image était un outil et un moyen de communication pertinent et efficace. De nos jours il s'agit d'images numériques dont l'utilisation a conquis tous les domaines et à tous les niveaux. A cet effet, le traitement numérique d'images s'est imposé, depuis plusieurs décennies, comme une spécialité à juste titre et dont les applications touchent pratiquement tous les domaines techniques. Les images numériques sont caractérisées par plusieurs paramètres dont la définition et la résolution. La définition d'une image représente en quelque sorte sa taille en nombre de pixels par ligne et par colonne. Quand à la résolution c'est ce nombre de pixel par unité de mesure en l'occurrence le pouce (ou inch). Ainsi, une image numérique de résolution supérieure voit sa taille augmenter inévitablement. Comme chaque pixel d'une image est codé sur un certain nombre d'octet. A titre d'exemple un octet par pixel pour les images en niveaux de gris et trois octets par pixels pour les images en couleurs. Dès lors, nous remarquons facilement que l'handicap majeur des images numériques et leurs tailles en octet qui va devenir très rapidement très volumineuse en fonction d'une haute résolution. Cette taille très élevée devient un véritable "fardeau" lorsqu'on doit traiter, transmettre ou stocker ce type d'images numériques. En effet, les bandes passantes des canaux de transmission sont généralement limitées et ne peuvent pas assurer des débits de transmission élevés. Tout comme les supports de stockage (disques durs, CDROM DVD, ...etc), malgré les avancées technologiques, restent toujours limités et ne permettent pas l'enregistrement de

volumes de données exagérés. La solution à toutes ces limites est bien sur la compression. Il s'agit d'exploiter les redondances possibles dans les images, ou tout autre type de données, afin de réduire leurs tailles le plus possible. Il est évident qu'il est nécessaire de perdre une certaine partie de l'information contenue dans ces images numériques afin de pouvoir atteindre des taux de compression plus importants. C'est ce que l'on appelle alors la compression irréversible ou encore la compression avec perte. D'un autre côté il existe d'autres moyens de compression sans perte, dites réversibles, utilisées principalement pour les données qui ne peuvent tolérer aucune dégradation ou erreur.

Parmi les méthodes de compression avec perte il y'a celles qui sont basées sur des transformations orthogonales. Plusieurs techniques de compression de ce type, depuis les années 80, sont devenues des standards et des normes comme par exemple JPEG, JPEG2000 ...etc (pour le cas des images). L'objectif de la transformation est de passer d'un domaine spatial où les pixels de l'image sont fortement corrélés à un domaine dit transformé où les coefficients sont pratiquement décorrélés.

Nous pouvons citer, parmi les méthodes par transformées :

- La méthode JPEG basée sur la transformation en cosinus discrète (*DCT : discrete cosine transform*).
- La méthode JPEG2000 basée sur la transformée en ondelette discrète ou DWT.

Ces méthodes de compression par transformée ont bien évidemment prouvé leurs efficacités pour des applications classiques (avec des moyens de traitement numériques performants). C'est ce que l'on appelle les méthodes de compression locales. Néanmoins, avec l'avancée technologique de nouveaux équipements ont vu le jour. Il s'agit essentiellement de réseaux de capteurs sans fil comme par exemple les réseaux de capteurs d'images sans fil. Ces petits équipements travaillent en réseau étalés sur de vastes zones et sont dotés de certaines qualités technologiques relativement limitées (processeurs de faibles performances, autonomies énergétiques limitées, bandes passantes faibles ...etc). La transmission d'images numériques à travers ce type de capteurs se trouve impossible aussi bien pour des images non compressées que pour des images compressées avec des schémas de compression classiques. En effet, la complexité mathématique des algorithmes de compression classiques devient une véritable problématique et une limitation à ces équipements dont les caractéristiques technologiques ne sont pas suffisantes. Il est donc tout à fait naturel de penser à de nouvelles approches pour la

compression des images dans le cas de réseaux de capteurs sans fil. Le compromis PSNR / débit, souvent utilisés comme critères de performances pour les algorithmes de compression d'images, n'est plus suffisant tout seul. Il faut prendre en considération la complexité de l'algorithme qui doit être la moins élevée possible afin d'économiser entre autres l'autonomie énergétique du capteur d'une part, et satisfaire la condition d'une bande passante limitée d'autre part.

Nous avons donc choisi de diviser ce mémoire de magistère en quatre chapitres :

- Le premier chapitre donne d'une manière très succincte un aperçu sur les méthodes de compression.
- Le second chapitre met l'accent sur les techniques de codages progressifs. Nous nous intéressons plus particulièrement à la technique SPIHT basée sur la DWT. Dans ce chapitre nous commençons par présenter la méthode SPIHT. Ensuite nous mettons en œuvre cet algorithme et nous étudions ses performances à l'aide de plusieurs images testes. Les images que nous avons utilisées pour nos tests et pour l'évaluation de cet algorithme ont été choisies selon leurs caractéristiques. Ceci dans l'objectif d'évaluer les performances de cette méthode selon plusieurs cas de figures. D'un autre côté les mêmes images ont été utilisées en niveaux de gris et ensuite en couleur pour valider le fonctionnement de cet algorithme pour les deux types d'images.
- Le troisième chapitre est consacré essentiellement aux généralités et aux différentes technologies et protocoles de communication dans les réseaux de capteurs sans fil ou RDC sans fil (en anglais WSN). Nous mettons l'accent sur les problématiques et les limites des RDC. Dans ce même chapitre nous exposons en détail les Réseaux de capteurs d'images qui représentent le point crucial de notre travail
- Le quatrième chapitre traite des versions adaptées des algorithmes de compression classiques, basées sur la DCT surtout, pour pouvoir les implémenter dans des nœuds de capteurs d'images sans fil.

# CHAPITRE 1

## *Aperçu*

# *sur les méthodes de compression*

## **6. Introduction**

Aujourd'hui nous vivons une évolution technologique de grande ampleur. Elle a touché pratiquement tous les domaines et à tous les niveaux. Parmi les points forts de cette évolution les systèmes de transmission numériques sous leurs diverses formes. En effet, cette révolution numérique à inonder, entre autres, notre quotidien par des images digitales de toutes sortes et pour diverses applications. Cependant, toutes ces données numériques, plus particulièrement les images et la vidéo, présentent un inconvénient majeur lié essentiellement au volume d'informations numériques qu'il est nécessaire de traiter, transmettre et/ou stocker. La transmission d'une telle quantité de données nécessite des débits très élevés directement en contradiction avec les bandes passantes des canaux généralement limitées. De même, le stockage de toutes ces données numériques exigent des supports de tailles importantes.

L'une des solutions indispensables dans ce cas est la compression. Cependant, cette compression s'accompagne souvent d'une sorte de dégradation. D'autant plus que cette altération des données et des images compressées augmente lorsque nous désirons atteindre des taux de compression plus élevé. La problématique de la compression en imagerie numérique est bien entendu trouver le meilleur compromis Débit/distorsion [1] [2].

## **7. Les méthodes de compression**

La compression peut être définie comme étant un système dont l'entrée est une image sans compression et la sortie est un flux de données numériques relativement court représentant l'image compressée. Le processus inverse est appelé décompression permettant la reconstruction de l'image à partir du flux de données numériques. Parfois, les systèmes de compression et de décompression ensemble sont appelés "*Codec*" (codage pour compression et décodage pour décompression). Suivant la qualité de sortie de ces systèmes on distingue deux types de méthodes de compression.

### ***7.1 Compression sans pertes***

Quand on parle de compression sans perte (*Lossless Compression*) on fait souvent allusion aux opérations de codage. Dans ce cas les données compressées et décompressées sont tout à fait équivalentes. Ces techniques de compression sans pertes ou de codage sont qualifiées de *compression réversible*. Elles sont plus particulièrement utilisées pour les données informatiques dont l'exactitude est décisive (Exemple : données textuelles, programmes, images médicales...). Normalement, les facteurs de compression accomplis par ces algorithmes n'aboutissent qu'à des valeurs faibles de l'ordre de 2 à 3 au maximum.

### ***7.2 Compression avec pertes***

Quant aux méthodes de compression avec pertes (*Lossy Compression*) elles sont basées essentiellement sur l'élimination d'une partie de l'information jugée redondante. Ceci a pour effet l'obtention de meilleurs taux de compression, de plus en plus élevés, au détriment d'une certaine dégradation. Pour cette raison ces techniques de compression sont dites irréversibles. En effet, l'image originale et l'image reconstruite après compression ne sont pas exactement les mêmes.

## **8. Mesures de performances des méthodes de compression**

Comme tout autre système, les mesures de performances d'un algorithme de compression d'images sont des critères importants pour la sélection et l'évaluation d'un algorithme. Les performances des algorithmes de compression peuvent être vues suivant plusieurs perspectives, dépendant des besoins d'applications. Parmi les critères souvent utilisés pour les mesures de performances dans le cas des méthodes de compression nous pouvons citer :

- Qualité objective mesurée entre autres par le SNR (Signal to Noise Ratio), le PSNR (Peak Signal to Noise Ratio), la MSE (Mean Square Error), le RC (Rapport de compression) ...etc.
- La qualité subjective des images reconstruites basée surtout sur la qualité visuelle
- La complexité relative de l'algorithme,
- La vitesse d'exécution
- ... etc.

### 8.1 Rapport de compression et débit

Le *rapport de compression* (RC) est la mesure de performance la plus populaire pour un algorithme de compression. Il est défini comme le rapport de nombre des bits nécessaires pour représenter les données originales au nombre de bits dans les données compressées. Considérons une image en niveaux de gris 256x256 pixels, si chaque pixel est représenté sur 01 octet, elle aura besoin de 65536 octets pour son stockage. Si la version compressée de l'image nécessite seulement 4096 octets, alors le rapport de compression achevé par cet algorithme est de 16 :1.

$$RC = \frac{\text{nombre de bits dans l'image originale}}{\text{nombre de bits retenus}}$$

(1.1)

Une autre variation du rapport de compression est le débit en bits par échantillon (*bits per points* ou *bits per pixels*). Cette mesure indique le nombre moyen des bits pour représenter un seule point ou pixel de l'image. Alors, pour l'exemple précédent, on peut dire que l'algorithme atteint un débit de 0.5 bits per pixel (*bpp*). Par conséquent, ce débit peut être mesuré par le rapport du nombre de bits d'un pixel de l'image non compressée sur le rapport de compression.

$$\text{Débit (bpp)} = \frac{\text{nombre de bits d'un pixel}}{RC} \quad (1)$$

Le rapport de compression que l'on peut atteindre par une méthode de compression sans perte est totalement dépendant de l'image d'entrée. Si le même algorithme est appliqué sur des images distinctes, l'algorithme va rendre des rapports de compression différents. Le rapport de compression maximum est limité par l'entropie de l'image

suivant le théorème de Shannon. Par exemple, il est difficile d'atteindre un rapport de compression pour une image dont les pixels sont principalement aléatoires [1] [2] [3].

### 3.2 Mesure de la qualité ou PSNR.

La mesure de la qualité des algorithmes de compression d'images est basée essentiellement sur certains critères comme par exemple [1] [2], [3], [4], [5] :

- la racine carrée de l'erreur quadratique moyenne (*RMSE* : *root-mean-squared error*),
- le rapport signal à bruit (*SNR* : *signal-to-noise ratio*)
- le pic du rapport signal à bruit (*PSNR* : *peak-signal-to-noise ratio*).

Ces critères permettant une comparaison entre le signal ou image original et le signal ou image reconstruite après compression. Dans le cas d'une image supposée de forme rectangulaire de taille  $M \times N$  pixels ces critères seront exprimés comme suit :

$$RMSE = \sqrt{\frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [I(i, j) - \hat{I}(i, j)]^2}$$

(1.3)

Avec  $i$  et  $j$  représentent la position du pixel dans l'image.

La mesure du *SNR* en décibel (*dB*) est calculée par l'expression suivante :

$$SNR = 10 \log_{10} \left( \frac{\sum_{i=1}^M \sum_{j=1}^N I^2(i, j)}{\sum_{i=1}^M \sum_{j=1}^N [I(i, j) - \hat{I}(i, j)]^2} \right)$$

(1.4)

Pour les images 8-bits, le *PSNR* est calculé par l'expression suivante :

$$PSNR = 20 \log_{10} \left( \frac{255}{RMSE} \right)$$

(1.5)

## 9. Classification des méthodes de compression des images

### 4.1 Codage de type dictionnaire [2]

En 1977, Lempel et Ziv propose un algorithme de codage universel [4]. Cet algorithme permet automatiquement de coder une suite de  $n$  symboles issues d'une même source  $X$  avec une longueur moyenne qui tend vers la longueur optimale  $\hat{H}(X) = \lim_{n \rightarrow \infty} \frac{1}{n} H(X^n)$  sous une simple hypothèse de stationnarité de la source. On assure ainsi l'optimalité asymptotique de la longueur moyenne du code sans la connaissance de la loi de la source.

L'algorithme proposé par Lempel et Ziv (LZ77), comme ses nombreuses variantes (LZ78, LZW, ...), est l'un des plus simples à programmer. Le principe de ces méthodes est de parcourir la chaîne de symboles et de coder les nouvelles occurrences des sous-chaînes déjà observées par une simple référence à l'occurrence précédente.

Le terme de codage, de type dictionnaire, provient de l'implémentation dans laquelle un dictionnaire de sous-chaînes est construit au fur et à mesure du parcours de la chaîne. Sans mentionner toute la littérature existante sur le sujet, il suffit de noter que c'est la famille utilisée dans les algorithmes de compression de données de type ZIP (ou ARJ) pour en mesurer le succès.

L'algorithme de compression d'image GIF est basé sur cette technique, plus précisément sur la variante LZW, proposée par Welsh [5] en 1984. En 1987, CompuServe, le leader des fournisseurs d'accès au réseau, introduit ce format afin d'accélérer la transmission des images sur celui-ci. Pour compresser une image, il suffit de réordonner la matrice des pixels en une liste de valeurs et de compresser cette liste à l'aide de l'algorithme LZW. Cette technique simple est suffisante pour obtenir un taux de compression de l'ordre de 2 et donc de réduire de moitié le temps de transmission [6].

### 4.2 Codage par prédiction [2]

Les méthodes prédictives proposent de transformer de manière réversible les chaînes de symboles en des chaînes plus simples pour l'algorithme de compression. Il s'agit d'aider celui-ci en introduisant un modèle permettant de prédire chaque symbole apparaissant dans la chaîne en fonction des symboles précédents et de coder à l'aide de l'algorithme de codage universel l'erreur de prédiction au lieu du symbole lui-même.

Ceci suppose que les symboles possèdent une interprétation, ce qui est le cas pour les images. Il n'est donc pas étonnant que, en 1995, lorsque l'algorithme PNG, conçu

comme une alternative non encombrée par des brevets de GIF, a été proposé, cette prédiction ait été ajoutée. Les modèles proposés utilisent des prédictions linéaires de l'intensité lumineuse en fonction de celle des voisins déjà connus. Les erreurs de prédictions sont alors codées par une autre variante de l'algorithme LZ77, c'est l'algorithme deflate. L'amélioration de performances est notable puisque l'on peut atteindre des taux de l'ordre de 3 avec l'algorithme PNG [6].

### ***4.3 Codage statistique [2]***

Les techniques de codage universel ne font pas intervenir la distribution de probabilité de la source mais s'y adaptent asymptotiquement. A l'opposé, si l'on connaît cette distribution, on peut construire un code quasi optimal. Deux algorithmes se disputent la prédominance pour ces codages dits statistiques : le plus ancien, l'algorithme de Huffman [7], est simple et efficace mais il est moins performant que l'algorithme de compression arithmétique [8], dont la complexité est plus grande. Le choix entre ces deux algorithmes se fait suivant les contraintes de performance et de complexité.

La voie de la modélisation statistique est une voie intermédiaire. Elle remplace la distribution inconnue par un modèle connu et utilise cette nouvelle distribution pour coder les symboles à l'aide d'un algorithme de codage statistique.

Si le modèle n'est pas trop éloigné de la réalité, il permet de compresser les données efficacement sans avoir recours à un comportement asymptotique.

Les modèles utilisés varient en complexité : les modèles les plus simples font l'hypothèse que tous les éléments de la chaîne sont indépendants et identiquement distribués tandis que les modèles les plus complexes conditionnent le choix de la distribution de probabilités pour un nouveau symbole à tous ceux déjà codés dans le passé.

Les meilleurs résultats de compression sans perte sont obtenus avec des modèles contextuels de type chaîne de Markov où la loi utilisée dépend du voisinage et est apprise au fur et à mesure du parcours de l'image. Ces modèles permettent des taux de compression dépassant 4 au prix d'un algorithme complexe et lent [6].

### ***4.4 Compression par transformées***

Les méthodes de transformations agissent, non pas dans le domaine spatial sur l'image numérique, mais dans le domaine transformé sur une autre représentation de l'image originale. Après avoir calculé la transformation de l'image d'entrée on obtient les coefficients transformés. Ensuite, il faut quantifier les coefficients pour qu'ils prennent moins de bits et ainsi le flux obtenu aura une taille plus faible que l'image originale.

La compression de l'image consiste à négliger les coefficients les moins significatifs (de point de vue amplitude), tout en garantissant une qualité acceptable de l'image reconstruite par transformation inverse. Si les coefficients dont l'amplitude est en dessous d'un certain seuil sont éliminés, il faut indiquer également leur adresse ou celle des coefficients retenus.

Dans certains cas, selon la transformation et l'image, les coefficients que l'on peut négliger se trouvent dans des régions bien définies du domaine transformé. On peut alors éviter l'indication des adresses des coefficients retenus ou éliminés [9].

#### ***4.5 Compression fractale [2]***

Contrairement aux autres techniques de compression habituelles, la compression fractale ne tente pas de réduire le nombre de couleurs (format GIF) ou de compresser de manière classique les octets composant l'image. Le principe est de remplacer l'image par des formules mathématiques.

La compression fractale a pour principe qu'une image n'est qu'un ensemble de motifs identiques en nombre limité, auxquels on applique des transformations géométriques (rotations, symétries, agrandissements, réductions). Evidemment, plus l'image possède cette propriété, meilleur sera le résultat.

Comme pour le format JPEG, l'image est découpée en blocs de pixel, mais ils sont ici de tailles variables. Il faut ensuite détecter les redondances entre ces blocs à diverses résolutions. On parle de transformations fractales basées sur un opérateur contractant. Ces transformations décrivent l'image de plus en plus finement. A la fin de ce processus, on ne stocke pas le contenu d'un bloc autant de fois qu'il a été "vu" dans l'image mais seulement les équations mathématiques permettant de représenter le contenu de ces blocs.

Au final on obtient une structure présentant des caractéristiques similaires à des échelles différentes. Pour retrouver l'image il suffira de décrire les transformations qui ont été appliquées aux blocs initiaux. Ce processus rend la compression indépendante de la taille de l'image. De plus, l'image produite est vectorisée et ne subit pas les effets de la pixellisation, contrairement au JPEG. Ce phénomène est surtout visible lors d'un zoom par exemple, l'image fractale peut devenir floue mais ne pixélise pas. Ceci est dû au fait que lors de l'agrandissement, ce ne sont pas les pixels qui sont élargis, mais toute l'image qui est recalculée mathématiquement.

Le problème lié à cette technique est la lenteur du procédé de compression, de l'ordre de 50 fois plus lent que pour une image JPEG. La décompression quant à elle est aussi rapide que pour les autres formats [10].

## **5. Conclusion**

Dans ce chapitre nous avons fait un tour d'horizons, d'une manière très succincte, sur les méthodes de compression les plus utilisées et les plus sollicitées à l'heure actuelle surtout dans les applications multimédias liées aux réseaux de communication (exemple internet). Nous avons, bien évidemment, noté clairement qu'il s'agit souvent de deux familles de méthodes de compression à savoir les techniques de compression sans pertes et avec pertes. La première famille est dite réversible dans le sens où les données reconstruites après les phases de compression et décompression sont tout à fait identiques aux données originales. Tandis que la deuxième famille de méthodes elle admet une certaine perte d'information, jugée non significative, au détriment d'un taux de compression élevé. Les méthodes de la seconde famille, qui nous intéressent plus particulièrement, sont souvent évaluées en fonction du rapport débit/distorsion. En effet, si on arrive à obtenir des taux de compression élevés cela signifie que l'on atteint des débits élevés. Seulement, au prix d'une certaine altération qui doit être la plus faible possible. Nous avons également rappelé qu'il existe bien d'autres critères d'évaluation de ces techniques et qui peuvent, dans certains cas, donner l'impression qu'ils sont contradictoires.

# CHAPITRE 2

## *Les Méthodes de Codage progressifs*

### **1. Schéma générale de compression par transformations**

La majorité des systèmes de compression d'image pouvant garantir des taux de compression élevés, en particulier les méthodes avec pertes, opèrent selon trois étapes de traitement (Figure 2.1) [2]. Ces méthodes tentent d'exploiter au mieux la redondance spatiale dans les images souvent très importante. A cet effet, ces méthodes commencent

par passer du domaine de représentation habituelle des images, à savoir le domaine spatial, à un domaine transformé où les coefficients sont très peu corrélés. Cette première étape est assurée par une transformation mathématique suivie par une seconde étape dite quantification. Cette seconde étape permet d'obtenir la compression désirée mais elle dégrade de manière irréversible l'image. La dernière étape de ces systèmes est un codage entropique (sans perte) qui produit le flux binaire représentant les données compressées.



Figure 2.1. Schéma de principe de la compression par transformée

### ***1.1 Transformation de l'image***

Cette étape permet de passer dans un domaine transformé où les coefficients sont complètement décorrélés les uns des autres. D'autre part, elle permet un compactage de l'énergie dans le sens où les coefficients obtenus sont concentrés dans une zone plutôt réduite. On parle alors de transformation avec représentation creuse.

Le but d'une transformation est de projeter le signal sur une base de fonctions dont les propriétés sont adaptées à la nature et aux caractéristiques du signal ou de l'image que l'on désire analyser. Cette projection est généralement orthogonale pour garantir une décorrélation des coefficients obtenus. [11].

### ***1.2 Quantification***

Dans le schéma de compression, l'étape de quantification est celle qui dégrade de manière irréversible le signal. Elle est cependant d'une importance capitale dans la réduction du débit binaire. La quantification est une opération qui transforme l'image d'entrée à l'aide d'un ensemble de valeurs appelé dictionnaire. Ce passage peut s'effectuer soit :

- échantillon par échantillon : dans ce cas on parlera de quantification scalaire (QS),
- ou bloc par bloc : c'est ce que l'on appelle la quantification vectorielle (QV).

### **1.3 Codage entropique**

Le codage entropique est utilisé dans une chaîne de compression sans perte, directement sur l'image. Il est également employé à la dernière étape de la chaîne de compression avec pertes (Figure 2.1) afin d'exploiter les redondances présentes à la sortie du quantificateur.

Les codes entropiques sont basés sur la génération de mots dont la longueur dépend de la probabilité d'apparition des symboles de la source qu'il représente (on parle également de codes à longueur variable) : un grand nombre de bits sera utilisé pour coder un symbole peu probable tandis qu'un symbole redondant sera codé sur très peu de bits. C'est le principe de base de la théorie de l'information. Le code ainsi obtenu doit être en outre uniquement décodable.

Il existe de nombreuses méthodes permettant de générer un code entropique, parmi lesquels le célèbre code de Huffman [1] [2] [11]. Cet algorithme admet cependant un inconvénient: il repose sur la connaissance complète de la source à coder (il existe cependant des versions adaptatives). D'autre part, il ne permet pas d'atteindre des débits inférieurs à 1 bit/échantillon.

Une autre approche du codage entropique, ne présentant pas l'inconvénient du codage d'Huffman, à savoir le codage arithmétique. La différence fondamentale entre les deux réside dans le fait que ce dernier codage, les symboles ne sont pas codés séparément. En effet, c'est l'ensemble du message à transmettre qui est construit au fur et à mesure du traitement des différents éléments de la source [11].

## **2. Transformation en cosinus discrète**

La transformation en cosinus discrète, aussi appelée DCT (*Discrete Cosine Transform*) est une transformation linéaire orthogonale. Elle est considérée comme une version simplifiée de la transformation de Fourier discrète (DFT). En effet, le noyau de projection utilisé pour la DFT est représenté par une exponentielle complexe (soit des bases de sinus et cosinus) alors que le noyau de projection de la DCT est simplement une base de cosinus. Les coefficients de la transformée ne sont donc pas complexes mais réels ce qui présente un avantage pour le codage et la quantification [12] [2].

### **2.1 La DCT 1-D**

La DCT d'une séquence  $x(n); n = 0,1,\dots, N-1$  est définie par :

$$X(k) = \sqrt{\frac{2}{N}} \cdot c_k \sum_{n=0}^{N-1} x(n) \cdot \cos\left(\frac{(2n+1)k\pi}{2N}\right), \quad k = 0,1,\dots, N-1. \quad (2.1)$$

et la transformation inverse est définie par :

$$x(n) = \sqrt{\frac{2}{N}} \cdot \sum_{k=0}^{N-1} c_k X(k) \cdot \cos\left(\frac{(2n+1)k\pi}{2N}\right) \quad (2.2)$$

$$\text{avec } c_k = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0. \\ 1, & \text{ailleurs.} \end{cases} \quad (2.3)$$

La transformation en cosinus discrète DCT utilise une matrice de transformation fixe dont les vecteurs de base sont approximativement proches de la classe des matrices à laquelle appartient la KLT (Karhunen Loeve Transform) [13]. Elle approxime la KLT dans le cas d'une grande corrélation des signaux d'entrées. A cause de la nature fixe de cette matrice, elle n'est pas adaptée au signal d'entrée comme dans le cas de la KLT [14].

## 2.2 La DCT 2-D

La DCT bidimensionnelle est d'un grand intérêt pratique qui a déjà montré son efficacité. C'est une transformation très populaire pour le codage d'images, comme le montre son adoption par la norme internationale JPEG pour la compression des images fixes.

La DCT d'un bloc de dimension  $(n \times n)$  est donnée par [15]:

$$F(u,v) = \frac{4.C(u).C(v)}{n^2} \sum_{j=0}^{n-1} \sum_{k=0}^{n-1} f(j,k) \cdot \cos\left[\frac{(2j+1)u\pi}{2n}\right] \cdot \cos\left[\frac{(2k+1)v\pi}{2n}\right] \quad (2.4)$$

et la transformation inverse est donnée par :

$$f(j, k) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} C(u).C(v).F(u, v). \cos\left[\frac{(2j+1)u\pi}{2n}\right]. \cos\left[\frac{(2k+1)v\pi}{2n}\right]$$

(2.5)

avec

$$C(\omega) = \begin{cases} \frac{1}{\sqrt{2}} & \text{si } \omega = 0. \\ 1 & \text{si } \omega = 1, \dots, n-1. \end{cases}$$

(2.6)

La matrice de transformation de la DCT peut être définie par la formule suivante :

$$A(n, k) = \sqrt{\frac{2}{N}} \cdot C(\omega) \cdot \cos\frac{(2.n+1)k\pi}{2.N}$$

(2.7)

et pour le cas 2-D, la transformation est réalisée d'une manière séparable par la relation :

$$F = AfA$$

(2.8)

Cette propriété de séparabilité vérifiée par la DCT2D, comme c'est le cas pour la DFT2D et bien d'autres transformations discrètes 2D, est d'un grand intérêt pratique. Elle permet d'implémenter ces transformations 2D en utilisant les transformations 1D.

### 3. Transformation en ondelettes discrète [2]

La transformée en ondelettes d'un signal  $x(t)$  peut être définie comme la projection sur la base des fonctions ondelettes.

$$TO(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t) \cdot \psi\left(\frac{t-b}{a}\right) \cdot dt, \text{ avec } a, b \in \mathbb{R}, a \neq 0$$

(2.9)

$$TO(a, b) = \int_{-\infty}^{+\infty} x(t) \cdot \psi_{a,b}(t) \cdot dt, \text{ avec } \psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right)$$

(2.10)

Les coefficients d'ondelettes ( $TO(a,b)$ ) dépendant de deux paramètres  $a$  et  $b$  :  $a$  étant le facteur d'échelle et  $b$  le facteur de translation. À l'échelle «  $a$  » le pas de translation est  $\frac{b}{a}$ .

Les fonctions  $\psi_{a,b}(t)$  sont obtenues à partir de la dilatation et de la translation de la fonction ou ondelette mère  $\psi(t)$ . Les fonctions  $\psi_{a,b}(t)$  sont par conséquent parfois appelées les ondelettes filles. Ces fonctions forment une base, c'est-à-dire que, si l'on note le produit scalaire entre deux fonctions  $x$  et  $h$  comme étant :

$$\langle x | h \rangle = \int x(t)h(t)dt \quad (2.11)$$

Alors on a :

$$\langle \psi_{a_1,b_1} | \psi_{a_2,b_2} \rangle = \delta_{a_1,a_2} \cdot \delta_{b_1,b_2} \quad (2.12)$$

### 3.1 Inversibilité

Tout comme la transformée de Fourier, la transformée en ondelettes est inversible.

$$x(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{a^2} \langle x | \psi_{a,b} \rangle \psi_{a,b} \cdot da \cdot db \quad (2.13)$$

où 
$$C_\psi = 2\pi \int_{-\infty}^{+\infty} |\hat{\psi}(\omega)|^2 \frac{d\omega}{\omega} \quad (2.14)$$

avec  $\hat{\psi}(\omega)$  la transformée de Fourier de  $\psi(t)$ .

### 3.2 Condition d'admissibilité

La fonction ondelette doit vérifier un certain nombre de propriétés. La première d'entre elles se nomme condition d'admissibilité.

Soit  $\psi(t) \in L^2$ , alors

$$\int_{-\infty}^{+\infty} \frac{|\hat{\psi}(\omega)|}{|\omega|} \cdot d\omega < \infty \quad (2.15)$$

Cette condition permet d'analyser le signal, puis de le reconstruire sans perte d'information. La condition d'admissibilité implique en outre que la transformée de Fourier de l'ondelette à la fréquence du continu (pour  $\omega = 0$ ) doit être nulle. Soit,

$$\hat{\psi}(\omega) \Big|_{\omega=0} = 0 \quad (2.16)$$

Ceci implique en particulier deux conséquences importantes :

- La première est que les ondelettes doivent posséder un spectre de type passe-bande.
- La seconde apparaît en réécrivant l'équation (2.16) de façon équivalente sous la forme :

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \quad (2.17)$$

Cette équation montre que  $\psi(t)$  doit être à moyenne nulle. L'ondelette  $\psi(t)$  est donc une fonction à largeur temporelle finie possédant un caractère oscillatoire. On est donc bien en présence d'une petite onde ou *une ondelette* [16].

### 3.3 Implémentation par banc de filtres

Une méthode équivalente et plus efficace pour calculer la transformée en ondelette est de convoluer le signal avec une paire de filtres miroirs en quadratures (QMF) convenablement choisis suivi par un sous-échantillonnage de facteur 2 ou décimation.

Les QMF qui décomposent ainsi le signal sont constitués d'un filtre passe-bas  $H_0$  et un filtre passe-haut  $H_1$ . Ils divisent ainsi la bande passante du signal exactement en son milieu. Les coefficients sont recombinaés pour synthétiser le signal  $x(t)$  par la transformée en ondelette inverse. Elle est obtenus à l'aide d'une opération de sur-échantillonnage de facteur 2 suivie par la paire des filtres QMF  $G_0$  et  $G_1$  [42] [2].

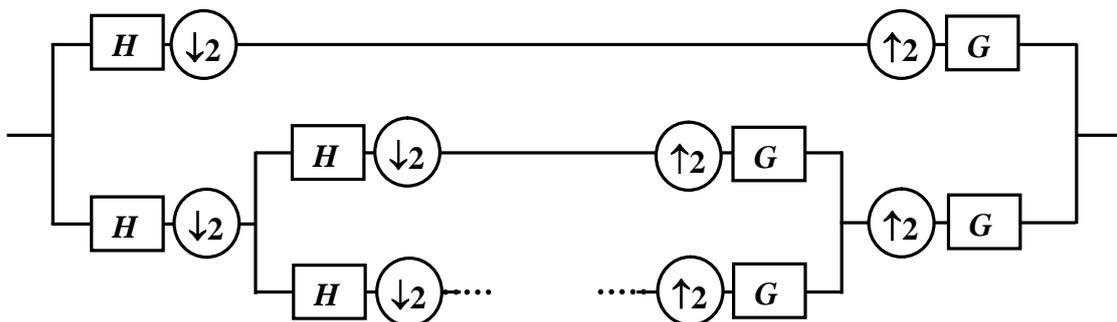


Figure 2.2. Schéma de principe de la Multirésolution [2]

La figure ci-dessus montre les opérations nécessaires pour la décomposition et la synthèse. Les réponses impulsionnelles des paires des filtres d'analyse et de synthèse sont reliées par les relations suivantes :

$$g_0(n) = (-1)^{n-1} \cdot h_1(n-1) \quad (2.18)$$

$$g_1(n) = (-1)^n \cdot h_0(n-1) \quad (2.19)$$

Pour le cas monodimensionnel (1D), les coefficients d'ondelettes à l'échelle  $m+1$  sont calculés par application récursive de la transformée en ondelettes du signal passe bas à l'échelle  $m$ . Après  $k$  niveaux de décomposition on obtient  $k$  ensembles de coefficients hautes fréquences  $C_{m,n}$  et une de bases fréquences, notée  $d_n$  [17].

Dans le cas bidimensionnel (2D), les coefficients d'ondelettes sont obtenus par application de la transformée en ondelettes (1D) sur les deux dimensions du signal à chaque niveaux de décomposition (lignes puis colonnes pour les images numériques). Alors, après  $k$  niveaux de décomposition on obtient  $3k$  parties (sous bandes) de coefficients hauts fréquences ou détails (à chaque niveau trois détails sont obtenus ; horizontales, verticales et diagonales) et une partie de basses fréquences appelée approximation ou bande de base.

C'est ce que l'on appelle habituelle l'analyse multirésolution. Il s'agit de l'application de l'algorithme de Mallat. Le principe est de décomposer un signal en une approximation, appelée également la bande de base ou basse fréquence (L en anglais de "low"), et en ses détails ou composantes hautes fréquences. La même opération est répétée, toujours sur l'approximation uniquement, jusqu'à l'obtention de la résolution souhaitée. S'il s'agit d'une image nous aurons un signal 2D. Dans ce cas chaque niveau de décomposition sera obtenu en deux étapes. La première étape concerne les lignes et la seconde étape les colonnes. Un niveau de décomposition sur une image nous donnera une approximation (Bande de base ou LL1 pour low-Low niveau1) et trois détails respectivement LH1, HL1 et HH1 (pour Low-High niveau1, High-Low niveau1 et High-High niveau1) :

- LH1 : une décomposition basse fréquence selon lignes suivie par une décomposition haute fréquence pour les colonnes. L'indice 1 pour indiquer sur un premier niveau de décomposition.
- HL1 : une décomposition haute fréquence selon lignes suivie par une

décomposition basse fréquence pour les colonnes. L'indice 1 pour indiquer sur un premier niveau de décomposition.

- HH1 : une décomposition haute fréquence selon lignes suivie par une décomposition haute fréquence pour les colonnes. L'indice 1 pour indiquer sur un premier niveau de décomposition

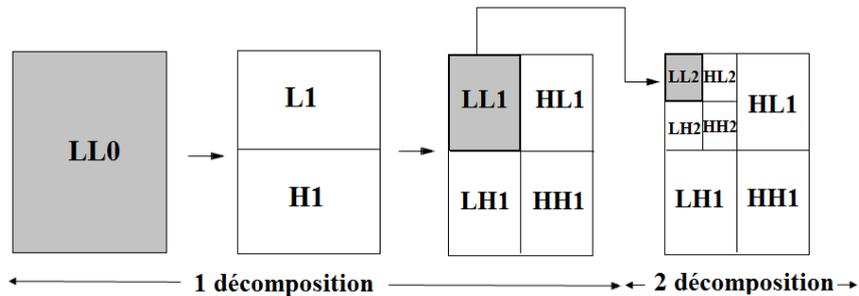


Figure 2.3 : Représentation des coefficients d'approximation et de détails.

### 3.4 Implémentation par le ‘lifting scheme’

Introduit en 1994 par Wim Sweldens, cette implémentation de la transformée en ondelettes ne repose pas sur la transformée de Fourier en s'affranchissant totalement de la notion de dilatation et de translation.

En effet, la méthode classique pour obtenir une transformation en ondelettes est bien évidemment le banc de filtres QMFB. Cependant, on remarque aisément que dans cette technique un échantillon sur deux obtenu par filtrage (après un traitement) est tout simplement supprimé (opération de décimation ou sous-échantillonnage). Autrement dit, la moitié du coup de calcul des opérations de filtrage est réalisée pour rien. L'idée de base du lifting en ondelettes est d'inverser les phases de filtrage et de sous-échantillonnage. Le nombre d'opérations à effectuer est donc réduit. Une autre propriété intéressante est que le schéma de lifting est facilement inversible.

## 4. Méthodes de codages progressifs

Les transformées en ondelettes offrent naturellement une représentation progressive de l'image. Il est intéressant de conserver cette propriété lors du codage des sous-bandes. Ainsi, dans les codeurs emboîtés (*embedded*), la quantification et le codage sont également réalisés de manière progressive, en commençant par coder partiellement les coefficients de plus fortes amplitudes, puis en raffinant la quantification de ces derniers

et en encodant de nouveaux. Nous présentons ici quelques codeurs progressifs basés sur des structures d'arbres ou de blocs.

Ces méthodes appliquent une quantification par approximations successives pour améliorer la précision de la représentation des coefficients d'ondelettes et pour faciliter le codage emboîté. Avec cette approche, la signification des coefficients d'ondelettes pour une série de seuils monotone décroissante  $T_n$  est enregistrée dans un ensemble de cartes binaires, appelées cartes de signification correspondant chacune à un plan de bits. La Figure 2.7 illustre cette notion de plan de bits. Par ailleurs, Shapiro a prouvé que le codage des cartes de signification représente une part importante du coût d'encodage total. Ainsi, en améliorant l'encodage de ces cartes de signification, on peut s'attendre à un gain de codage significatif. La technique utilisée pour encoder les cartes de signification est un algorithme par arbre de zéros, qui permet un codage peu coûteux en termes de débit.

Par ailleurs, ce type de méthodes permet de faire de la transmission progressive (amélioration de la qualité par ajout de bits). En effet, elles peuvent trier l'ordre des bits de codage tel que les bits les plus significatifs soient transmis en premier. Ainsi, pour augmenter la qualité de l'image reconstruite, il suffira d'ajouter de la précision en utilisant les bits supplémentaires.

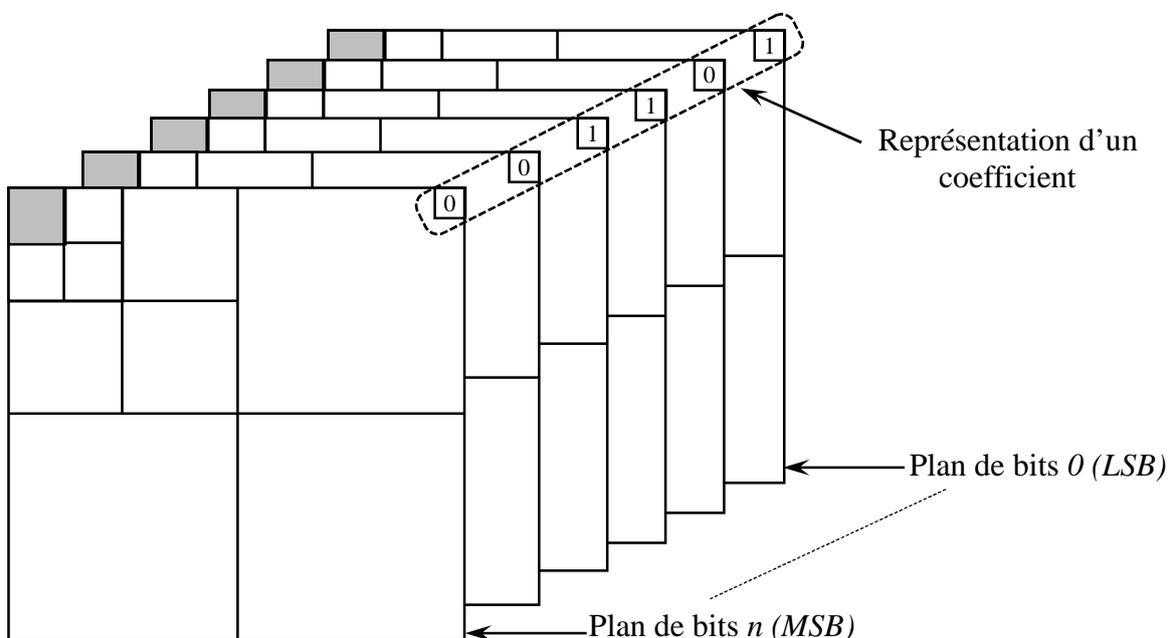


Figure 2.4. Représentation des plans de bits allant des bits les plus significatifs (MSB) vers les bits les moins significatifs (LSB) pour une transformée en ondelettes 2D. [2]

#### 4.1 L'algorithme EZW: Embedded coding with Zero tree of Wavelet coefficients

Le premier algorithme inter-bandes pour les images 2D se nomme EZW [18] [2] proposé en 1993 par Jerrom Shapiro. L'ensemble des méthodes qui ont suivi s'appuient sur des techniques communes. Elles exploitent complètement la notion de multi-résolution associée aux ondelettes. Leur schéma de codage utilise un modèle simple pour caractériser les dépendances inter-bandes parmi les coefficients d'ondelettes localisés dans les sous-bandes ayant la même orientation. La Figure 2.8 illustre ces dépendances à travers toutes les échelles. Le modèle est basé sur l'hypothèse des arbres de zéros, qui suppose que si un coefficient d'ondelettes  $w$  est non significatif pour un seuil donné  $T$ , c'est à dire  $|w| < T$ , alors tous les coefficients de la même orientation dans la même localisation spatiale à des résolutions plus fines sont supposés non significatifs pour ce même seuil  $T$ . Cette hypothèse des arbres de zéros est illustrée en 2D sur la Figure 2.8. On peut voir qu'un coefficient "parent" à une résolution donnée va engendrer 4 coefficients "enfants" à la résolution supérieure [11] [13] [18].

Notons que le nœud racine (correspondant aux coefficients de la dernière sous-bande LL) de l'arbre a seulement 3 enfants, alors que tous les autres nœuds à l'exception des extrémités en possèdent 4. En d'autres termes, à l'exception du nœud racine et des extrémités de l'arbre, le lien parent enfant pour EZW est le suivant :

$$O(i, j) = \{(2i, 2j), (2i, 2j+1), (2i+1, 2j), (2i+1, 2j+1)\} \quad (2.20)$$

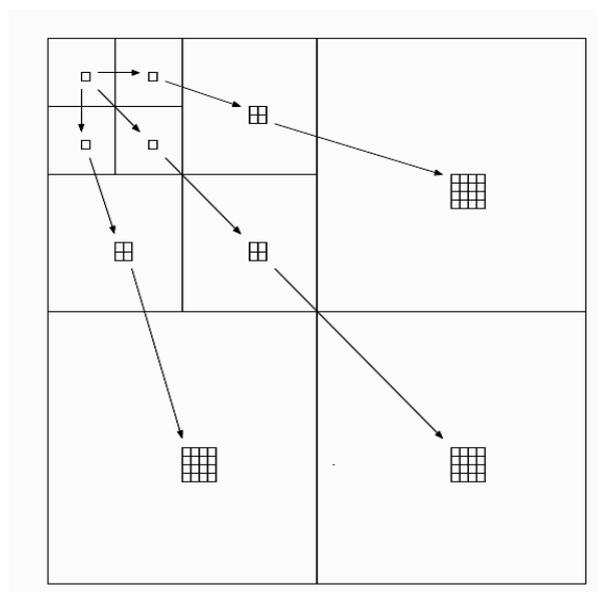


Figure 2.5 Structure d'arbre des coefficients d'ondelette.

La suite de l'algorithme est comme le suivant : on part d'un seuil entier  $T_n$  qui est une puissance de 2 de type  $T_n = 2^n$ , avec  $n = \lceil \log_2(|w_{\max}|) \rceil$  où  $w_{\max}$  est le coefficient le plus grand en valeur absolue de toutes les sous-bandes. Les coefficients d'ondelettes sont scannés dans un ordre hiérarchique de la résolution la plus faible vers la plus grande, et chaque coefficient  $w$  est testé pour savoir si sa valeur absolue est supérieure ou égale au seuil  $T_n$ , c'est-à-dire s'il est significatif pour le seuil donné. Si un coefficient est trouvé significatif, il est codé suivant son signe par le symbole POS pour un coefficient positif et NEG pour un coefficient négatif. Il est alors placé dans la liste des coefficients significatifs.

Si un coefficient est testé non significatif ( $|w| < T_n$ ), on examine tous ses descendants pour tester leur signification. Dans le cas où aucun descendant n'est pas significatif, on code un arbre de zéros (AZ). Si un descendant significatif apparaît on code un zéro isolé (ZI). Les coefficients qui descendent d'une racine d'arbre de zéros n'ont pas besoin d'être codés. Cette partie se nomme la passe dominante pour Shapiro ou passe de signification dans de nombreux articles.

Ensuite, la seconde passe (passe de raffinement ou subordonnée pour Shapiro) est réalisée sur les coefficients dans la liste significative. Pour chaque coefficient de cette liste, le bit situé dans le plan de bit inférieur (plan de bit  $2^{n-1}$ ) est codé. L'encodeur divise le seuil  $T_n$  par 2 ( $T_n \leftarrow T_n / 2$ ) et exécute une nouvelle passe dominante et de raffinement. Cette procédure se poursuit jusqu'à ce que l'on atteigne le débit voulu. Si un coefficient est trouvé significatif à une passe postérieure, il sera encore dans la liste significative à la passe courante et n'aura pas besoin d'être identifié comme significatif une autre fois. Le nombre de coefficients dans cette liste croît de façon monotone au fur et à mesure que les seuils  $T_n$  décroissent. Si l'on va jusqu'au dernier plan de bits (LSB), on obtient un codage sans perte car il n'y a plus d'étape de quantification.

Le décodeur utilise un algorithme similaire. Il initialise tous les coefficients à zéro et scanne à travers les mêmes directions que l'encodeur. Le décodeur reçoit un symbole du flux binaire pour chaque coefficient. Si ce symbole est POS ou NEG, l'amplitude du coefficient est au-dessus du seuil et on détermine le signe. Dans les deux cas, le coefficient est placé dans la liste significative. Si un symbole AZ est reçu, aucun des

descendants du coefficient courant n'est visité pendant la passe dominante.

On réalise ensuite la passe de raffinement. Pour chaque coefficient dans la liste de signification, un bit est sorti du flux binaire. Si c'est un 1, il est utilisé pour remplacer le bit 0 à la localisation  $\log_2(T_n) - 1$  dans la représentation binaire des coefficients [11] [18].

#### ***4.2 SPIHT: Set Partitioning In Hierarchical Trees***

L'algorithme SPIHT [19] reprend les principes évoqués dans EZW tout en proposant de partitionner récursivement les arbres de coefficients. Ainsi, là où EZW codait un coefficient non significatif isolé ('ZI'), SPIHT effectue un partitionnement récursif de l'arbre de manière à déterminer la position des coefficients significatifs dans la descendance du coefficient considéré. Les coefficients significatifs sont codés de manière similaire à EZW : leur signe est envoyé dès qu'ils sont identifiés comme étant significatifs et ils sont ajoutés à la liste des coefficients à raffiner. Cet algorithme fonctionne également par plans de bits. Il offre des performances remarquables, atteignant celles d'EZW sans codage entropique. En ajoutant un codage entropique de l'information de signifiante, un gain supplémentaire entre 0.3 et 0.6 dB peut être obtenu [1].

Les bits envoyés lors de la passe de signifiante correspondent au programme exécuté à l'encodeur lors de l'exécution de l'algorithme de classement en coefficients significatifs et non significatifs. En suivant le même programme, le décodeur reste synchrone avec les décisions de l'encodeur et retrouve la même classification. Cet algorithme repose sur la gestion de trois listes, de coefficients significatifs (LCS), de coefficients insignifiants (LCN) et d'ensembles insignifiants (LEN). Moyennant un seuil de signifiante divisé par deux à chaque itération, et dont la valeur initiale est transmise au décodeur.

L'algorithme se déroule de la manière suivante : la liste des coefficients significatifs est initialement vide, tandis que la liste de coefficients insignifiants contient les racines de chaque arbre (coefficients de la bande basse) et la liste d'ensembles insignifiants contient l'ensemble des descendants de chaque arbre. Cette partition initiale est segmentée récursivement au moyen de deux règles. Si un ensemble de descendants d'un nœud est significatif, il est séparé en quatre coefficients fils directs de ce nœud, et l'ensemble des autres descendants. Les fils directs sont ajoutés à la LCN ou la LCS en fonction de leur signifiante. Si au moins un élément de l'ensemble des autres descendants est significatif, cet ensemble est séparé en quatre ensembles insignifiants ajoutés à la LEN. Le fait de traiter

les coefficients par groupes de quatre permet d'effectuer un codage entropique efficace par la suite. Comme dans EZW, la passe de raffinement consiste à coder progressivement les bits de poids faibles des coefficients significatifs [1] [2].

### ***4.3 SPECK: Set-Partitioning Embedded Block Coder***

Offrant des performances comparables à SPIHT, l'algorithme SPECK [56] exploite des structures d'ensembles de coefficients non significatifs en blocs plutôt qu'en arbres. Ces structures de blocs permettent de s'affranchir efficacement de la non-stationnarité des coefficients en adaptant localement la statistique utilisée pour le codage.

Les coefficients sont initialement séparés en deux ensembles, l'un noté  $S$  contenant les coefficients de basses fréquences et l'autre, noté  $I$  contenant les autres coefficients. De la même manière que dans SPIHT, deux listes sont tenues à jour, pour représenter les coefficients significatifs (LCS) et les ensembles de coefficients non significatifs (LEN). La liste d'ensembles non significatifs contient des blocs de coefficients de taille variable, y compris les coefficients isolés vus comme des blocs de  $1 \times 1$  (stockés dans la LCN dans SPIHT). Cette liste triée va des blocs de plus petites tailles aux blocs de plus grandes tailles. Lors du déroulement de l'algorithme, un test de signifiante est réalisé sur chaque ensemble de la LEN à tour de rôle. Si l'ensemble est signifiant et non réduit à un seul coefficient, il est retiré de la liste et partitionné récursivement en quatre sous blocs sur lesquels ce test est effectué à nouveau. Si le bloc est réduit à un seul coefficient significatif, celui-ci est ajouté à la LCS. Dans tous les autres cas, l'ensemble est laissé ou ajouté dans la LEN. Les autres coefficients appartenant à  $I$  sont traités ensuite. Si cet ensemble est signifiant, il est séparé en trois blocs de coefficients correspondant aux sous-bandes de plus basses fréquences, et en un nouvel ensemble  $I$  contenant le reste des coefficients. Ces trois nouveaux blocs sont traités comme précédemment. Ce processus de séparation de l'ensemble  $I$  est répété jusqu'à ce qu'il soit insignifiant. Le codage des bits de raffinement est par ailleurs identique à SPIHT.

Cet algorithme regroupe plusieurs idées développées précédemment. Tout d'abord, le partitionnement adaptatif en quad-tree, développée dans [20] et réalisé lors de la séparation de  $S$ , permet de repérer rapidement les régions hautement énergétiques et de les coder avec un nombre minimum d'information de signifiante. La séparation récursive en sous-bandes réalisée sur l'ensemble  $I$  et initialement introduite dans [21] permet d'exploiter la structure hiérarchique en s'intéressant d'abord aux sous-bandes de plus haute

énergie a priori. Combinées avec le codage du partitionnement proposé dans SPIHT, ces techniques donnent un codeur par blocs offrant des résultats similaires [1] [2].

## 5. Algorithme de codage SPIHT

L'algorithme de codage SPIHT est parmi les algorithmes les plus connus et les plus utilisés dans le domaine de la compression. Il a été proposé initialement par Saïd et Pearlman en 1996 [19] pour le codage des coefficients d'ondelettes. Il a été ensuite utilisé pour la compression d'autres types de données telles que les signaux électrocardiogrammes (ECG) [22],[23] et les signaux vidéos [24]. Cet algorithme peut fonctionner en mode avec ou sans pertes.

Les principes de base de SPIHT sont les suivants [2] [19] [25] [26]:

- un rangement partiel par amplitude des coefficients d'ondelettes résultant de la quantification par approximations successives,
- un partitionnement dans des arbres hiérarchiques à chaque seuil appliqué. Les arbres sont triés sur la base de leur signification en deux catégories d'arbres et un ordonnancement de la transmission des bits de raffinement (l'amplitude de chaque coefficient significatif est progressivement raffinée).

Les descriptions complètes des deux algorithmes de codage et décodage SPIHT sont données ci-dessous. Commençons par définir les différents ensembles utilisés par cet algorithme :

- $O(i, j)$ : Ensemble des coordonnées de tous les enfants du nœud  $(i, j)$ .
- $D(i, j)$ : Ensemble des coordonnées de tous les descendants du nœud  $(i, j)$  (type A d'arbres de zéros)
- $L(i, j) = D(i, j) - O(i, j)$  (type B d'arbre de zéros).

Les règles de partitions sont les suivantes :

- a) La partition initiale est formée des ensembles  $\{(i, j)\}$  et  $D(i, j)$ , pour tous  $(i, j) \in LL_n$  qui ont un descendant. Dans chaque groupe de coefficients  $2 \times 2$  dans la dernière sous bande  $LL_n$ , un des coefficients n'a pas de descendances.

- b) Si  $D(i, j)$  est significatif alors il est découpé en  $L(i, j)$  plus 4 ensembles  $D(l, m)$  d'un seul élément avec  $(l, m) \in O(i, j)$
- c) Si  $L(i, j)$  est significatif alors il est partitionné en 4 sous-ensembles  $D(l, m)$  avec  $(l, m) \in O(i, j)$ .

### 5.1 Méthode de codage

Pour réaliser pratiquement un codage emboité, l'algorithme SPIHT stocke l'information significative dans 3 listes ordonnées :

1. La liste des coefficients significatifs (*LCS*),
2. La liste des coefficients non significatifs (*LCN*),
3. La liste des ensembles non significatifs (*LEN*).

Dans chaque liste, l'entrée, de coordonnées  $(i, j)$ , représente dans *LCS* et *LCN* un coefficient individuel et dans *LEN* elle représente soit l'ensemble  $D(i, j)$  soit l'ensemble  $L(i, j)$ . Pendant la passe de signification, les coefficients dans *LCN*, qui étaient non significatifs dans la passe précédente, sont de nouveau testés. Ceux qui deviennent significatifs sont placés dans *LCS*. Similairement, les ensembles de *LEN* sont évalués selon leur ordre d'entrée. Si un ensemble est trouvé significatif il est supprimé de cette liste puis partitionné. Les nouveaux ensembles, constitués de plus d'un élément, sont ajoutés à la fin de *LEN* avec le type (*A* ou *B*). Tandis que les simples coefficients sont ajoutés à la fin de *LCS* ou de *LCN* suivant leur signification. La liste *LCS* contient les coordonnées des coefficients qui seront traités dans la prochaine passe de raffinement.

L'opérateur de signification  $\sigma_{T_n}$  qui évalue la signification d'un sous-ensemble  $E$  pour un seuil donné  $T_n$  est donné par :

$$\sigma_{T_n}(E) = \begin{cases} 1 & \text{si } \exists w \in E : |w| \geq T_n \\ 0 & \text{si } \exists w \in E : |w| < T_n \end{cases} \quad (2.21)$$

tel que  $w$  représente un coefficient d'ondelette.

## Algorithme de codage SPIHT [2]

### 1. Initialisation :

Sortie  $n = \lceil \log_2(|w_{\max}|) \rceil \Leftrightarrow T_n = 2^n$ ,  $LCS = \emptyset$ ,  $LCN = \{(i, j) \in LL\}$ . LEN contient les mêmes coefficients que LCN excepté ceux qui n'ont pas de descendants.

### 2. Passe de signification :

2.1. Pour chaque  $(i, j) \in LCN$  faire :

2.1.1 Sortie  $\sigma_{T_n}(i, j)$

2.1.2 Si  $\sigma_{T_n}(i, j) = 1$  alors mettre  $(i, j)$  dans LCS et coder le signe de  $w(i, j)$

2.2 Pour chaque  $(i, j) \in LEN$  faire :

2.2.1 Si l'entrée est de type A

a. Sortie  $\sigma_{T_n}(D(i, j))$

b. si  $\sigma_{T_n}(D(i, j)) = 1$  alors

- Pour chaque  $(l, m) \in O(i, j)$  faire :

- Sortie  $\sigma_{T_n}(l, m)$

- si  $\sigma_{T_n}(l, m) = 1$  alors mettre  $(l, m)$  dans LCS et coder le signe de  $w(l, m)$

- si  $\sigma_{T_n}(l, m) = 0$  alors mettre  $(l, m)$  à la fin de LCN.

- Si  $L(i, j) \neq \emptyset$  alors mettre  $(l, m)$  à la fin de LEN comme une entrée de type B.

2.2.2. Si l'entrée est de type B

a. Sortie  $\sigma_{T_n}(L(i, j))$

b. si  $\sigma_{T_n}(L(i, j)) = 1$  alors

- mettre  $(l, m)$  à la fin de LEN comme une entrée de type A

- supprimer  $(i, j)$  de LEN

### 3. Passe de raffinement :

Pour chaque coefficient  $(i, j) \in LCS$  à l'exception de ceux incluse dans la même

passé de signification. Sortie le  $n$ -ème bit significatif de  $|w(i, j)|$

#### 4. Modification du pas de quantification :

$T_n = T_n / 2$  et aller à l'étape (2).

### 5.2 Méthode de décodage

Pour obtenir, l'algorithme de décodage, il suffit simplement de remplacer le mot *Sortie* par *Entrée* dans l'algorithme précédent. Donc, les deux algorithmes ont la même complexité algorithmique.

De plus, le décodeur exécute une tâche supplémentaire en modifiant l'image reconstruite. Pour un seuil  $T_n$  donné, quand un coefficient est déplacé dans la *LCS*, il est évident que  $T_n \leq w(i, j) < 2 \times T_n = 2^{n+1}$ . A cet effet, l'erreur produite par rapport à ce coefficient, compte tenu que les bits non-décodés sont mis à zéro, est  $0 \leq e(i, j) < T_n$ . Une solution relativement simple permettant de réduire la valeur maximale de cette erreur est d'ajouter au coefficient reconstruit, la valeur médiane de l'intervalle des variations de l'erreur  $e(i, j)$  en l'occurrence  $T_n/2$ . Ainsi, le décodeur utilise cette information plus le bit de signe juste après l'insertion dans la *LCS* (signe de  $\llcorner w(i, j)$ ) pour mettre  $\llcorner w(i, j) = \pm 1.5 \times T_n$ .

De manière identique, pendant la passe de raffinement le décodeur ajoute ou soustrait  $T_n/2$  à  $\llcorner w(i, j)$  quand on reçoit les bits de la représentation binaire de  $|w(i, j)|$ . De cette manière, la distorsion baisse simultanément pendant les 2 passes.

Enfin, on notera que l'algorithme SPIHT produit directement des symboles binaires. Ainsi, un codeur arithmétique n'est pas nécessaire même s'il est souvent implanté pour améliorer les performances du codage.

## 6. Les images en couleurs

A l'origine les images sont capturées à l'aide de capteurs CCD (Charge Coupled Device ou Dispositif à Transfert de Charge). S'il s'agit d'images en couleurs chaque capteur CCD est en réalité un triplet de capteurs. Chacun réagit à une couleur ou ce que l'on appelle une longueur d'onde relative au rouge, au vert ou au bleu. Les caméras qui convertissent donc la lumière reçue en un triplet de 3 valeurs (3 composantes) pour chaque pixel de l'image. Le codage couleur est RVB (Rouge, Vert et Bleu) ce qui permet par combinaison de reconstituer toutes les couleurs.

Le codage RVB et aussi parfois noté RGB (Red, Green, Blue). Le codage RVB des couleurs est aussi le codage utilisé par la télévision ou les écrans d'ordinateur pour reproduire les couleurs. En effet, d'après les principes de la colorimétrie Il faut 3 composantes indépendantes pour pouvoir avoir un espace de couleur complet.

Ces trois composantes RVB présentent une redondance entre elles correspondant au niveau de gris que l'on note parfois Y. C'est pourquoi la vidéo analogique couleur est souvent transportée dans un autre mode de représentation appelé YUV ou encore YCrCb. Ou Y représente la luminance et UV les composantes Rouge et Bleu sans la luminance et on les appelle alors les chrominances. Pour passer d'un espace RVB à l'espace YUV et inversement nous disposons d'expressions mathématiques assez simples. Il s'agit tout simplement de produits matriciels puisqu'il s'agit d'un changement de bases entre des espaces vectoriels de dimension 3.

$$Y = 0.257R + 0.504G + 0.098B + 16$$

$$Cr = V = 0.439R - 0.368G - 0.071B + 128$$

$$Cb = U = -0.148R - 0.291G + 0.439B + 128$$

(2.22)

Et inversement nous avons :

$$\begin{aligned}
 B &= 1.164(Y - 16) + 2.018(Cb - 128) \\
 G &= 1.164(Y - 16) - 0.813(Cr - 128) - 0.391(Cb - 128) \\
 R &= 1.164(Y - 16) + 1.596(Cr - 128)
 \end{aligned}$$

(2.23)

## 7. Résultats de compression

L'évaluation des résultats obtenus par la méthode de SPIHT sur plusieurs images de différents types a été effectuée selon deux procédés :

- En fonction du PSNR par rapport au débit
- En fonction de la qualité de l'image obtenue.

Chacune des images tests est compressée avec plusieurs débits différents allant de 0.1 jusqu'à 0.7 *bpp* en mesurant le PSNR obtenu. Pour la décomposition en ondelettes nous avons opté sur cinq (05) niveaux de décomposition avec les filtres d'ondelettes bi-orthogonaux 9/7 avec extension symétrique. Les valeurs du *PSNR* obtenus pour les images de Pepper (ou légumes), *Barbara*, *Boat (ou Bateau)* et *Baboon* sont présentées respectivement dans le Tableau 2.1.

Nous nous sommes contentés de ces valeurs de compression (des débits de 0.1 à 0.7 *bpp*) car ce sont les valeurs que l'on retrouve généralement dans la littérature scientifique spécialisée et ce sont les valeurs que l'on souhaite obtenir avec des PSNR améliorés. D'autre part, les images choisies pour nos tests sont des images standards utilisées par toute la communauté scientifique travaillant dans ce domaine. De même, ce sont des images avec des caractéristiques relativement différentes nous permettant ainsi d'évaluer les performances de cet algorithme sur un large éventail d'images différentes.

Enfin, notons que pour la DWT, ou transformée d'ondelettes discrètes, nous avons opté pour cinq niveaux de décomposition qui sont largement suffisants et nous ne pouvons pas espérer améliorer la qualité de la compression avec des niveaux supérieurs. Quand aux filtres numériques RIF utilisés dans ce cas ils ont prouvés leurs validité dans ce cas de figures. A savoir l'implémentation des ondelettes discrètes par filtres RIF et décimation.

Notons également que pour des images en niveaux de gris l'algorithme SPIHT est appliqué uniquement sur une seule composante à savoir la composante luminance ou Y. pour des images en couleurs la méthode de compression SPIHT est appliquée simultanément sur les trois composantes RVB ou bien encore Y, Cr et Cb.

Débit ( <i>bpp</i> )	Image Pepper	Image Barbara	Image Boat	Image Baboon
0.1	23.56	24.12	20.30	21.15
0.2	25.18	26.24	22.92	22.60
0.3	26.90	27.85	26.42	23.62
0.4	27.70	29.10	27.02	24.20
0.5	28.85	30.85	28.91	24.90
0.6	29.70	31.55	30.15	25.20
0.7	30.20	32.20	31.60	25.50

Tableau 2.1 Les *PSNR* en fonction des débits obtenus par SPIHT

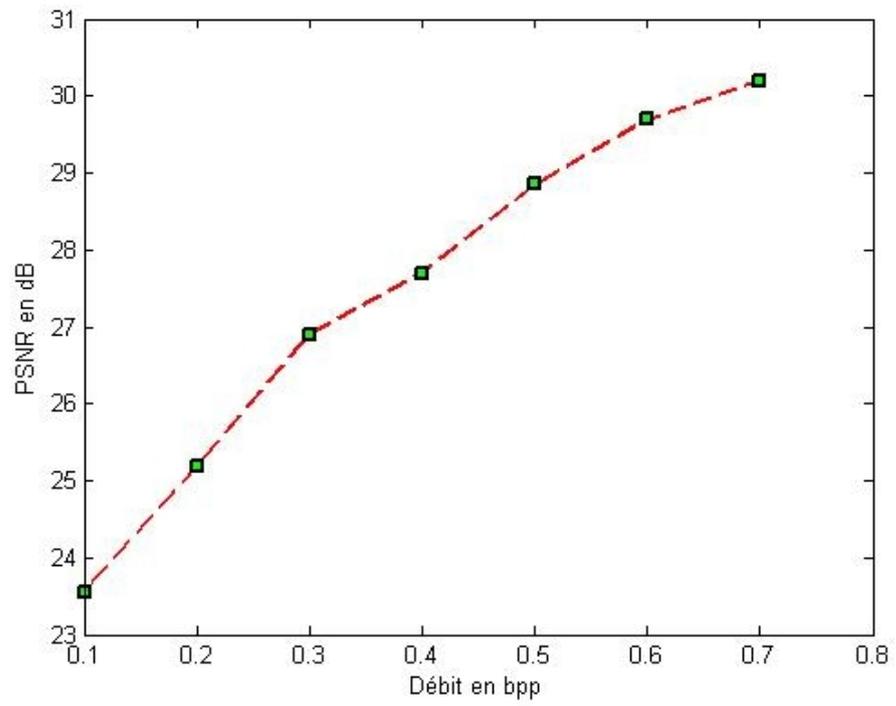


Figure 2.6a : Courbe PSNR en fonction du débit obtenu par SPIHT pour l'image Pepper

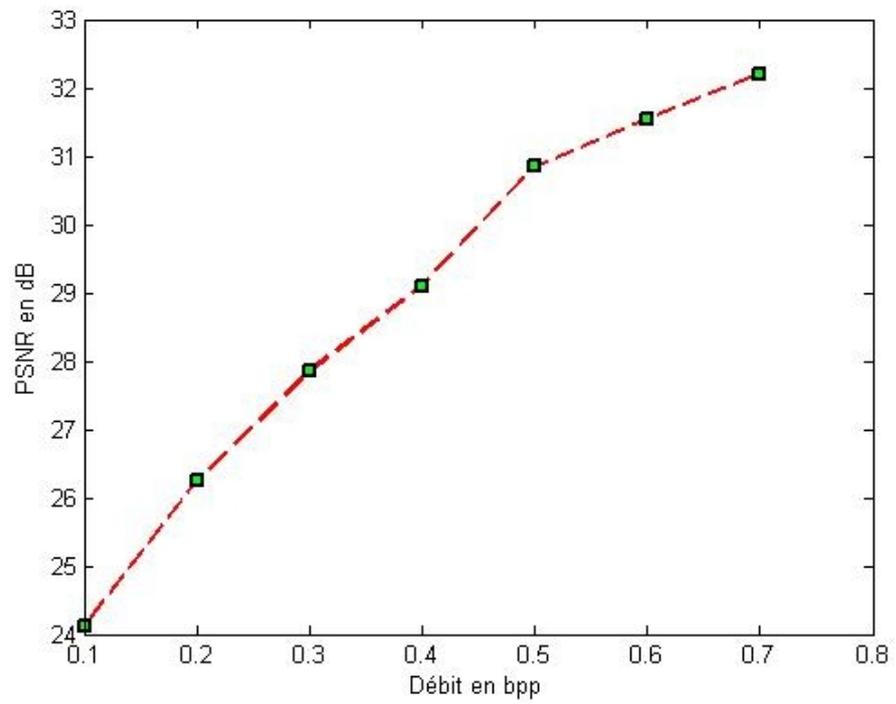


Figure 2.6b : Courbe PSNR en fonction du débit obtenu par SPIHT pour l'image Barbara

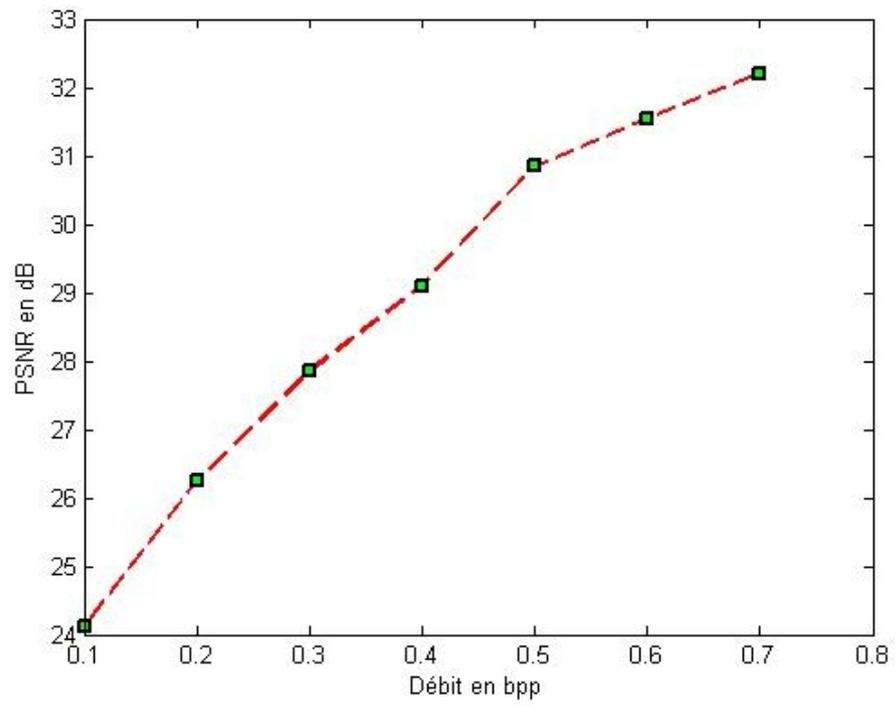


Figure 2.6c : Courbe PSNR en fonction du débit obtenu par SPIHT pour l'image Boat

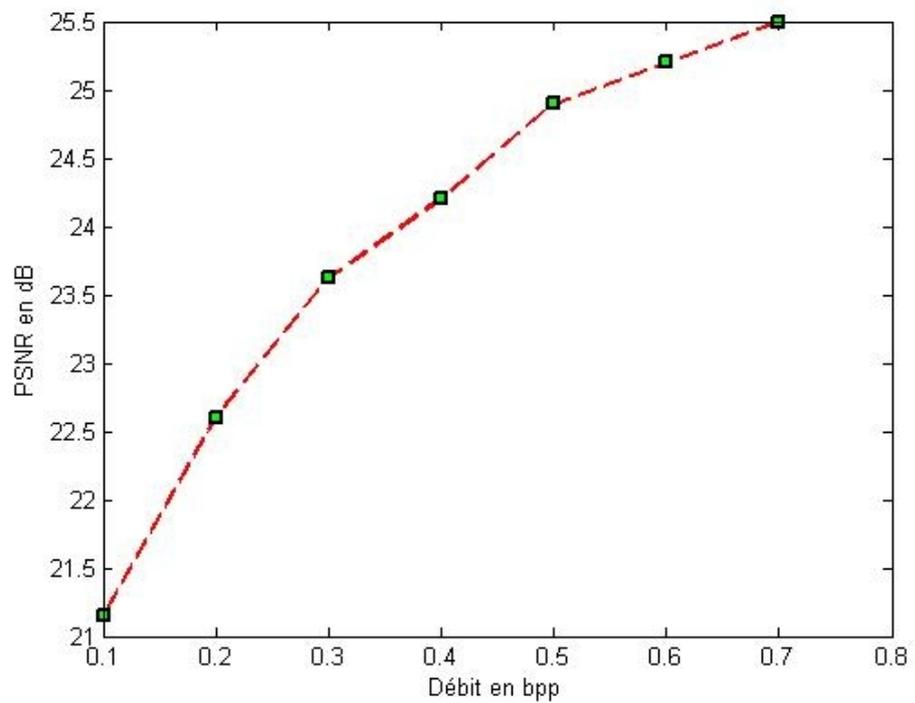


Figure 2.6d : Courbe PSNR en fonction du débit obtenu par SPIHT pour l'image Baboon



(a) Image originale



(b)  $PSNR = 25.18$  à  $0.2$  *bpp*



(b)  $PSNR = 26.90$  à  $0.3$

Figure 2.7. Résultats de compression de l'image Pepper par SPIHT



(a) Image originale



(b)  $PSNR = 26.13$  à  $0.2 \text{ bpp}$



(c)  $PSNR = 27.62$  à  $0.3 \text{ bpp}$

Figure 2.8. Résultats de compression de l'image *Barbara* par SPIHT



(b) Image originale

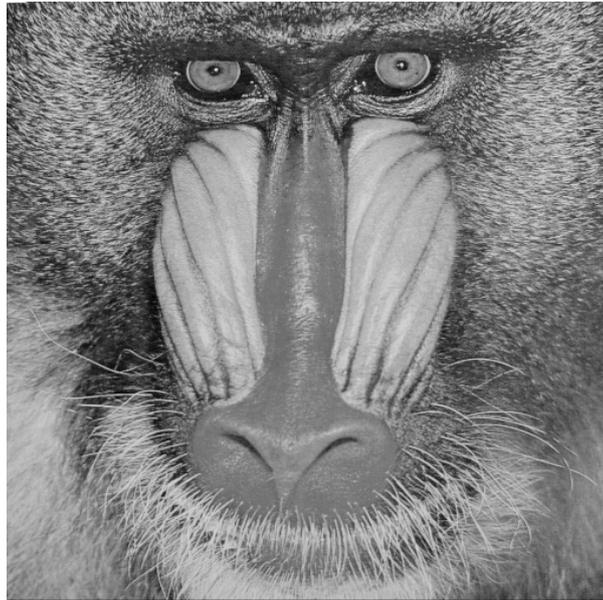


(b)  $PSNR = 26.97$  à  $0.4$  *bpp*



(c)  $PSNR=20.20$  à  $0.1$  *bpp*

Figure 2.9. Résultats de décompression de l'image *Boat* par SPIHT



(a) Image originale



(b)  $PSNR = 22.60$  à  $0.2 \text{ bpp}$



(c)  $PSNR = 23.62$  à  $0.3 \text{ bpp}$

Figure 2.10. Résultats de décompression de l'image *Baboon* par SPIHT

Débit ( <i>bpp</i> )	Image	Image	Image	Image
----------------------	-------	-------	-------	-------

	Pepper	Barbara	Boat	Baboon
0.1	23.77	24.32	20.65	21.34
0.2	24.98	26.54	23.05	22.75
0.3	27.02	28.05	26.60	23.88
0.4	27.90	29.04	27.20	24.45
0.5	29.15	31.06	29.04	25.04
0.6	30.02	31.76	30.20	25.65
0.7	30.55	32.40	31.84	25.77

Tableau 2.2 Les *PSNR* en fonction des débits obtenus par SPIHT pour des images en couleur

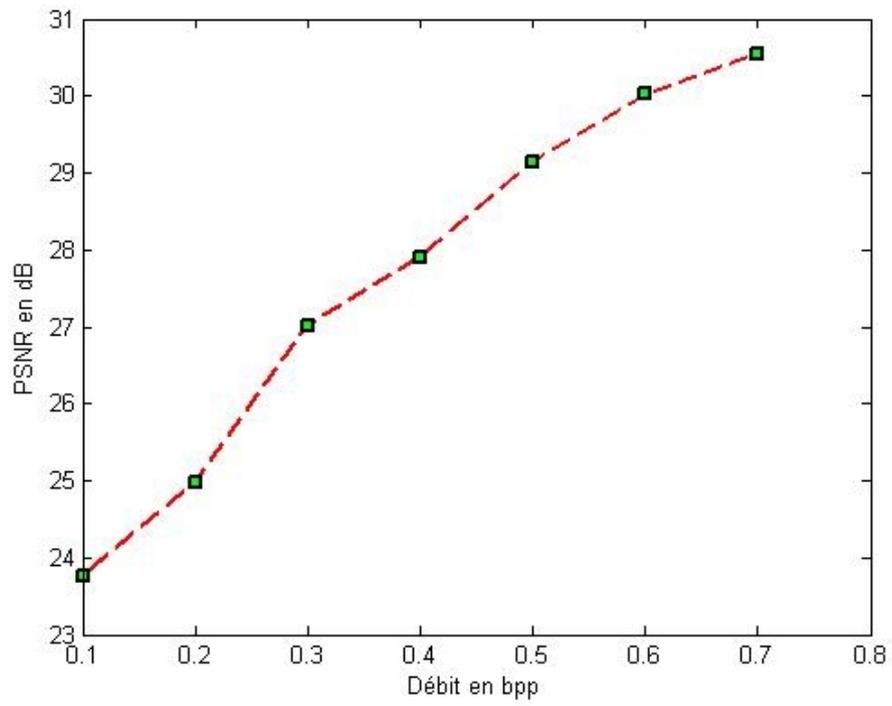


Figure 2.11a : Courbe PSNR en fonction du débit obtenu par SPIHT pour l'image Pepper couleur

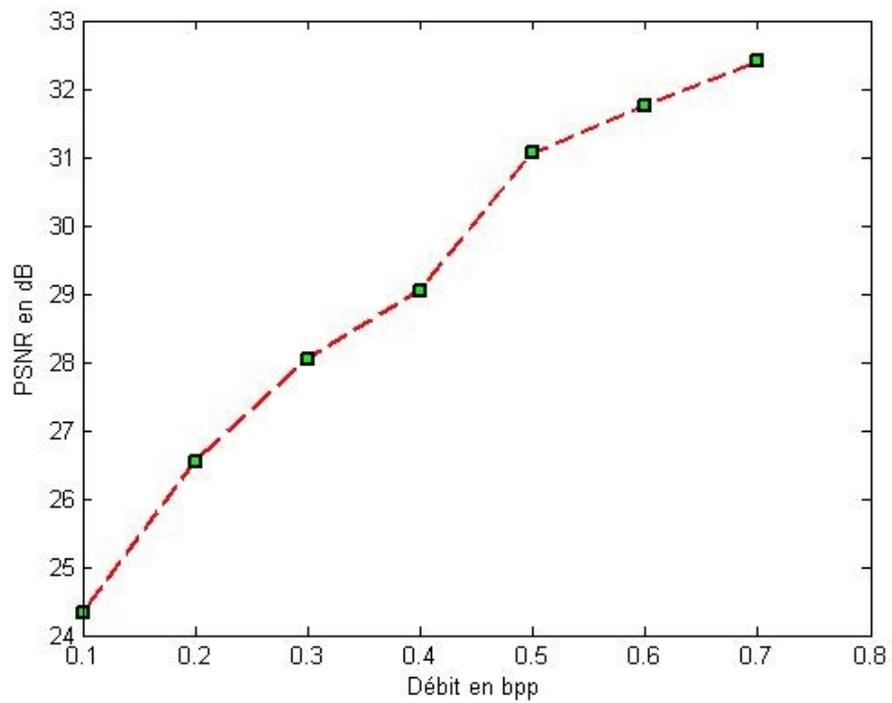


Figure 2.11b : Courbe PSNR en fonction du débit obtenu par SPIHT pour l'image Barbara couleur

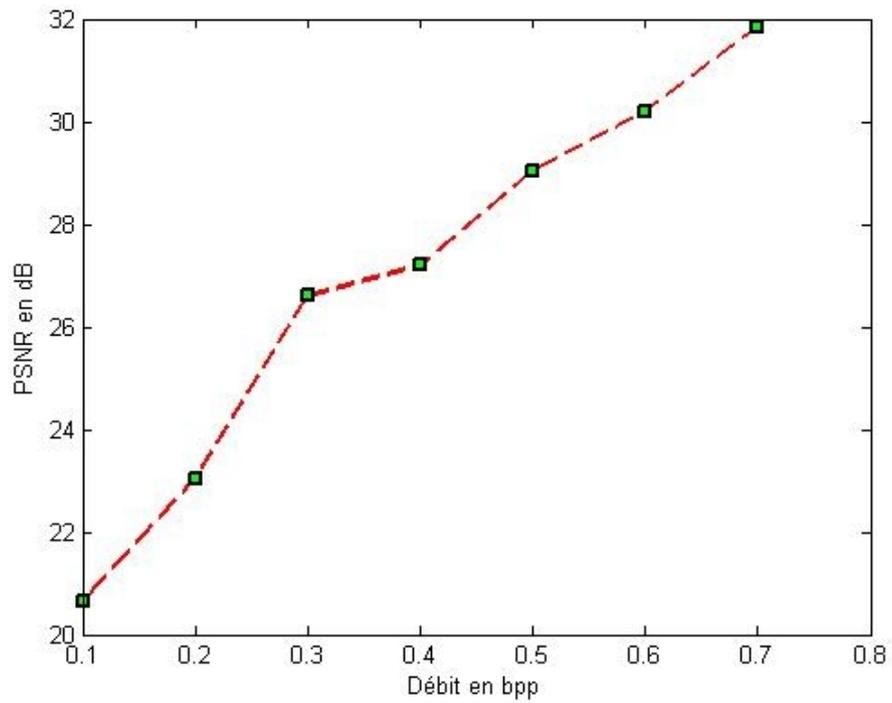


Figure 2.11c : Courbe PSNR en fonction du débit obtenu par SPIHT pour l'image Boat couleur

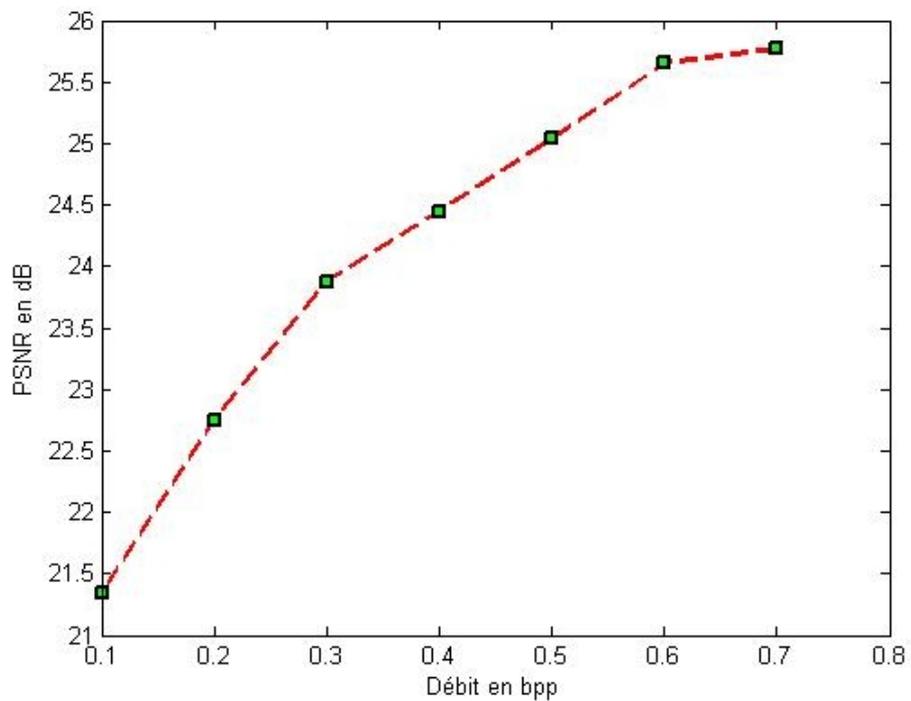


Figure 2.11d : Courbe PSNR en fonction du débit obtenu par SPIHT pour l'image Baboon couleur



(a) Image en couleur originale



Figure 2.12 Résultats de décompression de l'image *Pepper en couleur* par SPIHT



(a) Image en couleur originale



Figure 2.13 Résultats de décompression de l'image *Boat en couleur* par SPIHT

Les résultats de compression des images utilisées dans ces tests prouvent bien les bonnes performances de cet algorithme. Evidemment, on remarque d'importantes disparités du PSNR pour les mêmes débits entre les différentes images. Ceci est dû essentiellement aux caractéristiques de chacune de ces images. En effet, si nous prenons le cas de l'image Barbara nous remarquons de forts PSNR par rapport aux autres images et

plus particulièrement la troisième image celle du bateau ou "boat" pour les mêmes débits. L'image Barbara présente de larges zones homogènes avec peu de contours et de détails aigus. C'est ce que l'on peut appeler une image basse fréquences. Au contraire la troisième image présente un peu moins de zones homogènes et beaucoup de contours et de discontinuités. C'est le type même d'images avec des caractéristiques plutôt hautes fréquences ou de bandes relativement larges. Ce type de spécificités influe beaucoup sur la qualité de la compression surtout quand il s'agit de techniques basées sur les transformations d'une manière générale.

Les tests que nous avons effectués sur les images comprenaient des images en niveaux de gris et les mêmes images en couleur. Les résultats en termes de PSNR en fonction du débit pour ces deux types d'images ne présentent pas des différences importantes. Ceci prouve que l'information contenu dans la composante luminance est importante. En effet, cette composante que l'on note habituelle Y est redondante dans les trois composantes couleurs R, V et B des images en couleur.

Notons que si nous avons considéré des débits (en bpp) égal ou supérieur à la majorité des PSNR calculés, aussi bien pour les images en niveaux de gris que pour les images en couleur, seront nettement supérieurs à 30dB. Des images dont les PSNR sont compris entre 30 dB et 40dB sont jugées généralement comme des images de bonnes qualités

## **8. Conclusion**

Dans ce chapitre nous nous sommes contentés d'implémenter l'algorithme de SPIHT et d'évaluer ses performances sur quelques exemples d'images tests que l'on rencontre souvent dans les travaux relatifs au traitement des images d'une manière générale et la compression d'une manière particulière.

Les résultats de compression basée sur l'algorithme de SPIHT prouvent son efficacité. Cependant, cette efficacité est tributaire des caractéristiques des images

utilisées. De même, cette efficacité dépend énormément du type d'ondelette utilisée et du nombre de décomposition effectué. Les travaux ont montré qu'un nombre de décomposition égal à cinq est largement suffisant pour la majorité des cas et que nous ne pouvons pas espérer améliorer davantage en choisissant un nombre plus élevé.

# CHAPITRE 3

## *Les réseaux de capteurs sans fil (WSN)*

### **1. Introduction**

Les réseaux de capteurs sans fil (Wireless Sensor Networks) ont évolué et leurs applications se sont multipliées dans divers domaines suite aux progrès technologiques essentiellement des réseaux de communication et de l'électronique embarquées. Ceci, a contribué énormément au développement de dispositifs peu coûteux, de très faibles tailles, de faible puissance, pouvant communiquer entre eux, appelés capteurs sans fil. Ils

travaillent en réseau, communiquent en radiofréquence (RF) et peuvent être dispersés, souvent d'une manière aléatoire, sur de vastes zones souvent difficilement accessibles. Ces capteurs possèdent généralement une autonomie en énergie électrique de durée relativement élevée. Il s'agit de plusieurs mois voir même de plusieurs années.

Nous allons nous intéresser dans ce premier chapitre de notre mémoire de magistère à la présentation et la description des réseaux de capteurs sans fil. Nous allons surtout mettre le point sur les caractéristiques, les domaines d'applications et les protocoles de communications de ces réseaux. Dans ce même chapitre nous allons nous intéresser plus particulièrement aux réseaux de capteurs d'images sans fil. En effet, les images numériques de par leurs natures sont très encombrantes en quantité de données numériques. Ceci nous a toujours obligés à les coder et à les compresser dans le but de réduire la quantité de données de ces images tout en maintenant une certaine qualité de ces images. Néanmoins, les méthodes de compression connues et utilisés sont basées essentiellement des algorithmes très couteux en termes de calcul. Dans les systèmes de traitement numérique classique (DSP et autres) ceci ne pose aucun problème compte tenu de la puissance de calcul de leurs processeurs et aussi de leurs alimentations électriques à partir du secteur. Seulement, pour le cas des réseaux de capteurs d'images sans fil ce calcul prohibitif pose plusieurs problèmes très sérieux. D'abord les processeurs ou unités de calcul dans les nœuds de ces capteurs ne sont pas suffisamment puissants. D'autre part, ces nœuds de capteurs sont alimentés par des batteries qui doivent persister pendant une durée relativement élevée. Trop de calcul va certainement consommer la charge de ces batteries très rapidement ce qui est un handicap majeur. Il est donc nécessaire de revoir les méthodes de compression que nous devons implémenter dans ce type de capteurs en termes de puissance électrique consommée qui est directement reliée aux taux de calcul de ces algorithmes.

## **2. Les réseaux de capteurs sans fil**

Un réseau de capteurs sans fil (WSN) est un système distribué de grande échelle mettant en communication un grand nombre d'entités autonomes communément appelées "capteurs sans fil". Ces capteurs forment donc les nœuds (ou motes) du réseau. Dans un mode d'application classique, plusieurs nœuds capteurs sont déployés dans un certain environnement pour mesurer certains phénomènes physiques et faire remonter les

informations collectées à une station de base, nommée le nœud “puits” (ou en anglais « sink ») et qui est considérée comme une porte d’entrée vers le monde extérieur qui fait l’interface entre le réseau de capteurs et l’utilisateur des données). Dans le cas le plus simple, les capteurs seront dans le voisinage direct du puits (un réseau de type étoile à un saut). Cependant, dans le cas d’un réseau à grande échelle, les capteurs ne sont pas tous dans le voisinage du puits et les messages seront acheminés du nœud source vers le puits en transitant par plusieurs nœuds, selon un mode de communication à un saut (nœud-puits) ou multi-sauts comme l’illustre la figure 1.1 [27].

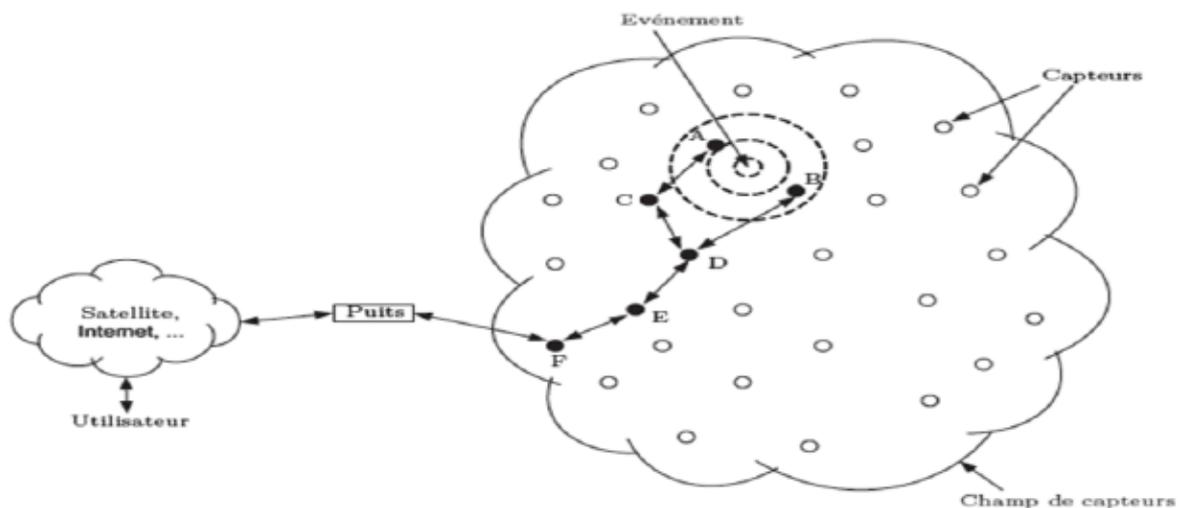


Figure 3.1 : Schéma d’un réseau de capteurs sans fil

### 3. Domaines d’applications des réseaux de capteurs sans fil

La miniaturisation et le faible coût des micro-capteurs, et l’évolution des supports de communication sans fil, ont élargi le champ d’application des réseaux de capteurs dans plusieurs domaines et même aussi d’étendre le domaine des applications existantes. Plusieurs applications envisagées dans les réseaux de capteurs font toujours l’objet d’une recherche et d’un développement actif universitaires ou industriels.

- **Application militaire**

Les premières applications potentielles des réseaux de capteurs ont concerné le domaine militaire. L’idée était de déployer un réseau de capteurs nanoscopiques (donc invisibles) sur des champs de bataille ou des zones ennemies pour surveiller le mouvement des troupes. En outre, un réseau de capteurs peut être déployé dans un endroit stratégique ou hostile, afin de surveiller les mouvements des forces ennemies, ou analyser

le terrain avant d'y envoyer des troupes (détection des armes chimiques, biologiques ou radiations). Et jusqu'aujourd'hui, les applications militaires sont les plus représentatives des applications trouvées actuellement dans le domaine des réseaux de capteurs sans fil.

- **Applications environnementales**

Le contrôle des paramètres environnementaux par les réseaux de capteurs peut donner naissance à plusieurs applications. Par exemple, le déploiement des thermo-capteurs dans une forêt peut aider à détecter un éventuel début de feu et par suite faciliter la lutte contre les feux de forêt avant leur propagation. Le déploiement des capteurs chimiques dans les milieux urbains peut aider à détecter la pollution et analyser la qualité d'air. De même leur déploiement dans les sites industriels empêche les risques industriels tels que la fuite de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.). D'autres applications environnementales sont destinées à la surveillance de certains phénomènes climatiques et sismiques afin de détecter ou de prévoir certaines catastrophes naturelles telles que l'éruption des volcans et les inondations.

- **Application à la surveillance**

L'application des réseaux de capteurs dans le domaine de la sécurité peut diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et des êtres humains. Ainsi, l'intégration des capteurs dans de grandes structures telles que les ponts ou les bâtiments aidera à détecter les fissures et les altérations dans la structure suite à un séisme ou au vieillissement de la structure.

- **Applications médicales**

Avec le développement de la technologie des micros-capteurs on est arrivée à une étape de la médecine à distance ou « télémédecine », les réseaux de capteurs peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humain grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau (surveillance de la glycémie, détection de cancers, ..). Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, battements du cœur, ... à l'aide des capteurs ayant chacun une tâche bien

particulière.

## 4. Constitution d'un nœud capteur sans fil

### 4.1. Architecture d'un nœud capteur sans fil

Un nœud capteur sans fil est un petit dispositif électronique intelligent, capable de prélever des mesures d'un phénomène naturel (par exemple température, lumière et pression ...etc.) de les traiter par les circuits de traitement et de les transmettre par un émetteur radio au centre de contrôle ou à l'utilisateur de ces informations via une station de base. Le nœud capteur sans fil est considéré comme un véritable système embarqué issue de la révolution technologique des instruments de mesure et des systèmes micro-électronique, et aussi du développement des systèmes de communication sans fil. Un nœud capteur est composé de plusieurs éléments ou modules correspondant chacun à une tâche particulière : captage, traitement, ou transmission de données. Il comprend également une source d'énergie qui est une batterie.

Un capteur sans fil est composé principalement des quatre unités élémentaires [28] :

- **Unité de Captage** : Ce module est l'unité qui contient le ou les capteurs embarqués sur le nœud, on note que le nœud capteur peut embarquer plusieurs capteurs de plusieurs phénomènes (capteur de température, lumière, humidité, ou autres). Le module de captage est suivi toujours par un convertisseur analogique numérique (CAN ou ADC) qui convertit les signaux provenant des capteurs (signaux analogiques) en signaux interprétables par l'unité de traitement (signaux numériques).
- **Unité de traitement** : Elle comprend un processeur associé à une petite unité de stockage, elle fonctionne à l'aide d'un système d'exploitation spécialement conçu pour les micro-capteurs (TinyOS par exemple). Elle exécute les protocoles de communications qui permettent de faire collaborer le nœud avec les autres nœuds du réseau. Elle peut aussi analyser les données captées pour alléger la tâche du nœud puits. Bien que ce ne soit pas obligatoire, il est souhaitable qu'il existe des moyens de reprogrammer facilement les capteurs dans le cas d'un changement dans les exigences de l'application.
- **Unité de Communication** : Elle est le plus souvent constituée d'un transmetteur radio qui fournit au capteur la capacité de communiquer avec les autres au sein

d'un réseau. Elle met en œuvre des protocoles de communication dépendant de la technologie utilisée (par exemple 802.11, 802.15.1, 802.15.4, etc. pour les technologies sans fil), l'unité de transmission radiofréquence comprennent une petite antenne d'émission/réception, des circuits de modulation, démodulation, filtrage et multiplexage, etc. Cette unité est la plus consommatrice d'énergie dans un nœud capteur.

- Unité de puissance** : pour des réseaux de capteurs sans fil autonomes, l'alimentation est une composante cruciale. Un nœud capteur est muni d'une ressource énergétique (une batterie standard de basse tension par exemple une pile AA normale d'environ 2.2 – 2.5 Ah fonctionnant à 1.5 V). Étant donné sa petite taille, cette ressource énergétique est limitée et généralement non-remplaçable. ceci fait souvent de l'énergie la ressource la plus précieuse d'un réseau de capteurs, car elle influe directement sur la durée de vie des micro-capteurs et donc du réseau entier. En plus des quatre unités fondamentales données, Il existe des nœuds capteurs qui sont dotés d'autres unités additionnels : les systèmes de « localisation » tels que GPS (Global Position System) afin d'identifier leur position géographique et un « mobilisateur » pour qu'ils puissent se déplacer. La figure suivante illustre les parties essentielles d'un nœud capteur, et les parties avec des traits discontinues représentent les unités facultatives.

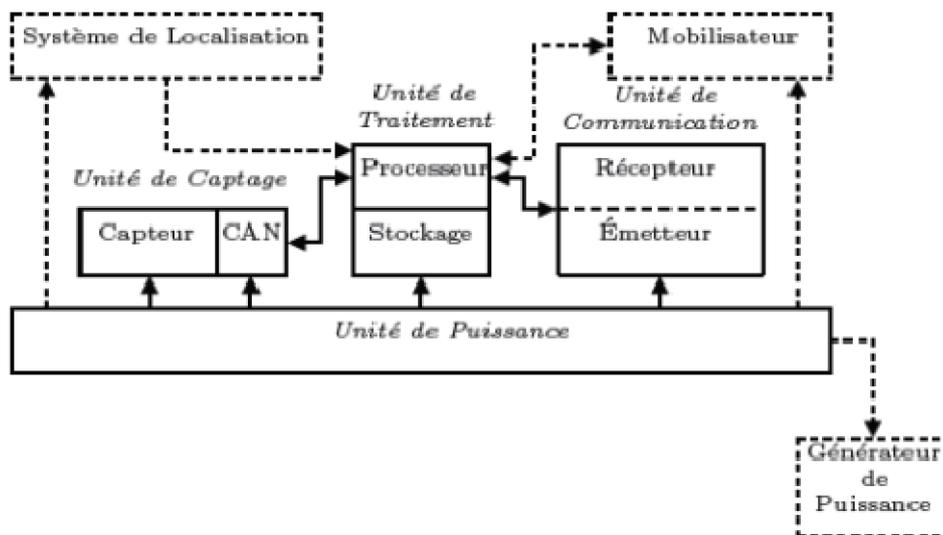


Figure 3.2 : constitution d'un nœud de capteur sans fil

## 4.2. Prototypes de nœuds de capteurs

Le titre de premier prototype de nœuds de capteurs sans fil identifiable dans la bibliographie correspond au module LWIM (Low-power Wireless Integrated Microsensors) développé dans le milieu des années 90 par l'Agence pour les Projets de Recherche Avancée de Défense (DARPA) des Etats-Unis et l'UCLA1. Il s'agissait d'un géophone équipé d'un capteur de transmission radiofréquence et d'un contrôleur PIC. Depuis un peu plus de 10 ans, la technologie des capteurs sans fil a beaucoup évolué. Les modules deviennent de plus en plus petits et les durées de vie prévues augmentent. Aujourd'hui, le marché de nœuds a été ouvert à l'industrie. Le fournisseur le plus connu est **Crossbow Inc**, avec son offre de capteurs **Mica2** et **MicaZ**. (Aujourd'hui, le nœud capteur MicaZ de crossbow devient la référence dans les travaux de recherche dans le domaine des réseaux de capteurs.)

Le tableau 3.1 présente les différents prototypes actuellement disponibles sur le marché et leurs caractéristiques techniques.

Les nœuds de capteurs sont fabriqués avec un mode de conception modulaire. En effet, tous les nœuds du tableau 3.1 sont des cartes intégrées qui regroupent l'unité de communication et l'unité de traitement, et l'unité d'alimentation, tandis que l'unité de captage est conçue comme une carte distincte qui peut être attachée sur l'unité principale. Cela permet de pouvoir réutiliser les mêmes unités pour différentes applications. Par exemple, un nœud Mica2 peut être combiné avec une carte MTS310 qui contient un capteur de température, un capteur lumière, un capteur de son, un capteur de champ magnétique, et un accéléromètre à deux axes. De même, nous pouvons combiner le nœud Mica2 avec une carte MTS420 pour le doter d'un capteur d'humidité et d'un capteur de pression barométrique, et même d'un GPS pour le positionnement géographique [27].

Plate-forme	Fabricant	Unité de Traitement	Unité de Communication	Unité de Captage	Unité de Puissance
MICA2	Crossbow	Atmel ATmega128L (128 Ko de mémoire de programme, 4 Ko RAM) 512 ko mémoire flash pour des données EEPROM 4 Ko (configuration)	CC1000 (radio transcepteur multi-freq. 868/916 - 433 - 315 MHz, 38.4Kbaud)	Connecteur pour carte de capteurs externe	2.7 - 3.3V
MICAZ	Crossbow	Atmel ATmega128L 512 ko mémoire flash pour des données EEPROM 4 Ko (configuration)	Chipcon CC2420 (radio transcepteur 802.15.4, bande ISM de 2400 à 2483.5 MHz, 250 kbps)	Connecteur pour carte de capteurs externe	2.7 - 3.3V
IRIS	Crossbow	Atmel ATmega128L (128 Ko de mémoire de programme, 8 Ko RAM) 512 ko mémoire flash pour des données EEPROM 4 Ko (configuration)	Radio transcepteur 802.15.4 (bande ISM, de 2400 à 2480 MHz, 250 kbps)	Connecteur pour carte de capteurs externe	2.7 - 3.3V
Imote2	Crossbow	Intel PXA271 256 ko mémoire SRAM 32 Mo mémoire SDRAM 32 Mo mémoire flash	TI CC2420 (bande ISM, de 2400 à 2483.5 MHz, 250 kbps)	Connecteur pour carte de capteurs externe	3.2 - 4.5V
Imote Sky	Moteiv (Sentilla)	Texas Instruments MSP430 F1611 (10Ko RAM, 48Ko Flash, 128o stockage d'information)	Chipcon CC2420	Connecteur pour carte de capteurs externe	2.1 - 3.6V
BTnode rev3	ETH	Atmel ATmega128L 64+180 Kbyte RAM EEPROM 4 Ko PIC 18F6720 (20 MHz), Mémoire interne :	Bluetooth, CC1000	Connecteur pour carte de capteurs externe	DC externe 3.8 - 5V ou 2AA
Particle 2/29	TECO	Mémoire interne : 128Ko de mémoire de programme, 4Ko RAM, 1Ko EEPROM, 512 Ko Mémoire flash pour des données	TR1001 (RFM, bande passante 125Ko, bands ISM 868.35 ou 315 MHz)	Connecteur pour carte de capteurs externe	0.9 - 3.3 V

Tableau 3.1 : exemples et caractéristiques de nœuds de capteurs sans fil

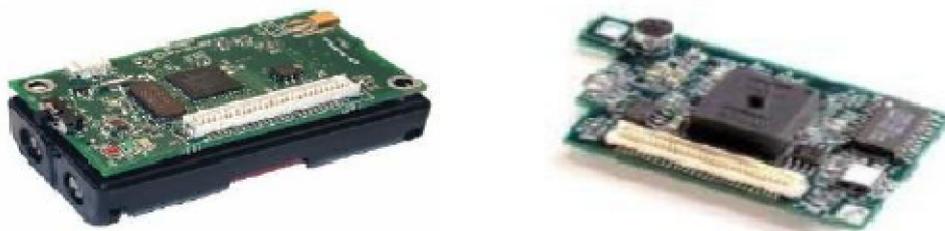


Figure 3.3 : Nœud de capteur Mica2 (à gauche) et carte de captage (à droite) [29]

### 4.3. Plateformes logicielles

Les systèmes d'exploitation embarqués généralement multitâches ont été développés à l'origine pour une plateforme PC et ont ensuite été modifiés ou adaptés à des systèmes embarqués tels que les nœuds capteurs. De nombreux nouveaux produits logiciels sont attendus, y compris dans le domaine de l'open source avec par exemple

TinyOS développé à l'Université de Berkeley ; un système d'exploitation conçu pour les capteurs embarqués sans-fil qui est déjà utilisé (en 2009) par plus de 500 universités et centres de recherche dans le monde. La réalisation de programmes sur cette plateforme s'effectue exclusivement en NesC (dialecte du langage C). Cet OS a notamment pour particularité une taille extrêmement réduite en termes de mémoire (quelques kilo-octets) [30].

## 5. Consommation énergétique d'un nœud capteur

Les nœuds-capteurs sont alimentés principalement par des batteries. Ils doivent donc fonctionner avec un bilan énergétique frugal. En outre, ils doivent le plus souvent avoir une durée de vie de l'ordre de plusieurs mois, voire de quelques années, puisque le remplacement des batteries n'est pas une option envisageable pour des réseaux avec des milliers de nœuds.

Afin de concevoir des solutions efficaces en énergie, il est extrêmement important de faire d'abord une analyse des différents facteurs provoquant la dissipation de l'énergie d'un nœud-capteur. Cette dissipation d'énergie se fait de manière générale selon plusieurs formes [28]:

- **Les collisions** : elles sont la première source de perte d'énergie. Quand deux trames sont émises en même temps et se heurtent, elles deviennent inexploitables et doivent être abandonnées. Les retransmettre par la suite, consomme de l'énergie.
- **L'écoute à vide (idle listening)** : un nœud dans l'état « idle » (état d'écoute non active) est prêt à recevoir un paquet, mais il n'est pas actuellement en train de recevoir quoi que ce soit. Ceci est coûteux et inutile dans le cas des réseaux à faible charge de trafic. Plusieurs types de radios présentent un coût en énergie significatif pour le mode « idle ». Eteindre la radio est une solution, mais le coût de la transition entre les modes consomme également de l'énergie, la fréquence de cette transition doit alors rester raisonnable.
- **L'écoute abusive (overhearing)** : cette situation se présente quand un nœud reçoit des paquets qui ne lui sont pas destinés. Puisque le médium est un environnement commun, lorsqu'un émetteur transmet ses trames de données, tous les nœuds qui se trouvent autour de lui sont obligés d'écouter cette transmission. Cette écoute

passive est nécessaire pour déterminer le moment où le médium de transmission se libère afin de transmettre les trames de données. Le coût de l'écoute abusive peut être un facteur dominant de la perte d'énergie quand la charge de trafic est élevée et la densité des nœuds est grande.

- **La transmission radio** : l'émetteur radio est un des composants les plus gourmands en énergie dans un nœud capteur, et donc que la plupart de l'énergie dissipée dans un nœud concerne la transmission et la réception de données. En effet, l'émetteur possède quatre modes de fonctionnement : transmission, réception, écoute non active ou « idle » et endormi (mode sommeil). Dans la plupart des cas, les consommations d'énergie en mode réception et en écoute non-active sont similaires et elles consomment environ la moitié du mode de transmission. Par contre, la consommation d'énergie en mode endormi est beaucoup plus basse.

## 6. Problématiques des réseaux de capteurs sans fil

Les problèmes posés par les réseaux de capteurs sans fil ont déjà été discutés. Nous résumons ci dessous les plus importants :

- **Energie** : l'énergie est considérée comme une ressource rare dans les applications de réseaux de capteurs sans fil. En effet, les nœuds généralement utilisent de batteries, souvent non rechargeables, Il en résulte que l'efficacité de la gestion de l'emploi de l'énergie disponible est une question souvent vitale pour les réseaux.
- **Ressources limitées des nœuds** : Les nœuds ont donc des ressources très limitées, en comparaison avec les équipements informatiques que nous avons de nos jours (tels que les ordinateurs portables), en termes de mémoire disponible, de capacité et de vitesse de traitement, de débit de transmission ou bande passante.
- **Environnement de communication non contrôlable** : Les réseaux de capteurs souffrent de tous les problèmes de l'usage d'une communication sans fil, tels que des problèmes d'interférences et des problèmes de sécurité (attaques). Les signaux radio émis par les nœuds peuvent être sérieusement endommagés par les interférences présentes dans le milieu.

## 7. Les réseaux de capteurs d'images sans fil

Le développement des micro-caméras et microphones a observé une forte évolution au cours des dernières années. Ces dispositifs deviennent de plus en plus petits et bon marché, et fournissent de plus en plus de performances en termes de rapidité et de qualité d'image. Aujourd'hui, nous trouvons ces micro-caméras embarquées dans pratiquement tous les téléphones cellulaires et les assistants numériques personnels (PDA), sans augmentation significative du coût de l'équipement, de son poids et de sa forme.

Les réseaux de capteurs sans fil n'ont pas été en dehors de ce progrès et aujourd'hui, nous pouvons déjà voir les résultats des dernières avancées de microphones et micro-caméras CMOS, sous la forme de cartes de capteurs compatibles avec des prototypes de noeuds capteurs sans fil, tels que ceux déjà présentés dans la partie prototypes de noeuds capteurs. Cela a permis d'envisager concrètement un nouveau type d'applications utilisant des « *réseaux de capteurs d'images sans fil* ». En fonction des exigences imposées à l'application, et bien évidemment en fonction du type de technologie disponible, les réseaux de capteurs d'images peuvent être de deux types : [31]

- **Réseaux de capteurs d'images fixes** : Des capteurs d'images numériques peuvent prendre des photos qui peuvent être mémorisées en format matriciel ou vectoriel. Ce type de capteur est facile à réaliser et peut être adapté facilement à des dispositifs avec des ressources limitées, tels que les noeuds de capteurs sans fil.
- **Réseaux de capteurs de vidéo** : Des capteurs d'images numériques peuvent aussi envisager de prendre des séquences d'images et de transmettre le flux vidéo vers le puits. Cette application exige des noeuds avec des capacités de calcul, de mémoire et de communication d'un tout autre ordre de grandeur que pour les images fixes. Les séquences d'image doivent être compressées fortement pour satisfaire à la contrainte de bande passante des liaisons sans fil. Ces applications consomment nécessairement une quantité d'énergie bien supérieure à celles utilisant des images fixes. Mais, en raison de la difficulté que comporte la transmission d'un flux vidéo, la plupart des prototypes de capteurs d'images sont dédiés aux images statiques.

Un réseau de capteurs d'images se compose de plusieurs noeuds capteurs d'image (caméras), installées dans divers points d'un champ de capture. Ces capteurs transmettent

l'information capturée (qui est principalement l'image) d'une manière autonome et sans fil à la station de base.

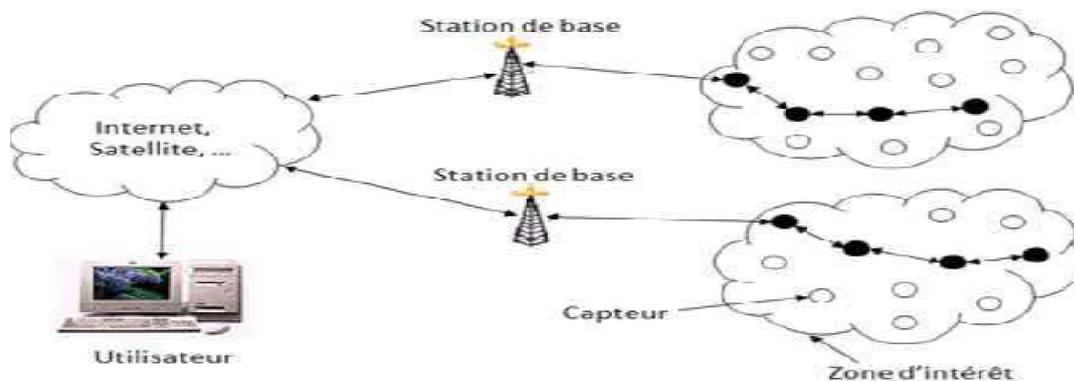


Figure 3.4 : architecture d'un réseau de capteurs d'images sans fil

A son tour la station de base (ou puits) diffuse cette information sur un réseau locale d'entreprise (type Ethernet par exemple), ou à un ou plusieurs administrateurs et utilisateurs du réseau de capture utilisant le réseau mondiale internet, comme le montre la figure 3.4.

## 8. Spécificités des réseaux de capteurs d'images

Les travaux sur des images sont différents des travaux sur des signaux numériques ou analogiques plus simples, comme ceux des premiers réseaux de capteurs. Ces différences sont dues essentiellement à la complexité du signal capturé. En effet, tandis que pour le codage d'un signal simple tel que le niveau de température ou la pression barométrique ou autres, un ou deux octets sont suffisants, le codage d'une image numérique conduit à l'emploi de plusieurs centaines ou milliers d'octets. Cette différence de grandeur a des conséquences sur différents facteurs : capture du signal, besoins en mémoire, traitement du signal et transmission de données. [27]

- **Capture du signal** : La complexité du matériel est multipliée par rapport aux captures de phénomènes simples. En effet, un capteur de caméra CMOS est normalement composé de nombreux capteurs photo-sensibles que capturent les différentes intensités pour chaque pixel. Tandis que pour la capture d'un signal de lumière un seul photo-capteur est suffisant, pour capturer une image nous avons

besoin de beaucoup plus (normalement un par pixel). Cette évidence entraîne avec elle un coût supplémentaire en énergie et en temps de capture.

- **Besoins de mémoire** : Comme nous l'avons dit, tandis que pour le codage d'un signal simple cela sollicite quelques bits d'information, le codage d'une image numérique conduit à l'emploi de plusieurs centaines ou milliers d'octets. En particulier, la quantité de mémoire nécessaire dépend principalement de deux facteurs clés : La résolution de l'image et le format.
- **Traitement du signal** : la compression d'image est l'une des opérations nécessaires effectués sur les images afin de diminuer la quantité de données nécessaires pour la représenter et par conséquent diminuer l'énergie de transmission consommée par l'émetteur radio. Mais, compte tenu des capacités limitées des matériels utilisés dans les réseaux de capteurs, le traitement d'image à la source devient très difficile. Les temps de calcul sont considérablement augmentés et l'énergie investie est parfois plus importante que celle économisée.
- **Transmission d'image** : la transmission d'image est une opération compliquée, à cause de la contrainte d'énergie, et du bas débit du matériel d'émission, et aussi du risque de perte de paquets engendré par le canal de transmission, toutes ces limites auront des effets indésirables sur les images finales reçus.

## **9. Compression d'image dans les réseaux de capteurs sans fil**

### **9.1. Introduction**

Comme le transmetteur radio est l'un de composants électroniques les plus gourmands en énergie dans un nœud de capteurs, il est clair qu'un moyen d'économiser de l'énergie est de réduire la quantité de données à transmettre. La solution est alors la compression de données. En effet, moins il y aura de données à transmettre et moins le tranciever radio consommera d'énergie. Nous allons voir que ce raisonnement n'est pas toujours valable car la compression a un coût d'énergie qui peut être très élevé. Le problème est de disposer d'un algorithme de compression de données qui soit peu gourmand en énergie (moins calculatoire) et qui ait un bon rapport débit-distorsion. Une longue liste d'algorithmes de compression est aujourd'hui disponible, dont plusieurs

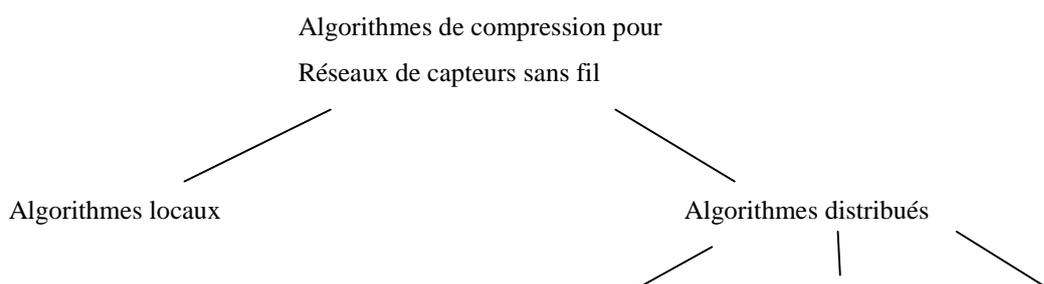
concernant la compression d'images. Cependant, la limitation de ressources des nœuds de capteurs, comme la mémoire ou la vitesse des processeurs, rend inapplicables la plupart des algorithmes de compression existants utilisés dans l'informatique traditionnelle.

Pour que la compression de données soit rentable à la source, il faut que le traitement des données et la transmission des données compressées coûte moins d'énergie qu'un scénario en absence de compression. En effet, quelques travaux de recherches ont démontré que la complexité de certains algorithmes de compression conduit à des

Consommations d'énergie plus importantes que la transmission simple d'une image sans aucun traitement. Dans ce chapitre, nous concentrons notre étude dans les algorithmes de compression d'image spécifiques aux réseaux de capteurs.

## 9.2. Classification des algorithmes de compression d'image pour réseaux de capteurs

En principe, les méthodes de compression d'image traditionnelles ont été conçues pour travailler localement sur un seul et même processus. Dans les réseaux de capteurs, cela implique l'exécution des calculs au niveau du nœud capteur source seulement (ou dans un seul nœud). D'autre part, la nature distribuée des réseaux de capteurs nous permet d'imaginer des approches qui considèrent la distribution du traitement de données entre plusieurs capteurs. D'une manière générale, deux types d'algorithmes de compression distribués ont été rapportés. Le premier est basé sur la distribution du processus de compression d'une image à travers plusieurs nœuds, et le deuxième est basé sur la corrélation existante entre deux ou plus images principalement quand elles contiennent des scènes voisines. [32]



Distribution de la charge	Fusion d'images	Algorithmes distribués
D'images individuelles	(overlapping)	hétérogènes

Figure 3.5 classification des méthodes de compression

Seulement dans ce travail nous concentrons notre étude sur les algorithmes de compression d'image locaux (qui s'exécutent au niveau du nœud source).

## 10. Conclusion

Les besoins d'applications pour les réseaux de capteurs d'images deviennent de plus en plus nombreux. La demande aujourd'hui concerne notamment les applications environnementales, militaires et de surveillance. Avec la contrainte de la limitation des ressources des nœuds, des mécanismes de traitement et de transmission d'images efficaces en énergie d'autres approches et méthodes restent à développer.

Dans le domaine du traitement d'images, on remarque des efforts de plusieurs auteurs pour adapter les algorithmes de compression d'images de l'informatique traditionnelle, comme JPEG ou JPEG2000, aux contraintes particulières des réseaux de capteurs sans fil. Cependant, même si les algorithmes classiques présentent de remarquables performances en termes de ratio débit/distorsion, des expériences sur de vrais capteurs démontrent que ces algorithmes sont beaucoup trop coûteux en énergie, plus coûteux que la transmission d'une image sans compression. Plusieurs adaptations ont été proposées pour rendre ces algorithmes moins gourmands en énergie, les résultats sont discutables. Plus de recherche est donc nécessaire dans ce domaine afin de créer des algorithmes de compression d'image de faible complexité calculatoire et donc de faible consommation énergétique, afin de maximiser la durée de vie ou l'autonomie énergétique du nœud capteur d'images le plus longtemps possible.

# CHAPITRE 4

## *Algorithme de Compression à faible consommation*

### **1. Introduction**

La compression de l'image à la source, dans le cas d'un réseau de capteurs d'images sans fil, est la solution naturelle pour réduire la consommation énergétique des nœuds. Cette opération va permettre au nœud source, et par conséquent aux nœuds de transit (réseaux multi sauts), de traiter (beaucoup) moins de paquets. Mais compresser une

image engage des calculs, donc une dépense d'énergie. Pour que cela soit rentable pour la source, il faut évidemment que le coût d'énergie de la compression de l'image soit inférieur à l'économie d'énergie qu'elle amène sur la transmission des paquets. En effet, les méthodes de compression bien connues comme SPIHT, JPEG et JPEG2000, ont beau fournir d'excellentes performances en termes de rapport débit-distorsion, elles ne sont pas éligibles dans le contexte des réseaux de capteurs. Deux raisons à cela :

- D'abord, leur complexité est telle que cela coûte plus d'énergie de compresser l'image que de la transmettre sans compression, et cela est prouvé expérimentalement
- Ensuite, ces méthodes résistent très mal aux erreurs (perte de paquets) apportées par le canal de transmission (l'absence de quelques bits d'information au récepteur suffit pour entraîner une dégradation brutale de la qualité de l'image décodée, voire même l'impossibilité de décoder l'image).

Dans ce chapitre, nous allons nous intéresser à des méthodes de compression d'image numérique basée sur le célèbre algorithme JPEG à partir de la DCT et aussi sur l'algorithme EZW utilisant la DWT. Dans le cas de la DCT nous allons présenter l'algorithme JPEG zonale, mais avec une méthode qui nous permet de réduire le taux de calcul introduit par les différentes étapes formants JPEG. Ceci afin de minimiser l'énergie consommée par les opérations de calcul au niveau du nœud capteur d'image et augmenter ainsi la durée de vie de ces nœuds formant le réseau de capteurs d'images.

## 2. Algorithme de compression JPEG

Les principales étapes de la chaîne de compression d'image par l'algorithme JPEG sont montrées sur la figure suivante :

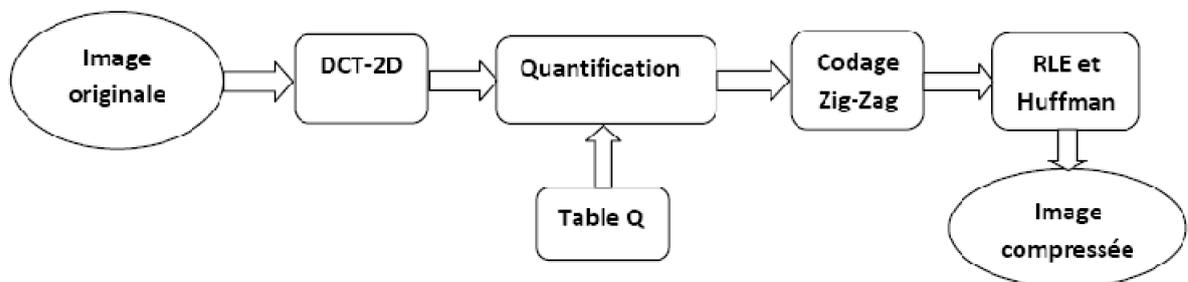


Figure 4.1 : principe de base du codeur JPEG

- **La transformée par DCT**

La transformée DCT (Discrete Cosine Transform) est une transformation numérique qui est appliquée à chaque bloc et pour chaque « couleur ». Cette transformée est une variante de la transformée de Fourier. Cette méthode permet de décrire chaque bloc en une carte de fréquences et en amplitudes plutôt qu'en pixels et couleurs. La valeur d'une fréquence reflète l'importance et la rapidité d'un changement, tandis que la valeur d'une amplitude correspond à l'écart associé à chaque changement de couleur. À chaque bloc de 8x8 pixels, sont ainsi associées 8x8 fréquences, Le calcul classique de chaque coefficient DCT (F(u,v)) pour un bloc de 8x8 est réalisé par l'équation suivante :

$$F(u, v) = \frac{C(u)C(v)}{4} \sum_{x=0}^7 \sum_{y=0}^7 \left[ \frac{f(x,y) \cos\frac{(2x+1)u\pi}{16} \cos(2y + 1) v\pi}{16} \right] \quad (4.1)$$

Avec u et v représentant les deux fréquences spatiales, respectivement horizontales et verticales, du bloc considéré. La taille du bloc est supposé carré 8x8 donc :

$$\text{Avec } C(0) = \frac{1}{\sqrt{2}}$$

- **Quantification**

La quantification est l'étape dans laquelle on perd réellement des informations (et donc de la qualité visuelle), mais c'est celle qui fait gagner beaucoup de place (contrairement à la DCT, qui ne compresse pas). La DCT a retourné, pour chaque bloc, une matrice de 8x8 nombres (dans l'hypothèse que les blocs de l'image font 8x8 pixels). La quantification consiste à diviser cette matrice par une autre, appelée matrice de quantification, et qui contient 8x8 coefficients savamment choisis par le codeur. Le but est ici d'atténuer les hautes fréquences, c'est-à-dire celles auxquelles l'œil humain est très peu sensible. Ces fréquences ont des amplitudes faibles, et elles sont encore plus atténuées par la quantification (les coefficients sont même ramenés à 0).



Des expériences et des testes réalisés sur un capteur d'image de type « cyclops » monté sur un nœud sans fil de type « Mica2 » de crossbow, ont aboutis à la mesure de la consommation énergétique d'un cycle de travail de ce capteur .Un cycle de travail qui correspond aux étapes de capture de l'image, et ensuite une compression par l'algorithme JPEG, et la transmission de l'image compressée par paquets jusqu'au nœuds puits. Les résultats trouvés sont montrés sur la figure suivant :

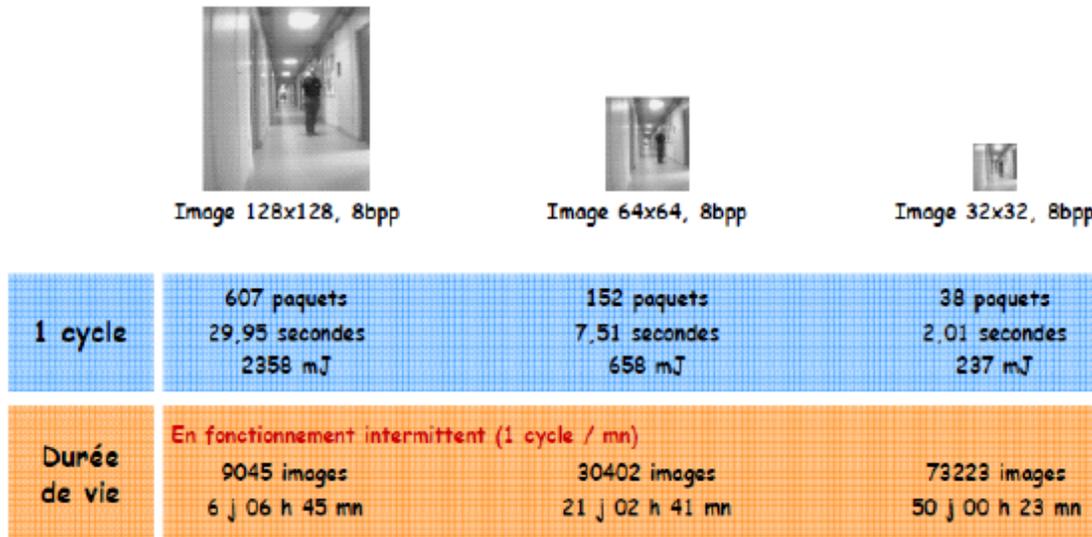


Figure 4.4 : Consommation énergétique d'un capteur d'images exécutant le JPEG [32]

On peut dire d'après ces résultats, que l'algorithme de compression JPEG est très couteux en énergie, et la durée de vie du capteur d'image est de quelques jours seulement en d'autre terme, la pile alimentant le nœud capteur d'image ne le suffit que pour quelques jours, or, dans le cahier des charges de la mise en place d'un réseau de capteur d'image on veut un réseau de durée de vie de quelques mois voire quelques années, ce qui rend l'algorithme JPEG inapplicable dans les applications qui exigent une longue durée de vie des nœuds .

#### 4. Temps d'exécution du JPEG

Une métrique importante pour évaluer un algorithme de compression est son temps d'exécution. Naturellement, plus de temps est passé pour le calcul, le plus de l'énergie sera consommée par le processeur.

De toutes les étapes de l'algorithme JPEG, c'est la DCT (Discrete Cosine Transform) qui coûte le plus de calculs, et du temps, et donc le plus d'énergie. Avec une telle complexité, il existe un besoin pour optimiser cette opération, pour la rendre plus rapide et moins couteuse en énergie, et donc applicable sur des composants électroniques plus limités que

les ordinateurs de nos jours, comme les appareils photo-numériques, les noeuds de capteurs d'image sans fil.

## 5. Modèle de la consommation énergétique du JPEG

Nous pouvons supposer que l'énergie consommée par un capteur d'image sans fil lorsqu'il exécute le JPEG est donnée par l'expression suivante [33]

$$E_p = E_{dct} + E_q + E_z + E_{RLE} + E_{Huf} \quad (4.2)$$

$E_{dct}$  : L'énergie consommée lors de l'exécution de la DCT2D

$E_q$  : L'énergie consommée lors de l'exécution de la quantification

$E_z$  : L'énergie consommée lors de l'exécution du balayage en Zigzag

$E_{RLE}$  : L'énergie consommée lors de l'exécution du codage RLE

$E_{Huf}$  : L'énergie consommée lors de l'exécution du codage Huffman

### 5.1 Modèle énergétique de la DCT2D

Pour développer le modèle énergétique de la DCT2D commençons par rappeler l'expression de cette dernière :

$$F(u, v) = \frac{C(u)C(v)}{4} \sum_{x=0}^7 \sum_{y=0}^7 \left[ \frac{f(x,y) \cos(2x+1)u\pi}{16} \cos(2y+1)v\pi \right]$$

(4.3)

Avec  $u$  et  $v$  représentant les deux fréquences spatiales, respectivement horizontales et verticales, du bloc considéré. La taille du bloc est supposé carré 8x8 donc :

$$\text{Avec } C(\mathbf{0}) = \frac{1}{\sqrt{2}}$$

Sous forme matricielle nous pouvons écrire  $F(u,v)$  comme suit :

$$F_{(SXS)} = A_{(SXS)} P_{(SXS)} A_{(SXS)}^T$$

(4.4)

Avec

$$A(u, v) = \sqrt{\frac{1}{4}} \cos \frac{(2v+1)u\pi}{16} \text{ Si } 1 \leq u \leq k-1$$

Chaque produit matriciel 8x8 de l'équation (4.4) consiste à calculer 64 coefficients. Chaque coefficient nécessite 8 multiplications et 7 additions. Ainsi, l'énergie dissipée pour le calcul d'un produit de deux matrices est :

$$e_{dct} = 128(8e_{mult} + 7e_{add})$$

(4.5)

Où  $e_{mult}$  et  $e_{add}$  sont respectivement l'énergie consommée pour une instruction de multiplication et une instruction d'addition.

Pour une image de taille NxN, c'est à dire possédant  $\left(\frac{N}{8}\right)^2$  blocs, l'énergie totale dissipée dans cet étage est :

$$E_{dct} = \left(\frac{N}{8}\right)^2 e_{dct} = \left(\frac{N}{8}\right)^2 128(8e_{mult} + 7e_{add})$$

(4.6)

## 5.2 Modèle énergétique de la quantification

L'opération de quantification fait intervenir un bloc DCT2D (généralement 8x8 coefficients) à savoir  $F(u,v)$  et une matrice de quantification notée  $Q(u,v)$ . Pour chaque bloc 8x8 la quantification consiste à effectuer :

$$F^Q(u, v) = IntegerRound\left(\frac{F(u, v)}{Q(u, v)}\right) \text{ pour } 0 \leq u, v \leq 7$$

(4.7)

Ainsi, nous avons 64 divisions et 64 opérations d'arrondi par bloc. L'énergie dissipée pour un bloc peut être donnée par :

$$e_q = 64(e_{div} + e_r)$$

(4.8)

Où  $e_{div}$  et  $e_r$  sont respectivement l'énergie dissipée par une opération de division et par une

opération d'arrondi. Pour une image possédant  $\left(\frac{N}{8}\right)^2$  blocs, l'énergie totale dissipée dans cet étage est :

$$E_{1q} = \left(\frac{N}{8}\right)^2 e_{1q} = \left(\frac{N}{8}\right)^2 64(e_{1div} + e_{1r}) \quad (4.9)$$

### 5.3 Modèle énergétique du balayage Zigzag

Cet étage consiste en un réarrangement des 63 coefficients AC de la DCT2D 8x8. L'énergie consommée dans ce cas pour chaque bloc peut être représentée par :

$$e_z = 63e_{sh} \quad (4.10)$$

Où  $e_{sh}$  représente l'énergie dissipée par une opération de décalage. Pour une image possédant  $\left(\frac{N}{8}\right)^2$  blocs, l'énergie totale dissipée dans cet étage est :

$$E_z = \left(\frac{N}{8}\right)^2 e_z = \left(\frac{N}{8}\right)^2 63e_{sh} \quad (4.11)$$

Dans notre travail nous avons négligé le modèle énergétique du codage RLE et de Huffman. En effet, le calcul de la DCT2D est de loin le plus onéreux en termes de coût de calcul. Si nous arrivons à réduire le calcul de la DCT nous sommes sûrs d'avoir un rendement énergétique important dans toutes les parties du JPEG en particulier les parties citées ci-dessus.

## 6. Principe de l'algorithme JPEG Zonale

L'algorithme que nous allons développer dans cette partie de notre mémoire permet de réduire le calcul et le coût énergétique introduit par toutes les étapes de la compression par JPEG.

Comme la DCT classique de l'algorithme de compression JPEG est appliquée sur des blocs de pixels de taille fixe de 8x8 pixels, alors nous proposons un algorithme JPEG basé sur une version simplifiée de la DCT spéciale communément appelée DCT zonale. Cela consiste à prendre en considération uniquement les coefficients de DCT2D 8x8 sur une région appropriée, généralement une forme géométrique simple, du bloc de la DCT2D 8 ×

8. Il s'agit de calculer seulement les coefficients les plus importants, c'est-à-dire ceux de basses fréquences, appartenant à cette forme géométrique préalablement choisie. Ainsi, nous pouvons prétendre à un gain en temps de calcul dans les différentes phases du JPEG à savoir :

- Dans le calcul de la DCT2D  $8 \times 8$ . Nous n'allons calculer que les coefficients appartenant à cette zone choisie,
- La quantification concernera uniquement ces coefficients,
- Le balayage aussi va s'intéresser qu'à ces coefficients,
- Le codage RLE et/ou de Huffman ne sera appliqué que sur les coefficients retenus. En ce qui concerne la forme géométrique de cette zone de coefficients, nous pouvons distinguer deux formes possibles :
  - Une forme carrée de longueur  $n$ , avec  $n < 8$ , on parle alors de la norme de compression S-JPEG (Square-JPEG)
  - Une forme triangle rectangle de cathéter  $n < 8$ , on parle alors de la norme de compression T-JPEG (Triangular-JPEG)

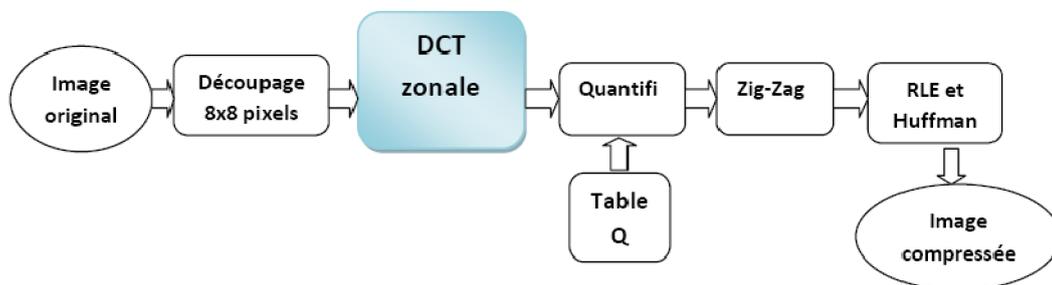


Figure 4.5 Schéma de principe de la compression zonale

### 6.1. Méthode S-JPEG

La zone des coefficients (ou fréquences) calculés correspond à un carré de longueur  $n$ , avec  $n < 8$ , situé dans la partie supérieure gauche du bloc  $8 \times 8$  de la DCT2D (Figure 3.2). On aura alors  $n^2$  coefficients à traiter et les autres sont négligés (ils seront mis à zéro côté décodeur). Plus le paramètre  $n$  est petit, et plus le temps pour exécuter la DCT, puis quantifier et coder les coefficients sera rapide [33] [34] [35].

DC	5	3	0				
8	-1	1	-5				
-5	0	-2	2				
2	1	2	1				

Figure 4.6 : forme carrée de la DCT zonale

## 6.2. Méthode T-JPEG

Dans cette méthode, la zone des fréquences calculées est définie par un triangle rectangle de cathète , situé dans la partie supérieure gauche de la zone carrée précédemment mentionnée (Figure 3.6). Dans cette forme, il y a seulement  $\frac{\rho(\rho+1)}{2}$  coefficients ou fréquences significatives au lieu de  $\rho^2$  [33] [34] [35].

DC	5	3	0				
8	-1	1					
-5	0						
2							

Figure 4.7 : forme triangulaire de la DCT zonale

Remarquons que pour une valeur de donnée, la forme triangulaire va permettre de diminuer encore plus le temps de calcul pour les étapes de quantification et de codage des coefficients comparée à la forme carrée, puisqu'il y a moins de coefficients à traiter.

## 7. Comparaison des performances des deux formes

Une comparaison de l'efficacité énergétique des deux formes (carrée et triangulaire) de la DCT zonale rapide a été menée. Pour une valeur de donnée, la forme triangulaire est plus économique en énergie puisqu'il y a moins de coefficients à calculer, à quantifier et à encoder. En contrepartie, la distorsion de l'image sera plus élevée, notamment pour des niveaux de quantification de moyenne et de haute qualité, puisqu'un plus grand nombre de coefficients sont négligés. Toutefois, l'efficacité énergétique se rapporte au rapport énergie distorsion pour un débit donné, ou encore au rapport énergie

débit pour une distorsion donnée. Nous avons mené une série d'expériences sur les mêmes images utilisées précédemment, à savoir images Pepper, Barbara, boat et Baboon pour des débits de 0.1 à 0.7bpp. Dans tous les cas, il apparait que la forme carrée est la plus efficace. Cela s'explique facilement: Les résultats montrent en effet que la distorsion de l'image obtenue est pratiquement la même dans les deux formes lorsque le nombre de coefficients à traiter est similaire. Par exemple, pour la forme carrée avec  $N = 4$ , il y a 16 coefficients à traiter et pour la forme triangulaire avec  $N = 5$ , il y en a 15. Ces deux scénarios sont comparables à débit constant puisqu'ils donnent à peu près le même niveau de distorsion. Et au final, le coût de la DCT est en faveur de la forme carrée, alors que les coûts de la quantification et du codage des coefficients sont très proches. Prenons un autre exemple symbolique : Pour la forme carrée avec  $N = 6$  et la forme triangulaire avec  $N = 8$ , il y a 36 coefficients à traiter.

	Débit ( <i>bpp</i> )	Image légumes	Image Barbara	Image Boat	Image Baboon
	0.1	22.15	23.02	18.75	19.05
	0.2	24.05	25.10	21.05	20.05
Forme carrée	0.3	25.85	26.85	24.18	21.20
Avec $N=6$	0.4	26.30	27.88	25.46	22.05
	0.5	27.15	29.15	26.04	22.60
	0.6	28.05	30.01	27.10	24.04
	0.7	29.01	30.85	28.05	24.50

Tableau 4.1 Les *PSNR* en fonction des débits obtenus par une DCT zonale de forme carrée avec  $N=6$

	Débit ( <i>bpp</i> )	Image légumes	Image Barbara	Image Boat	Image Baboon
	0.1	19.26	19.75	17.87	18.45
	0.2	22.60	23.80	20.80	19.45
Forme	0.3	23.10	24.20	22.88	20.80
triangulaire	0.4	23.90	25.65	23.66	21.65
Avec $N=6$	0.5	25.05	27.75	24.95	22.05

	0.6	26.90	28.50	25.45	23.75
	0.7	27.85	29.20	26.05	24.02

Tableau 4.2 Les *PSNR* en fonction des débits obtenus par une DCT zonale de forme triangulaire avec  $\alpha=6$

	Débit ( <i>bpp</i> )	Image legumes	Image Barbara	Image Boat	Image Baboon
	0.1	18.85	19.12	17.55	17.85
	0.2	21.05	23.12	20.15	19.95
Forme carrée	0.3	21.88	23.85	22.10	20.00
Avec $\alpha=4$	0.4	22.50	24.25	22.87	20.95
	0.5	23.60	26.05	24.55	21.80
	0.6	25.10	27.40	25.85	22.75
	0.7	26.08	28.20	26.82	23.15

Tableau 4.3 Les *PSNR* en fonction des débits obtenus par une DCT zonale de forme carrée avec  $\alpha=4$

	Débit ( <i>bpp</i> )	Image legumes	Image Barbara	Image Boat	Image Baboon
	0.1	18.34	18.85	16.80	17.00
	.2	21.40	22.85	19.75	19.05
Forme	0.3	21.03	23.40	21.68	19.20
triangulaire	0.4	21.80	23.55	22.04	20.15
Avec $\alpha=4$	0.5	23.10	25.95	23.55	20.95
	0.6	24.98	26.75	24.05	22.25
	0.7	25.95	27.65	24.85	22.04

Tableau 4.4 Les *PSNR* en fonction des débits obtenus par une DCT zonale de forme triangulaire avec  $\alpha=4$

	Débit ( <i>bpp</i> )	Image légumes	Image Barbara	Image Boat	Image Baboon
	0.1	16.65	17.02	15.75	15.65
	.2	18.80	21.05	18.20	17.75
Forme carrée	0.3	19.40	21.25	20.05	17.95
Avec $\alpha=2$	0.4	20.65	22.05	20.20	18.70
	0.5	21.45	24.80	21.90	19.30
	0.6	22.85	26.10	23.05	20.15
	0.7	23.15	26.80	23.95	20.70

Tableau 4.5 Les *PSNR* en fonction des débits obtenus par une DCT zonale de forme carrée avec  $\alpha=2$

	Débit ( <i>bpp</i> )	Image Légumes	Image Barbara	Image Boat	Image Baboon
	0.1	15.05	15.90	14.10	14.90
	.2	18.65	19.95	17.85	16.25
Forme triangulaire	0.3	19.05	20.10	18.85	16.50
	0.4	19.80	20.80	19.05	17.15
Avec $\alpha=2$	0.5	20.70	23.85	20.15	17.90
	0.6	21.35	24.20	20.85	19.60
	0.7	22.50	25.05	21.35	20.20

Tableau 4.6 Les *PSNR* en fonction des débits obtenus par une

		Image légumes	Image Barbara	Image Boat	Image Baboon
	6	27.15	29.15	26.04	22.60
Débit=0.5 bpp	4	23.60	26.05	24.55	21.80
	2	21.45	24.80	21.90	19.30

Tableau 4.7 Les *PSNR* en fonction de  $\alpha$  pour une forme carrée et un débit constant de 0.5 bpp

		Image légumes	Image Barbara	Image Boat	Image Baboon
	6	25.05	27.75	24.95	22.05
Débit=0.5 bpp	4	23.10	25.95	23.55	20.95
	2	20.70	23.85	20.15	17.90

Tableau 4.8 Les *PSNR* en fonction de  $\alpha$  pour une forme triangulaire et un débit constant de 0.5 bpp

A titre d'exemple, si nous prenons l'image Barbara et nous représentons les courbes *PSNR* en fonction du débit pour les trois valeurs de  $\alpha$  aussi bien pour une zone carrée que pour une zone triangulaire nous obtenons les résultats suivants (figures 4.8 et 4.9). Il est évident que chaque fois que  $\alpha$  diminue la qualité des images compressées, obtenues après reconstruction, est moindre. Ceci s'explique tout simplement par l'effet d'un filtrage passe-bas plus intense. En effet, en réduisant plus  $\alpha$  nous allons éliminer plus

de composantes HF de la DCT.

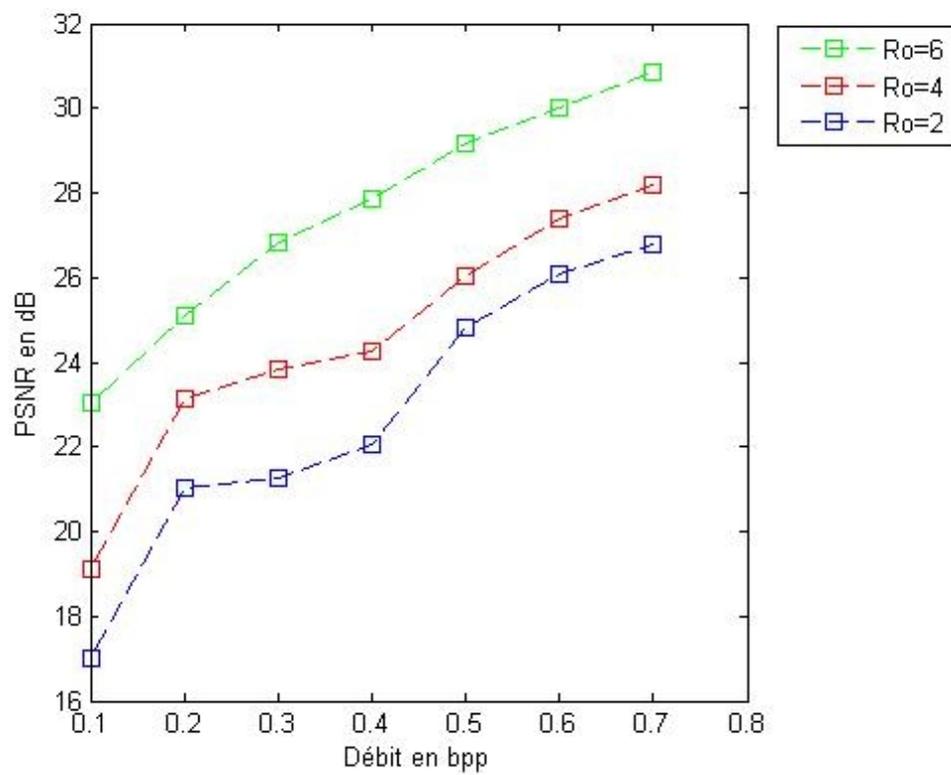


Figure 4.8 : les résultats de la DCT zonale sur l'image Barbara (zone carrée)

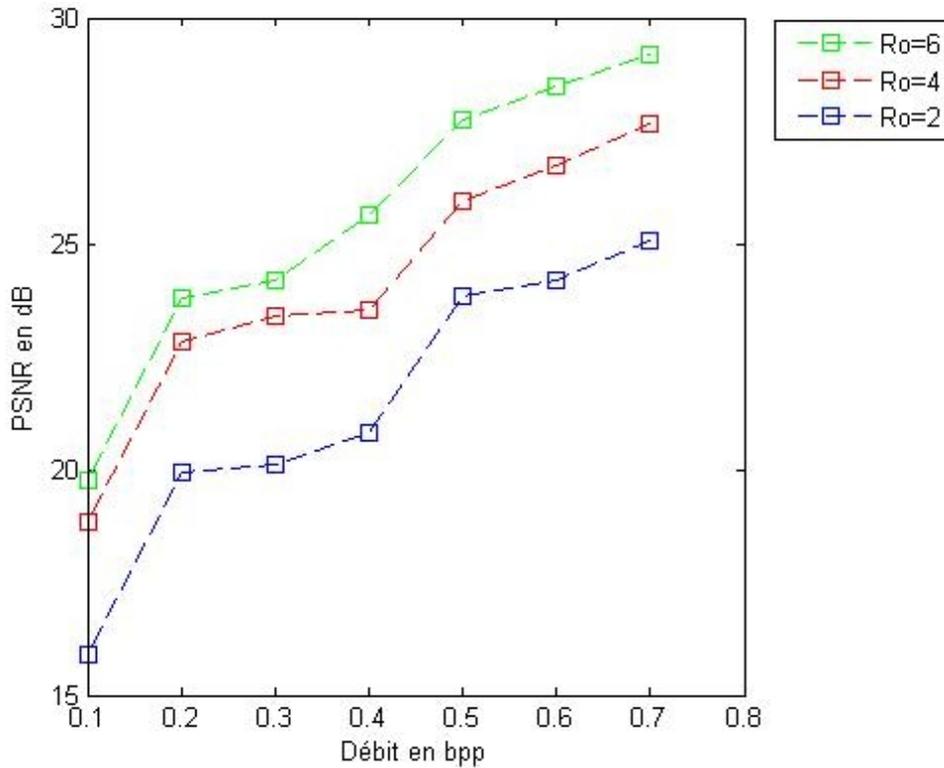


Figure 4.8 : les résultats de la DCT zonale sur l'image Barbara (zone triangulaire)

## 8. Calcul du gain énergétique entre JPEG et JPEG zonale

Considérons une zone triangulaire avec  $\rho=4$  représentant les coefficients de la DCT 2D que nous allons garder (figure 3.7)

Le nombre de coefficients à calculer dans ce cas est :

$$C_\rho = \frac{\rho(\rho + 1)}{2} \quad < \quad 64 \quad (4.12)$$

Ainsi les différents modèles énergétiques deviennent

### Modèle énergétique de la DCT2D zonale

En considérant toujours la même zone triangulaire de dimension  $\rho$  nous pouvons déterminer, de la manière, l'énergie dissipée par le calcul de la DCT zonale. Nous obtenons donc une énergie qui dépend entre autres de  $\rho$ .

$$E_{dct}^* = \left(\frac{N}{8}\right)^2 (8\rho + C_\rho) (8e_{mult} + 7e_{add}) \quad (4.13)$$

### Modèle énergétique de la quantification en prenant en considération la zone

Même si cette zone concerne essentiellement la DCT elle aura une conséquence sur les opérations de quantification. En effet, nous n'aurons plus besoin de quantifier des coefficients qui n'ont pas été calculés. L'énergie dissipée par l'opération de quantification dans le cas où les coefficients sont calculés par une DCT zonale triangulaire prend la forme suivante :

$$E_q^z = \left(\frac{N}{8}\right)^2 C_\rho (e_{div} + e_r)$$

(4.14)

### Modèle énergétique du balayage en Zigzag en prenant en considération la zone

Enfin, cette DCT zonale aura aussi des conséquences sur le balayage zigzag. Le résultat est représenté par le calcul de la consommation énergétique de cette opération de quantification :

$$E_{dct}^z = \left(\frac{N}{8}\right)^2 (C_\rho - 1) e_{sh}$$

(4.15)

Ainsi, si nous désirons comparer la consommation énergétique entre le JPEG classique et le JPEG zonale nous aurons :

	La DCT	La quantification	Le zigzag
JPEG normale	$E_{dct} = \left(\frac{N}{8}\right)^2 e_{dct} = \left(\frac{N}{8}\right)^2 128(8e_{mult} + 7e_{sh})$	$E_q = \left(\frac{N}{8}\right)^2 e_q = \left(\frac{N}{8}\right)^2 64(e_{div} + e_r)$	$E_z = \left(\frac{N}{8}\right)^2 e_z = \left(\frac{N}{8}\right)^2 63e_{sh}$
JPEG zonale	$E_{dct}^z = \left(\frac{N}{8}\right)^2 (8\rho + C_\rho)(8e_{mult} + 15e_{sh})$	$E_q^z = \left(\frac{N}{8}\right)^2 C_\rho (e_{div} + e_r)$	$E_z^z = \left(\frac{N}{8}\right)^2 (C_\rho - 1)e_{sh}$

Rendement	$\frac{E_{dct}^\triangleright}{E_{dct}} = \frac{8\rho + C_\rho}{128}$	$\frac{E_q^\triangleright}{E_q} = \frac{C_\rho}{64} = \frac{\rho(\rho+1)}{128}$	$\frac{E_z^\triangleright}{E_z} = \frac{C_\rho - 1}{63}$
-----------	---	---	--

Tableau 4.9 : Rendement énergétique entre JPEG zonale et JPEG

De cette manière les différents rendements, selon les différentes parties du JPEG, dépendent seulement du paramètre  $\rho$  qui représente les dimensions du triangle.

Le rendement de la DCT peut donc se calculer par l'expression suivante :

$$R_{dct} = \frac{E_{dct}^\triangleright}{E_{dct}} = \frac{8\rho + C_\rho}{128}$$

(4.16)

## 9. CONCLUSION

Dans ce quatrième chapitre, nous nous sommes intéressés à l'étude, l'implémentation et l'évaluation de l'algorithme JPEG utilisant une DCT zonale rapide. Cette DCT zonale rapide réduit le nombre de coefficients à calculer, et donc à quantifier et à encoder. Elle entraîne mécaniquement une réduction de la complexité de calcul de la chaîne de compression, et par incidence une réduction de la consommation d'énergie sur les systèmes de capture. Elle est particulièrement intéressante dans le contexte des réseaux de capteurs sans fil où le problème de la consommation d'énergie est un paramètre crucial.

# **CONCLUSION GENERALE**

## **Conclusion générale**

La compression des données et particulièrement des images date déjà de plusieurs décennies. Cependant, elle est toujours d'actualité compte des exigences technique et technologique liées aux multimédias d'une manière générale et aussi aux moyens de communications comme les réseaux informatiques. De nombreux algorithmes ont vus le jour et certains sont devenus des normes et des standards. Les techniques de compression avec pertes assurent généralement des taux de compression relativement élevés mais avec un certain degré de dégradation. Les objectifs des nouvelles techniques de compression est

donc de réduire plus et d'altérer les moins possible. Parmi les techniques de compression avec perte nous avons les méthodes basées sur les transformations comme par exemple la DCT et la DWT.

Dans notre mémoire nous nous sommes intéressés en premier lieux à une techniques de compression basée sur la DWT et opérant sous forme progressive à savoir la technique SPIHT. Il s'agit d'une amélioration d'une autre technique de la même espèce en l'occurrence EZW. Nous avons donc mis au point cet algorithme et nous l'avons testé et évalué sur plusieurs images tests. Les résultats sont très concordants avec les travaux d'autres chercheurs dans le même domaine. D'autre part, ces résultats prouvent bien que cet algorithme est très performant essentiellement pour certains types d'images. Evidemment, il est tributaire de la transformée en ondelette et les limites de cette dernière l'handicapent un peu.

Dans la seconde partie de notre travail nous nous sommes intéressés aux réseaux de capteurs sans fil et la problématique posée lorsque nous souhaitons compresser des images transmises via ces réseaux. En effet, nous nous trouvons devant un dilemme : il faut compresser car les images sont très volumineuses et ce type de réseaux n'a pas suffisamment de ressources en termes de bande passante pour véhiculer cette quantité de données. D'un autre côté ces équipements, à savoir les WSN, ont une autonomie énergétique limitée et une complexité des algorithmes exécutés par leurs processeurs va réduire énormément la durée de l'autonomie énergétique. La solution est donc de proposer et d'adopter d'autres techniques de compression voir même une autre vision de stratégie pour cette compression.

Nous avons donc pris un exemple assez simple d'un algorithme classique à savoir le JPEG et nous avons essayé de réduire sa complexité algorithmique dans le but de rendre la consommation énergétique plus faible.

## REFERENCES BIBLIOGRAPHIQUES

### Références bibliographiques

- [1] V. Chappelier "Codage progressif d'images par ondelettes orientées" Thèse de Doctorat. Université de Rennes 1, 2005.
- [2] A. Boukaache "Compression d'images hautes définitions en utilisant une transformation hybride" Thèse de doctorat 2012 Guelma algérie
- [3] P. J. Burt & E. H. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31(4), pp. 532-540, 1983.
- [4] J. W. Woods & S. D. O'Neil, "Subband coding of images," *IEEE Transactions on*

- Acoustics, Speech and Signal Processing*, vol. 34 (no 5): pp. 1278--1288, Oct. 1986.
- [5] E. H. Adelson, E. Simon, & R.Hingorani, "Orthogonal pyramid transform for image coding,"*In Visual Communications and Image Processing: VCIP'87*, volume 845 II, pp. 50-59, 1987.
  - [6] S. G. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation,"*IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11 (no 7): pp. 674-693, Juillet 1989.
  - [7] M. Vetterli, "Wavelets, approximation and compression,"*in IEEE Signal Processing Magazine*, pp. 59-73, Sept. 2001.
  - [8] E. Le Pennec, "*Bandelettes et représentation géométrique des images*,"Ph.D., Ecole Polytechnique, Dec. 2002.
  - [9] E. J. Candès, "*Ridgelets: Theory and Applications*,"Thèse de doctorat, Stanford university, 1998.
  - [10] E. J.Candès, & D. L.Donoho, "Curvelets – a surprisingly effective nonadaptive representation for objects with edges," *in Curve and Surface Fitting*, A. Cohen, C. Rabut, and L. L. Schumaker, Eds. Saint-Malo: Vanderbilt University Press, 2000.
  - [11] Y. Gaudeau, "*Contributions en compression d'images médicales 3D et d'images naturelles 2D*," Thèse de Doctorat de l'Université Henri Poincaré, Nancy 1, 2006.
  - [12] P. Bas, "Compression d'Images Fixes et de Séquences Vidéo,"*Cours ENSERG/INPG*, Laboratoire des Images et des Signaux de Grenoble, 2003.
  - [13] V.A. Allen & J. Bellian, "Sub-Band Coding of the Discrete Cosine Transform in ECG Compression,"*Proc. 15th Ann. Inter. Conf. IEEE Eng. Med. Biol. Soc.*, pp.790-791, Oct. 1993.
  - [14] V.A. Allen & J. Bellian, "ECG Data Compression Using the Discrete Cosine Transform (DCT),"*Proc. Comput. Cardiol. IEEE*, pp. 687-690, Oct. 1992.
  - [15] H. Lee & K.M. Buckley, "Heart-Beat Data Compression Using Temporal Beats Alignment and 2 – D Transforms,"*Conf. Rec. Thirtieth Asilomar Conf. Sig. Syst. Comput.*, pp. 1224-1228, 1997.
  - [16] R. Le Page, "*Détection et Analyse de l'Onde P d'un Electrocardiogramme : Application au Dépistage de la Fibrillation Auriculaire*," Thèse de doctorat à l'université de Bretagne occidentale, Fév. 2003.
  - [17] M.L. Hilton, "Wavelet and Wavelet Packet Compression of Electrocardiograms,"*IEEE Trans. Biomed. Eng.*, vol. 44, pp. 394-402, May 1997.
  - [18] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions Signal Processing*, vol. 41, pp. 3445-3462, Dec. 1993.
  - [19] A.Said, & W. A. Pearlman, "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Trans. On Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243-250,1996.

- [20] A. Islam & W. A. Pearlman, "An embedded and efficient low-complexity hierarchical image coder," in *SPIE Visual Communications and Image Processing*, vol. 3653, pp. 294-305, Jan. 1999.
- [21] A. Said & W. A. Pearlman, "Low-complexity waveform coding via alphabet and sample-set partitioning," in *SPIE Visual Communications and Image Processing*, vol. 3024, pp. 25-37, Feb. 1997.
- [22] S. Tai, C. C. Sun & W. C. Yan, "A 2-D ECG compression method based on wavelet transform and modified SPIHT," *IEEE Trans. Biomed. Eng.*, 52(6), pp. 999-1008, 2005.
- [23] Z. Lu, D. Y. Kim & W. A. Pearlman, "Wavelet compression of ECG signals by the set partitioning in hierarchical trees algorithm," *IEEE Trans. Biomed. Eng.*, 47(7), pp. 849-856 2000.
- [24] W.A. Pearlman, B.-J. Kim & Z. Xiong, "Embedded video subband coding with 3D SPIHT," Chap. 24 in *Wavelet Image and Video Compression*, N. P. Topiwala, Ed., 397-432, Kluwer Academic Publishers, Boston, 1998.
- [25] S. Chang & L. Carin, "A Modified SPIHT Algorithm for Image Coding With a Joint MSE and Classification Distortion Measure," *IEEE Trans. Image Processing*, 15(3), pp. 713-725, 2006.
- [26] M. Akter, M. B. I. Reaz, F. Mohd-Yasin & F. Choong, "A Modified-Set Partitioning in Hierarchical Trees Algorithm for Real-Time Image Compression," *Journal of Communications Technology and Electronics*, 53(6), pp. 642-650, 2008.
- [27] C. Duran-Faundez, "Transmission d'images sur les réseaux de capteurs sans fil sous la contrainte de l'énergie", thèse de Doctorat, université de Nancy, 2009.
- [28] Rahim Kacimi, "Techniques de conservation d'énergie pour les réseaux de capteurs sans fil", thèse de Doctorat, université de Toulouse, Septembre 2009.
- [29] Crossbow Technology Inc. <http://www.xbow.com/>.
- [30] TinyOS: *An operating system for networked sensors*. <http://www.tinyos.net/>.
- [31] Koppe, Enrico, Achim Liers, Hartmut Ritter et Jochen Schiller. "Low-power image transmission in wireless sensor networks using ScatterWeb technology". In: First Workshop on Broadband Advanced Sensor Networks. San Jose, CA.2004
- [32] C. D-Faundez , V. Lecuire, "Compression et transmission d'images sur réseau de capteurs sans fil sous la contrainte de l'énergie," Journées non thématiques ResCom, Strasbourg, octobre, 2008.
- [33] A. Mammeri, A. Khoumsi, D. Ziou, B. Hadjou, "Modeling and Adapting JPEG to the energy requirements of VSN, T-JPEG. In: *Proceedings of 17th International Conference on Computer Communications and Networks*", ICCCN '08. US Virgin Islands.
- [34] L. Makkaoui, J-M. Moureaux, "Efficacité énergétique d'une DCT zonale rapide dans le contexte de la compression d'image dans les réseaux de capteurs sans fil", rapport CRAN.2009.
- [35] A. Larab, S. Chaib "Compression des images dans les réseaux de capteurs d'images

