

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي و البحث العلمي

BADJI MOKHTAR-ANNABA UNIVERSITY
UNIVERSITE BADJI MOKHTAR-ANNABA



جامعة باجي مختار - عنابة

Faculté des Sciences de l'Ingéniorat
Département d'Electronique

Année : 2013/2014

THESE

Présentée en vue de l'obtention du diplôme de *DOCTORAT 3^{ème} CYCLE*

CODAGE ET COMPRESSION D'IMAGES POUR DES RESEAUX DE CAPTEURS SANS FIL

Option
Multimédia et Communications Numériques

Par
KOUADRIA Nasreddine

Directeur de Thèse: Djemil MESSADEG

MCA. Univ. Badji Mokhtar-Annaba

Co-Directeur de Thèse: Nouredine DOGHMANE

Prof. Univ. Badji Mokhtar-Annaba

Devant le Jury

Président: M. TOUMI Salah

Pr. Univ. Annaba

Examineurs: M. DJEDDI Mabrouk

Pr. Univ. Boumerdes

M. ZIANI Rezki

Pr. Univ. Tizi ousou

M. KADDECHE Mohamed

MCA. Univ. Annaba

Année Universitaire : 2013/2014

Remerciements

Je tiens à remercier en premier lieu "**ALLAH**" qui m'a donné la patience pour finir ce travail, et que j'espère qu'il me guidera pour réussir dans le reste du chemin.

J'adresse mes plus sincères remerciements à **mes parents** qui m'ont toujours encouragé, il m'est impossible de trouver des mots pour leur remercier pour tout ce qu'ils ont fait.

Je tiens à remercier mon directeur de thèse, **Dr. Messadeg Djemil**, d'avoir dirigé ce travail avec compétence et professionnalisme.

Je tiens à remercier tout particulièrement mon co-directeur de thèse, **Pr. Noureddine Doghmane**, pour son encadrement de qualité, pour son coaching efficace et surtout pour avoir su trouver du temps quand il le fallait malgré son agenda toujours chargé.

Je tiens à remercier **Melle Khaoula Mechouek**, pour son aide et sa patience.

Je tiens à remercier le président de jury, **M. TOUMI Salah**, professeur à l'université de Annaba qui me fait l'honneur de présider ce jury.

Je remercie également les membres du jury, **M. DJEDDI Mabrouk**, professeur à l'université de Boumerdes, **M. ZIANI Rezki**, professeur à l'université de Tizi ousou, et **M. KADDECHE Mohamed**, MCA à l'université de Annaba d'avoir bien voulu juger ce travail.

Mes remerciements vont également à tous ceux qui m'ont aidé de prêt ou de loin à finaliser cette thèse et qui ont contribué dans ma formation.

Dédicace

Je dédie ce travail à :

Tous les étudiants qui,

Malgré le déficit énorme en moyens matériels et pédagogiques,

ont pu s'accomplir et réaliser de très bons résultats !

ملخص

الاستعمال المتزايد لشبكات الاستشعار اللاسلكية البصرية يسمح لمجموعة واسعة من التطبيقات ذات أهداف ومعوقات مختلفة. ومع ذلك، فإن المشكلة المشتركة لجميع تطبيقات شبكات الاستشعار تبقى الطاقة المحدودة لهذه الأخيرة. هذه الأطروحة تركز أساساً على دراسة وتقييم تحويل جيبى متقطع (DCT) موفر للطاقة. التحويل الذى تم تطويره يمكن استخدامه في ضغط الصور في شبكات الاستشعار اللاسلكية البصرية. هو مزيج من تحويلات جيبية متقطعة اقترحت مؤخراً مع طريقة منطقية (zonal approach). وبالتالي، يتم تقليل التعقيد الحسابي بشكل ملحوظ. تظهر النتائج التجريبية أن التحويل المقترح يؤدي إلى انخفاض كبير في الوقت اللازم للحساب في اللاقطات وهذا سوف يؤدي بالتالي إلى توفير في استهلاك الطاقة. التحويل المقترح مناسب جداً للاستعمال في شبكات الاستشعار اللاسلكية التي تعمل بالبطاريات.

كلمات مفتاحية

شبكات الاستشعار اللاسلكية البصرية، لاقطات الصور، ضغط الصورة، تحويل جيبى متقطع، توفير الطاقة، طريقة منطقية zonal approach

Résumé

Les réseaux de capteurs sans fil d'images sont utilisés aujourd'hui dans de nombreuses applications qui diffèrent par leurs objectifs et leurs contraintes individuelles. Toutefois, le dénominateur commun de toutes les applications de réseaux de capteurs d'images reste leur énergie qui est limitée. La contribution de cette thèse porte principalement sur l'étude et l'évaluation d'une transformée en cosinus discrète efficace en énergie. La transformée proposée peut être utilisée dans la compression d'image utilisée dans les réseaux de capteurs sans fil d'images. Il s'agit d'une combinaison de transformées en cosinus discrète de faibles complexités, récemment proposées et une approche d'une DCT zonale. Ainsi, la complexité de calcul est réduite de manière significative. Les résultats expérimentaux montrent que la transformée proposée conduit à une réduction significative des temps de calcul au niveau du nœud capteur. Ceci, se traduira par une économie en énergie. La DCT proposée est très souhaitable pour l'implantation dans les réseaux de capteurs sans fil alimentés par des batteries embarquées.

Mots clés

Réseaux de capteurs sans fil, capteur d'image, compression d'images, conservation de l'énergie, transformée en cosinus discrète, approche zonale.

Abstract

The increasing development of Wireless Visual Sensor Networks today allows a wide variety of applications with different objectives and constraints. However, the common problem of all the applications of sensor networks remains their energy limitation. This thesis concentrates mainly on the study and evaluation of an energy-efficient discrete cosine transform (DCT). It can be used in image compression in wireless visual sensor networks. It is a combination of a recently proposed low complexity discrete cosine transforms and a pruned approach. Thus, the computational complexity is reduced significantly. The experimental results show that the proposed transform leads to a significant reduction in computation time at the target sensor node. This will consequently result in a saving in energy consumption. The proposed DCT is very suitable for implementation in wireless sensor networks powered by batteries.

Keywords

Wireless sensor network, camera sensor, image compression, energy consumption, discrete cosine transform, zonal approach

Liste des abréviations

A/D	Analogue/Digital
AC	Alternating Current
Ah	Ampeur heure
BDCT	Binary Discrete Cosine Transform
bpp	bit per pixels
CAN	Convertisseur Analogique Numérique
CB-DCT	Cintra Bayer - DCT
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
CS	Compressive Sensing
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
DA	Distributed Arithmetic
DARPA	Defense Advanced Research Projects Agency
dBm	décibels milliWatt
DC	Direct Current
DCT	Discrete cosine Transform
DCT 1D	Discrete cosine Transform one dimension
DCT 2D	Discrete cosine Transform tow dimensions
DFT	Discrete Fourier Transform
DHT	Discrete Hartley Transform
DST	Discrete Sine Transform
DWT	Discrete Walvelet Transform
EBCOT	Embedded block coding with optimised truncation
FFT	Fast Fourier Transform
GSM	Global System for Mobile Communications
HEVC	High Efficiency Video Coding
HVS	Human Visual System
intDCT	Integer Discrete Cosine Transform
JPEG	Joint Photographic Experts Group

JPEG2000	Joint Photographic Experts Group 2000
KLT	Karhunen-Loeve Transform
LAN	Local Area Network
LBT	Lapped Biorthogonal Transform
LEACH	Low Energy Adaptive Clustering Hierarchy
MAC	Media Access Control
MCU	MicroController Unit
MEMS	Microelectromechanical Systems
MPEG	Moving Picture
MTE	Minimize Total Energy
PSNR	Peak Signal to Noise Ratio
QCIF	Quarter Common Intermediate Format
RCPC/CRC	Rate-Compatible Punctured Convolutional Codes/Cyclic redundancy check
RGB	Red Green Bleu
RISC	Reduced instruction set computer
ROI	Region Of Interest
RTS/CTS	Request to Send / Clear to Send
SDCT	Signed Discrete Cosine Transform
SensIT	Sensor Information Technology
S-LZW	Sensor-Lempel-Ziv-Welch
S-MAC	Sensor Media Access Control
SPIHT	Set Partitioning In Hierarchical Trees
TC	Taux de Compression
TDMA	Time Division Multiple Access
UIT-T	Union internationale des télécommunications
WISN	Wireless Image Sensor Network
WSN	Wireless Sensor Network
ZCL	Zonal Cordic Loeffler

Liste des figures

Figure 1.1. <i>Architecture conventionnelle d'un réseau de capteurs sans fil</i>	9
Figure 1.2. <i>Architecture conventionnelle d'un nœud capteur</i>	11
Figure 1.3. <i>Pile de protocoles d'un réseau de capteurs sans fil</i>	13
Figure 1.4. <i>Classification des réseaux de capteurs d'images en fonction de leur architecture de communication.</i>	30
Figure 1.5 <i>Classes d'application des réseaux de capteurs d'images</i>	31
Figure 2.1 : <i>Architecture typique d'un réseau de capteurs d'images</i>	39
Figure 2.2. <i>Classification générale des algorithmes de compression d'images pour les réseaux de capteurs d'images</i>	43
Figure 2.3. <i>Classification des algorithmes de compression à la source</i>	44
Figure 2.4 : <i>Architecture typique d'une compression d'image à la base de transformées</i>	44
Figure 2.5 : <i>Architecture typique d'une décompression d'image à la base de transformée</i>	45
Figure 2.6 : <i>Schéma bloc de la chaîne de compression JPEG séquentiel</i>	47
Figure 2.7: <i>Schéma bloc de la chaîne de Décompression JPEG séquentielle</i>	53
Figure 2.8. <i>Coût énergétique de la DCT dans une chaîne JPEG</i>	54
Figure 2.9. <i>Transformée en ondelettes 2D. Schéma de décomposition classique en sous bandes</i>	57
Figure 2.10 : <i>Compression par échantillonnage compressé</i>	61
Figure 3.1: <i>DFT vs DCT</i>	77
Figure 3.2 : <i>Principe de la méthode récursive</i>	79
Figure 3.3 : <i>Principe de la méthode de calcul indirecte</i>	80

Figure 3.4 : Schéma bloc de l'algorithme de Loeffler 8 points	83
Figure 3.5 Schéma bloc de la DCT Chen 8 points	84
Figure 3.6 Principe de la réalisation ligne colonne de la DCT	86
Figure 3.7 Schéma bloc de la réalisation matérielle de la séparation ligne colonne	86
Figure 3.8 : Formes de la DCT zonale : carrée et triangulaire pour $L = 4$	90
Figure 3.9 : L'algorithme rapide pour le calcul de la BDCT	94
Figure 3.10 : L'algorithme rapide pour le calcul de la version zonale de la BDCT ($L=4$)	95
Figure 3.11 : L'algorithme rapide pour le calcul de la CB-DCT	98
Figure 3.12 : L'algorithme rapide pour le calcul de la version zonale de la CB-DCT ($L= 4$)	99
Figure 3.13 : L'algorithme rapide pour le calcul de la DCT entière (ICT)	101
Figure 4.1: Consommation énergétique des différents algorithmes appliqués sur un bloc 8×8 obtenu par le simulateur Atmel Studio avec le microcontrôleur ATmega128L.	109
Figure 4.2: Image originales de tests (8 bpp)	112
Figure 4.3: image Barbara codée à 0.5 bpp par la transformée de Loeffler zonale, pour différentes valeurs de L	119
Figure 4.4: image Barbara codée à 0.5 bpp par la transformée BDCT zonale, pour différentes valeurs de L	120
Figure 4.5: image Barbara codée à 0.5 bpp par la transformée CB-DCT zonale, pour différentes valeurs de L	121
Figure 4.6: image Barbara codée à 0.2 bpp par la transformée de Loeffler zonale, pour différentes valeurs de L	122
Figure 4.7: image Barbara codée à 0.2 bpp par la transformée BDCT zonale, pour différentes valeurs de L	123
Figure 4.8: image Barbara codée à 0.2 bpp par la transformée CB-DCT zonale, pour différentes valeurs de L	124
Figure 4.9: Courbes débit – distorsion moyenne de l'algorithme Loeffler zonale	125
Figure 4.10: Courbes débit – distorsion moyenne de l'algorithme BDCT zonale	125
Figure 4.11: Courbes débit – distorsion moyenne de l'algorithme CB-DCT zonale	126

Figure 4.12: image Goldhill codée à 0.5 bpp et pour $L = 6$, par les transformées : (a)-CB-DCT zonale, (b)-ZCL et (c)-BDCT zonale.129

Figure 4.13: image Goldhill codée à 0.5 bpp et pour $L = 2$, par les transformées : (a)-CB-DCT zonale, (b)-ZCL et (c)-BDCT zonale.130

Figure 4.14: Courbes PSNR moyenne des algorithmes CB-DCT zonale, BDCT zonale et ZCL en fonction du paramètre 'L'. Les résultats sont donnés pour un bitrate de 0.5 bpp131

Figure 4.15: Courbes débit – distorsion des algorithmes BDCT zonale et ZCL pour les images : (a)-Lena, (b)- Goldhill2. Les résultats sont donnés pour une consommation énergétique = 6 μJ pour ZCL et 5,77 μJ pour BDCT zonale.134

Figure 4.17: Courbe débit – distorsion moyenne : comparaison entre la table de quantification dyadique et la table de quantification standard136

Liste des tableaux

Tableau 3.1 Valeurs des transformées d'une rampe	75
Tableau 3.2 Valeurs des transformées et des transformées inverses d'une rampe	76
Tableau 3.3 Comparaison entre les différents algorithmes en termes de nombre d'opérations	82
Tableau 3.4 : Pourcentage de coefficients contenant plus de 99% de l'énergie totale	87
Tableau 3.5 : Complexité de calcul de la DCT pour différentes tailles	88
Tableau 3.6 : Complexité de calcul de la DCT zonale	89
Tableau 3.7 : PSNR pour les deux formes zonale avec une consommation énergétique de 0.04 mJ	90
Tableau 3.8 : Comparaisons des lignes de la matrice SDCT avec ceux de la matrice de HADAMARD pour $N = 8$ en termes de nombre d'entrées qui sont différentes	93
Tableau 3.9 : Complexité calculatoire de la DCT binaire pp.94	
Tableau 3.10 : Comparaison des complexités calculatoire : BDCT vs DCT réelle	95
Tableau 3.11 : Complexité calculatoire de la version zonale de la DCT binaire	96
Tableau 3.12 : Complexité calculatoire de la CB-DCT	98
Tableau 3.13 : Comparaison des complexités calculatoire : CB-DCT vs DCT réelle	99
Tableau 3.14 : Complexité calculatoire de la version zonale de la CB-DCT	100
Tableau 3.15 : Complexité calculatoire de la DCT entière	102
Tableau 3.16 : Comparaison des complexités calculatoire : DCT entière vs DCT réelle	102
Tableau 3.17 : Complexité calculatoire de la version zonale de la DCT entière	102
Tableau 4.1 : Nombre de cycles des différents algorithmes pour le calcul de la DCT appliqués sur un bloc 8×8 obtenu par le simulateur Atmel Studio avec le microcontrôleur ATmega128L.	106

Tableau 4.2: Consommation énergétique et temps d'exécution pour le calcul des algorithmes sur un bloc 8×8 obtenu par le simulateur Atmel Studio avec le microcontrôleur ATmega128L. (a)-Loeffler zonale, (b)BDCT zonale, (c)-CB-DCT zonale	107
Tableau 4.3: Consommation énergétique et temps d'exécution pour le calcul de l'algorithme intDCT zonale sur un bloc 8×8 obtenu par le simulateur Atmel Studio avec le microcontrôleur ATmega128L.....	108
Tableau 4.4: Consommation énergétique et temps d'exécution pour le calcul de l'algorithme ZCL sur un bloc 8×8 avec le microcontrôleur ATmega128L.	108
Tableau 4.5: Résultats de décompression des différentes images de tests par la transformée de Loeffler zonale : (a),(b),(c),(d),(e) et (f) respectivement pour $L = 3,4,5,6,7$ et 8	114
Tableau 4.6: Résultats de décompression des différentes images de tests par la transformée BDCT zonale : (a),(b),(c),(d),(e) et (f) respectivement pour $L = 3,4,5,6,7$ et 8	115
Tableau 4.7: Résultats de décompression des différentes images de tests par la transformée CBDCT zonale : (a),(b),(c),(d),(e) et (f) respectivement pour $L = 3,4,5,6,7$ et 8	117
Tableau 4.8: Résultats de décompression des différentes images de tests par la transformée :(a)-BDCT zonale, (b)-CBDCT zonale et (c)-Z-CL DCT. Les résultats sont donnés pour $L = 2,3,4,5,6,7$ et 8 pour un bitrate de 0.5bpp.....	128
Tableau 4.9 Consommation énergétique des différents algorithmes appliqués sur un bloc 8×8 obtenu par le simulateur Atmel Studio avec le microcontrôleur ATmega128L.	132
Tableau 4.10: Consommation énergétique et temps d'exécution pour le calcul des algorithmes : Z-CL, CBDCT zonale et BDCT zonale. Les résultats sont donnés pour l'image 'Baboon' pour un bitrate de 0.5 bpp	133
Tableau 4.11: Résultats de décompression des différentes images de tests par la transformée de Loeffler et la table de quantification dyadique	135

Table des matières

Introduction générale

Chapitre 1 : Réseaux de capteurs sans fil et problèmes de l'énergie

1.1. Introduction	7
1.2. Caractéristiques des réseaux de capteurs	9
1.2.1. Architecture d'un nœud capteur	9
1.2.2. Les différents facteurs de conception.....	10
1.2.3. Architecture protocolaire pp.....	11
1.2.4. Consommation d'énergie d'un nœud capteur.....	12
1.2.5. Surconsommation d'énergie d'un nœud capteurs	14
1.2.6. Durée de vie d'un réseau de capteurs sans fil	15
1.3. Mécanismes de conservation de l'énergie.....	16
1.4. Techniques de conservation de l'énergie.....	18
1.4.1. Techniques de Duty cycling	18
1.4.2. Techniques de mobilité	19
1.4.3. Techniques orientées données	20
1.5. Applications des réseaux de capteurs scalaires	23
1.5.1. Exemples d'applications typiques	23
1.6. Les réseaux de capteurs d'images	25
1.6.1. Spécificités des réseaux de capteurs d'images	26
1.6.2. Applications des réseaux de capteurs d'image	28
1.6.3. Défis en matière de recherche.....	33
1.7. Conclusion.....	35
Chapitre 2 : La compression d'image dans les réseaux de capteurs sans fil	
2.1. Introduction	38
2.2. La compression d'images dans les réseaux de capteurs d'images.....	41
2.3. La compression à la base de transformées.....	44
2.3.1. Introduction.....	44
2.3.2. Norme JPEG.....	46
2.3.3. Adaptation de JPEG aux contraintes des réseaux de capteurs d'images.....	53

2.3.4. Compression locale par JPEG2000	55
2.3.5. Schéma basé sur le codage SPIHT	57
2.4. Compression distribuée	58
2.4.1. Introduction	58
2.4.2. Compression distribuée d'images corrélées	58
2.4.3. Compression distribuée sur une architecture clustérisée	59
2.5. Échantillonnage compressé (Compressive Sampling : CS)	60
2.5.1. Théorème de Nyquist	61
2.5.2. Application de l'échantillonnage compressé aux réseaux de capteurs d'images	61
2.6. Conception d'une méthode de compression pour un réseau de capteur visuel	62
2.7. Conclusion	63
Chapitre 3 : Transformations de faible complexité	
3.1. Introduction	65
3.2. Les variantes de la DCT	66
3.3. DCT 1D conventionnelle	69
3.4. DCT 2D conventionnelle	71
3.4.1. Définitions	71
3.4.2. Propriétés de la DCT	72
3.5. Intérêt de l'utilisation de la DCT	74
3.6. Algorithmes rapides pour le calcul de la DCT 1D exacte	77
3.6.1. Calcul récursif	78
3.6.2. Calcul indirect	79
3.6.3. DCT à base de convolution	80
3.6.4. Factorisation directe	81
3.7. Architectures pour la DCT 2D	84
3.7.1. Réalisation à base de séparation ligne colonne	85
3.7.2. Réalisation directe	86
3.8. Réduction des coefficients à calculer : DCT Zonale	87
3.9. Combinaison des deux approches : Réduction des coefficients et des opérations à calculer	91
3.9.1. La DCT binaire (BDCT)	91
3.9.2. La DCT arrondie (CB-DCT)	96
3.9.3. La DCT entière 4×4	100

3.10. Conclusion	103
Chapitre 4 : Evaluation des performances	
4.1. Introduction	105
4.2. Nombre de cycles	105
4.3. Consommation énergétique et temps d'exécution	106
4.4. Qualité des images reconstruites	109
4.4.1. Métriques d'évaluation	109
4.4.2. Résultats de compression par les DCT de faibles complexités	111
4.4.3. Discussion	117
4.5. Comparaison avec la méthode proposée par Lecuire et al	127
4.5.1. Comparaison en fonction du paramètre 'L'	127
4.5.2. Comparaison à consommation énergétique égale	133
4.6. Table de quantification dyadique	135
4.7. Conclusion	136
Conclusions et perspectives	138
Références.....	141

Introduction générale

1. Problématique

Les avancées technologiques dans le domaine de la microélectronique et de l'informatique ont donné naissance à une nouvelle génération de réseaux appelés réseaux de capteurs sans fils. Ces derniers forment un ensemble d'appareils électroniques miniaturisés autonomes, alimentés par batterie (pile), équipés de capteurs et pouvant communiquer entre eux sans fils. L'émergence de ces réseaux de capteurs sans fils ouvre la voie au déploiement de nouvelles applications de surveillance des grands systèmes dans une variété de domaines notamment ceux qui s'étalent sur de vastes étendues géographiques. Ils sont capables de superviser une région ou observer un phénomène particulier, de fournir des informations utiles par la combinaison des mesures prises par les différents capteurs, pour les transmettre ensuite de nœud en nœud et les faire parvenir à une station de base.

Cette nouvelle génération de réseaux promet de révolutionner notre façon de vivre, provoque une interaction avec l'environnement qui nous entoure. Grâce à leur flexibilité, leur faible coût et la facilité de déploiement, plusieurs domaines d'applications sont alors envisagés. Nous citons pour l'exemple, la prévision, la détection et la surveillance des phénomènes et catastrophes naturelles, le contrôle de l'environnement, la cartographie de la biodiversité, le bâtiment intelligent, l'agriculture de précision, la surveillance et la maintenance préventive des machines, la médecine et la santé, la logistique et les transports intelligents etc...

Les réseaux de capteurs sans fils sont souvent déployés dans des environnements limités en matière de ressources. Pour les surveillances de longue durée, recharger les batteries est parfois impossible en raison de la densité de déploiement élevée. Cette opération est pratiquement ou économiquement irréalisable. La conservation de l'énergie des nœuds capteurs est devenue donc un problème majeur.

La consommation d'énergie des capteurs influence directement la durée de vie du réseau qui est devenue le critère de performance fondamental. Plusieurs travaux de recherche essayent d'identifier les activités les plus consommatrices en énergie et aussi d'optimiser la consommation énergétique des nœuds capteurs afin d'améliorer leurs performances, notamment la maximisation de leurs durées de vie. La littérature pourtant sur les réseaux de capteurs sans fils reconnaît que l'émetteur radio est l'un des composants les plus gourmands en énergie [1-2], par conséquent, une quantité d'énergie considérable dissipée par un nœud-capteur concerne la transmission et la réception des données.

Les réseaux de capteurs se chargent de capturer des données simples telles que la température, la luminosité, la pression atmosphérique ou les vibrations. Ces données, dites scalaires, peuvent être codées sur quelques octets. Récemment, nous assistons à une nouvelle génération de capteurs d'images miniatures qui sont peu coûteux en énergie. Ces capteurs peuvent être adaptés à une gamme d'applications très variée telle que la détection, le dénombrement et la localisation d'objets par la vision (comptage de la faune dans les réserves naturelles, localisation des départs de feu en forêt, etc.). Cependant, les volumes de données manipulées par ces derniers sont très largement supérieurs aux mesures scalaires classiques, ce qui les rend particulièrement gourmands en énergie. En fait, pour coder une mesure de type scalaire, telle que l'humidité et/ou la température, quelques octets sont largement suffisants (ce qui peut être contenu dans un seul paquet). Quand il s'agit d'une image, l'exemple précédent n'est pas valide, parce que généralement plusieurs milliers d'octets sont nécessaires pour la représenter. Par conséquent, beaucoup de paquets vont être générés par le nœud-capteur pour transmettre l'image entière, et cela mène à une consommation d'énergie considérable.

Comme l'émetteur radio est l'un des composants les plus gourmands en énergie, il semble évident que le coût d'énergie de la transmission de l'image peut être réduit significativement en compressant l'image à la source. Toutefois, les nœuds capteurs sont par nature très limités en capacité de calcul et de mémoire, c'est une conséquence directe des contraintes de consommation d'énergie, de miniaturisation et de coût de fabrication.

En raison de ces limitations, les méthodes de compression d'images existantes ne sont pas éligibles en l'état dans le contexte des réseaux de capteurs d'images. Malgré que, ces méthodes fournissent d'excellentes performances en termes de caractéristiques débit-distorsion, elles exigent beaucoup de calculs et d'accès mémoire. Dans la plupart du temps, et comme c'est le cas de JPEG2000, SPHIT et aussi JPEG, l'opération de compression d'une image coûte plus d'énergie qu'une transmission directe sans compression [3-4].

L'efficacité énergétique doit donc être recherchée, afin de prolonger la durée de vie du capteur d'images ainsi que la durée de vie du réseau tout entier. La conception de méthodes de compression tenant compte de cette efficacité est donc nécessaire pour que le déploiement de réseaux de capteurs d'images puisse être réalisable.

Une chaîne de compression classique est composée de trois étapes essentielles, la transformation, la quantification et le codage. La première étape est la plus complexe à implanter et par conséquent elle est la plus exigeante en matière d'énergie [5]. La présente thèse forme une contribution à l'efficacité énergétique des méthodes de compression d'images, précisément à l'étape de transformation. Notre objectif est de proposer et de concevoir une transformation peu calculatoire et efficace en énergie, et d'intégrer cette dernière dans une chaîne de compression d'image. Par conséquent nous faisons un compromis entre l'énergie consommée par les capteurs et la qualité visuelle des images reconstruites.

2. Organisation du manuscrit

Ce manuscrit est organisé en quatre chapitres. Dans le chapitre 1, nous présentons un état de l'art sur les réseaux de capteurs sans fils et aux technologies associées. Nous commençons par présenter le fonctionnement général de ce type de réseau, ses différentes caractéristiques, ses applications potentielles. Ensuite nous abordons le cas spécifique des réseaux de capteurs d'images, nous traitons les disparités par rapport aux réseaux de capteurs de données scalaires et les principaux axes de recherche de ce domaine en tenant compte de la contrainte énergie. Ce dernier point permet de positionner notre contribution par rapport aux travaux faisant référence dans la même thématique.

Dans le chapitre 2 nous présentons un état de l'art sur les techniques de compression d'images d'une façon générale. Nous traitons aussi le cas particulier des techniques de compression d'images spécifiques aux réseaux de capteurs sans fils. Enfin, nous présentons les travaux les plus représentatifs, en traitement et compression d'images.

Le chapitre 3 constitue le cœur de notre travail de thèse : Nous présentons une transformation optimisée pour la compression d'image dans les réseaux de capteurs sans fils. Après avoir exposé le principe de la transformée en cosinus discrète (DCT) en mettant l'accent sur son utilité et ses principaux algorithmes rapides, nous décrivons l'architecture proposée pour le calcul de la DCT 2D. Cette transformation est d'une part, peu calculatoire donc peu gourmande en énergie, et d'autre part, elle assure une bonne qualité visuelle de l'image restituée après décompression.

Le chapitre 4, constitue une évaluation des performances de notre contribution. Nous commençons par évaluer l'efficacité énergétique des méthodes proposées. Ensuite nous

étudions l'impact de la dimension (paramètre L), sur la qualité visuelle des images en considérant les mesures de qualité objectives (PSNR) et subjectives. Nous proposons certains paramètres optimaux pour l'utilisation de l'architecture proposée. Enfin, et pour donner de la crédibilité aux méthodes proposées, nous procédons à une comparaison [6] avec des travaux de références dans le même domaine, particulièrement les travaux publiés dans [7]

Chapitre 1

Réseaux de capteurs sans fil et problèmes de l'énergie

1.1 Introduction

Grâce aux progrès dans les domaines de la microélectronique, des communications sans fil et de l'informatique embarquée, ces dernières années nous avons assisté à une miniaturisation du matériel informatique. Cette dernière a apporté une nouvelle génération de réseaux informatiques et télécoms présentant des défis technologiques importants à savoir : 'Les réseaux de capteurs sans fil'. De très nombreux états de l'art ont été proposés [8-12]. Nous allons représenter dans le présent chapitre le fonctionnement des réseaux de capteurs sans fil, les mécanismes et les principes proposés pour économiser de l'énergie, les applications potentielles de ces derniers et enfin nous traitons le cas particulier des réseaux de capteurs d'images.

La mise en œuvre de simples possibilités de traitement, de stockage, de détection et de communication dans des dispositifs miniature, à faible coût et leur intégration dans ce qu'on appelle des réseaux de capteurs sans fil ouvrent la porte à une multitude de nouvelles applications. Les réseaux de capteurs constituent une catégorie de réseaux sans fil, similaires aux réseaux mobiles ad-hoc [13], comportant un très grand nombre de nœuds. Ils sont également caractérisés entre autre par un déploiement très dense et à grande échelle dans des environnements souvent limités en terme de ressources [8]. Les réseaux de capteurs se forment automatiquement et s'adaptent aux changements d'état des nœuds. Ils sont donc capables de fonctionner sans aucune intervention humaine avec une grande tolérance aux défaillances, être déployés de manière aléatoire et opérer en milieu hostile ou difficile d'accès. Un réseau de capteurs est généralement constitué de nombreux nœuds repartis dans une zone appelée "sensor field" ou champ de capteurs. Ces nœuds sont reliés à une ou plusieurs passerelles, appelé "puits" ou bien "station de base (sink)". Elle assure l'interface entre le réseau de capteurs sans fil et l'utilisateur. L'interconnexion du puits à un réseau à infrastructure (Exemple : Internet par liaison satellite) permet les accès à distance pour les utilisateurs éloignés grâce notamment à l'interconnexion avec d'autres réseaux l'interconnexion avec d'autres réseaux (Internet, satellite . . .) et la récupération des données (Voir Fig. 1.1).

Les applications des réseaux de capteurs sans fil sont extrêmement variées [14] et ont connu un très grand succès. Ils détiennent un potentiel qui révolutionne de nombreux secteurs de notre économie et notre vie quotidienne. Citons à titre indicatif des applications de surveillance environnementale (études climatiques, suivi de la pollution des sols, de l'eau, de l'air, agriculture de précision), du domaine militaire (surveillance tactique terrestre et

maritime, cartographie et pistage de cibles), de la santé, la modernisation de la médecine, de l'agriculture, de la télématique et de la logistique [12], de la prévention des risques (feux de forêts, inondations, activité sismique ou volcanique, structure des grands ouvrages). Dans la majorité des applications, les ondes radio sont le moyen privilégié pour la communication entre capteurs. D'autres techniques de transmission sont toutefois possibles, notamment les ondes acoustiques qui sont bien appropriées dans les milieux sous-marins.

Parmi toutes les applications potentielles des réseaux de capteurs sans fil, celles utilisant des capteurs d'image sont appréciables pour tout ce qui concerne la reconnaissance, vigilance environnemental, la localisation, le dénombrement d'objets par la vision, la protection des zones sensibles. Dans un tel réseau, une partie des nœuds du réseau sont équipés de mini-caméras à technologie CMOS. le problème de la conservation de l'énergie devient plus important quand il s'agit des capteurs d'images. En effet, le volume de données est de plusieurs ordres de grandeur supérieur à une valeur scalaire classique (Exemple : température, humidité) [15].

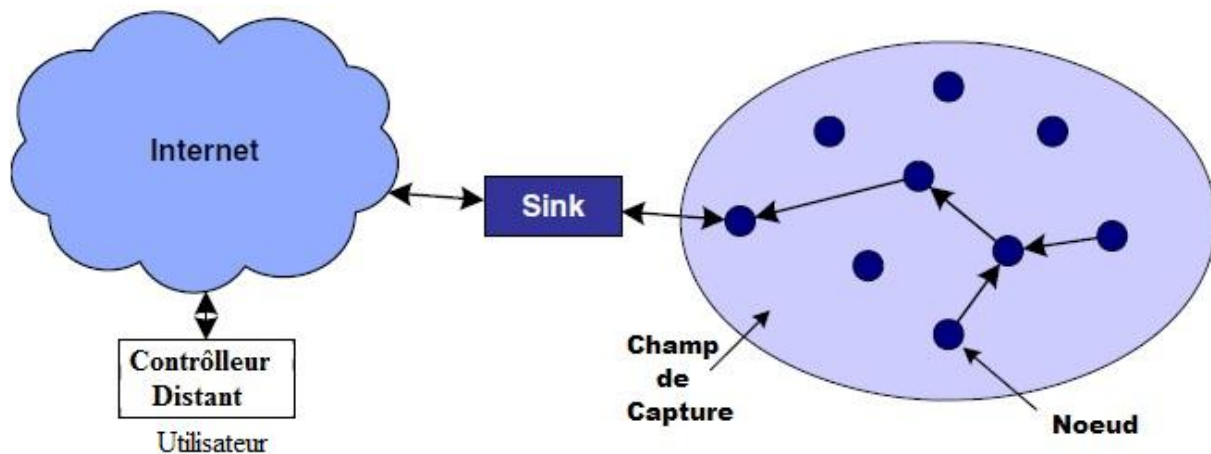


Figure 1.1. Architecture conventionnelle d'un réseau de capteurs sans fil

Ce chapitre vise à présenter un état de l'art des travaux traitant les réseaux de capteurs sans fil, de données scalaires et d'images, en mettant l'accent sur les mécanismes et les techniques de conservation de l'énergie. Il est structuré en six parties : la section 1.2 présente les caractéristiques de fonctionnement générales des réseaux de capteurs, indépendamment du type des informations collectées. La section 1.3 aborde les principes de base permettant d'améliorer leur efficacité énergétique, quel que soit le type de capture, et qui peuvent donc être utilisés dans le cas des capteurs d'images. La section 1.4 résume les principales techniques de conservation de l'énergie qui existent dans littérature. La section 1.5 traite des

exemples d'applications typiques des réseaux de capteurs sans fil. Le cas particulier des réseaux de capteurs d'image, les spécificités et les applications potentielles propres à ce type de réseau sont détaillés dans la section 1.6. Cette section permet de positionner notre contribution au problème de la conservation de l'énergie par rapport aux travaux faisant référence en compression et en transmission d'images.

1.2. Caractéristiques des réseaux de capteurs

1.2.1. Architecture d'un nœud capteur

Un nœud capteur est composé de plusieurs éléments ou modules correspondant chacun à une tâche particulière d'acquisition, de traitement, ou de transmission de données (Voir figure 1.2). Il comprend également une source d'énergie [9].

1. *L'unité d'acquisition des données* : le principe de fonctionnement des détecteurs est souvent le même : il s'agit de répondre à une variation des conditions d'environnement par une variation de certaines caractéristiques électriques (par exemple pour une thermistance, une variation de température entraîne une variation de la résistance). Les variations de tension sont ensuite converties par un convertisseur Analogique-Numérique afin de pouvoir être traitées par l'unité de traitement. On trouve aussi des structures plus complexes pour détecter d'autres phénomènes : les MEMS (pour Microelectromechanical systems) [8-9]. Ils sont utilisés pour une grande variété de phénomènes physiques (accélération, concentration chimique...).
2. *L'unité de traitement des données* : les microcontrôleurs utilisés dans le cadre de réseaux de capteurs sont à faible consommation d'énergie. Leurs fréquences sont assez faibles, moins de 10 MHz pour une consommation de l'ordre de 1 mW. Une autre caractéristique est la taille de leur mémoire qui est de l'ordre de 10 Ko de RAM pour les données et de 10 Ko de ROM pour les programmes [9], [12]. Cette mémoire consomme la majeure partie de l'énergie allouée au microcontrôleur, c'est pourquoi on lui adjoint souvent de la mémoire flash moins couteuse en énergie. Outre le traitement des données, le microcontrôleur commande également toutes les autres unités notamment le système de transmission.
3. *L'unité de transmission de données* : les composants utilisés pour réaliser la transmission sont des composants classiques. Ainsi on retrouve les mêmes problèmes que dans tous les réseaux sans-fil : la quantité d'énergie nécessaire à la transmission augmente avec la distance. Pour les réseaux sans-fil classiques (LAN, GSM) la consommation d'énergie est

de l'ordre de plusieurs centaines de milliwatts, et on se repose sur une infrastructure alors que pour les réseaux de capteurs, le système de transmission consomme environ 20 mW et possède une portée de quelques dizaines de mètres. Pour augmenter ces distances tout en préservant l'énergie, le réseau utilise un routage multi-sauts.

4. *La source d'énergie* : pour des réseaux de capteurs sans fil autonomes, l'alimentation est une composante cruciale. Il y a essentiellement deux aspects : premièrement, stocker l'énergie et la fournir sous la forme requise ; deuxièmement, tenter de reconstituer l'énergie consommée par un réapprovisionnement grâce à une source externe au nœud-capteur telles les cellules solaires. Le stockage de l'énergie se fait traditionnellement en utilisant ses piles. à titre indicatif, ce sera souvent une pile AA normale d'environ 2.2 – 2.5 Ah fonctionnant à 1.5 V [12].

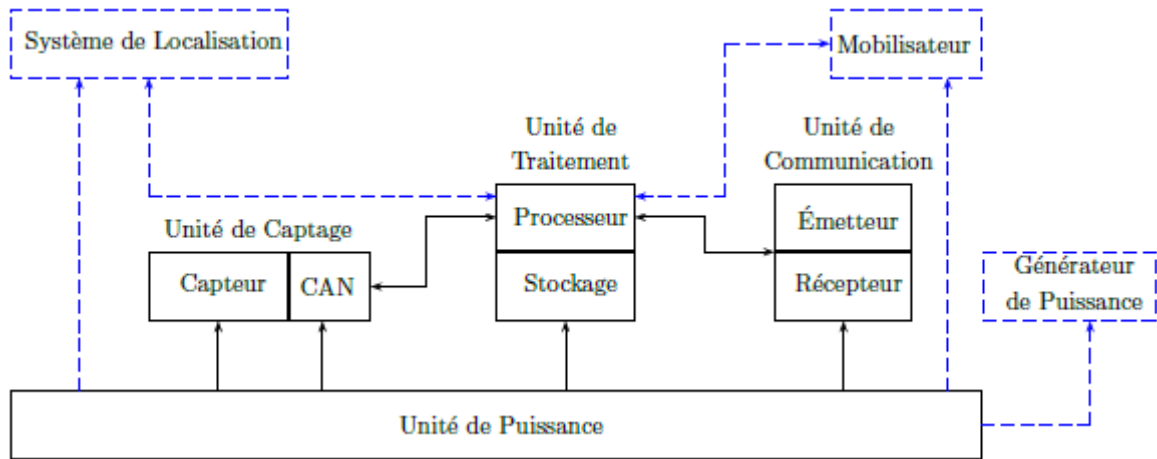


Figure 1.2. Architecture conventionnelle d'un nœud capteur

1.2.2. Les différents facteurs de conception

La conception des réseaux de capteurs est influencée par de nombreux facteurs comme la tolérance aux pannes, les coûts de production, la consommation d'énergie, l'environnement ou la topologie du réseau. Ces facteurs [9] représentent la base de la conception de protocoles ou d'algorithmes pour les réseaux de capteurs. Nous résumons ces principaux facteurs comme suit :

1. *Tolérance aux pannes* : Les nœuds peuvent être sujets à des pannes dues à leur fabrication (ce sont des produits de série bon marché, il peut donc y avoir des capteurs défectueux) ou plus fréquemment à un manque d'énergie. Les interactions externes (chocs, interférences)

peuvent aussi être la cause des dysfonctionnements. Afin que les pannes n'affectent pas la tâche première du réseau, il faut évaluer la capacité du réseau à fonctionner sans interruption.

2. *Coût de fabrication* : Les nœuds sont des produits fabriqués en série du fait de leur grand nombre. Il faut que le coût de fabrication de ces nœuds soit tel que le coût global du réseau ne soit pas supérieur à celui d'un réseau classique afin de pouvoir justifier son intérêt.
3. *Topologie du réseau* : En raison de leur forte densité dans la zone à observer, il faut que les nœuds-capteurs soient capables d'adapter leur fonctionnement afin de maintenir la topologie souhaitée.
4. *Consommation d'énergie* : L'économie d'énergie est une des problématiques majeures dans les réseaux de capteurs. En effet, la recharge des sources d'énergie est souvent trop coûteuse et parfois impossible. Il faut donc que les capteurs économisent au maximum l'énergie afin de pouvoir fonctionner. Les réseaux de capteurs fonctionnant selon un mode de routage par saut, chaque nœud du réseau joue un rôle important dans la transmission de données. Le mauvais fonctionnement d'un nœud implique un changement dans la topologie et impose une réorganisation du réseau.

1.2.3. Architecture protocolaire

La pile de protocoles utilisée par le puits (Sink) ainsi que par tous les nœuds-capteurs est donnée dans la Fig.1.3. Cette pile de protocoles combine routage et gestion d'énergie et intègre les données avec les protocoles réseau. Elle communique de manière efficace (en termes d'énergie) à travers le support sans fil et favorise les efforts de coopération entre les nœuds-capteurs.

La pile de protocoles comprend une couche application, une couche transport, une couche réseau, une couche liaison de données, une couche physique, un plan de gestion d'énergie, un plan de gestion de mobilité et un plan de gestion des tâches. Selon les tâches de détection, différents types de logiciels d'application peuvent être construits et utilisés dans la couche application. La couche transport contribue au maintien du flux de données si l'application du réseau de capteurs l'exige.

La couche réseau s'occupe de l'acheminement des données fournies par la couche transport. Comme l'environnement est sujet au bruit et que les nœuds-capteurs peuvent être mobiles, le protocole MAC doit tenir compte de la consommation d'énergie et doit être en mesure de

réduire les collisions entre les nœuds voisins lors d'une diffusion par exemple. La couche physique répond aux besoins d'une modulation simple mais robuste, et de techniques de transmission et de réception.

En outre, les plans de gestion d'énergie, de mobilité et des tâches surveillent et gèrent la consommation d'énergie, les mouvements, et la répartition des tâches entre les nœuds-capteurs. Ces plans aident les nœuds-capteurs à coordonner les tâches de détection et à réduire la consommation d'énergie globale.

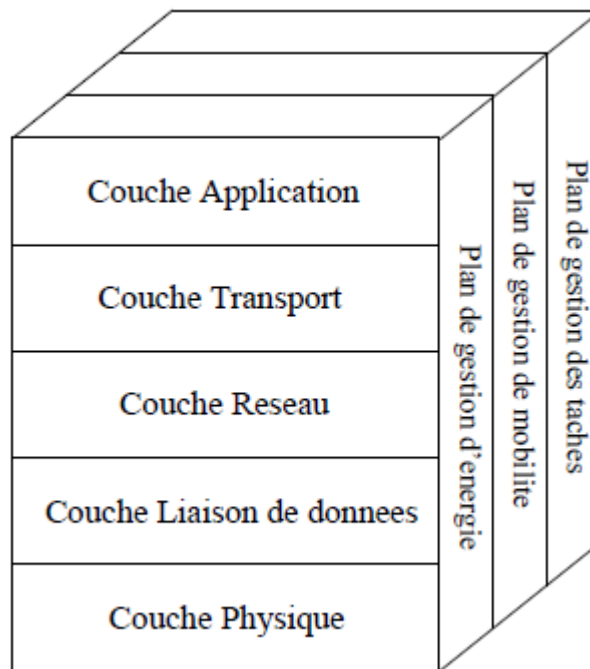


Figure 1.3. Pile de protocoles d'un réseau de capteurs sans fil

1.2.4. Consommation d'énergie d'un nœud capteur

Les nœuds-capteurs sont alimentés principalement par des piles embarquées. 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 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 [16]. Cette dissipation d'énergie est due principalement aux composants suivants:

1. *Le Microcontrôleur (MCU)*: Généralement les MCUs possèdent divers modes de fonctionnement : actif, "idle", et sommeil, à des fins de gestion d'énergie. Chaque mode est caractérisé par une quantité différente de consommation d'énergie. Par exemple, le MSP430 consomme 3 mW en mode actif, 99 μ W dans le mode "idle" et seulement 15 μ W dans le mode sommeil. Toutefois, la transition entre les modes de fonctionnement implique un surplus d'énergie et de latence. Ainsi, les niveaux de consommation d'énergie des différents modes, les coûts de transition entre les modes et encore le temps passé par le MCU dans chaque mode ont une incidence importante sur la consommation totale d'énergie d'un nœud-capteur.
2. *La radio* : la radio opère dans quatre modes de fonctionnement : émission, réception, "idle", et sommeil. Une observation importante dans le cas de la plupart des radios est que le mode "idle" induit une consommation d'énergie significative, presque égale à la consommation en mode réception [17]. Ainsi, il est plus judicieux d'éteindre complètement la radio plutôt que de passer en mode "idle" quand l'on a ni à émettre ni à recevoir de données. Un autre facteur déterminant est que, le passage de la radio d'un mode à un autre engendre une dissipation d'énergie importante due à l'activité des circuits électroniques. Par exemple, quand la radio passe du mode sommeil au mode émission pour envoyer un paquet, une importante quantité d'énergie est consommée pour le démarrage de l'émetteur lui-même [16]. Un autre point important est que les données des constructeurs sous-estiment assez régulièrement ces différentes consommations [18], en particulier concernant la consommation dans le mode "idle".
3. *Le détecteur ou le capteur proprement dit* : il y a plusieurs sources de consommation d'énergie par le module de détection, notamment l'échantillonnage et la conversion des signaux physiques en signaux électriques, le conditionnement des signaux et la conversion analogique-numérique. Etant donné la diversité des capteurs, il n'y a pas de valeurs typiques de l'énergie consommée. En revanche, les capteurs passifs (température, sismiques, ...) consomment le plus souvent peu d'énergie par rapport aux autres composants du nœud-capteur. Notons que les capteurs actifs tels que les sonars, les capteurs d'images, etc. peuvent consommer beaucoup d'énergie.

En outre, il existe d'autres formes de dissipation d'énergie telles que les lectures et les écritures mémoire. Un autre aspect non négligeable est le phénomène d'autodécharge de la batterie. En effet, cette dernière se décharge d'elle-même et perd de sa capacité au fil du temps.

Apporter une étude quantitative et comparative précise de la consommation de chaque composant d'un nœud-capteur est quasiment impossible en raison du grand nombre de plateformes commerciales existantes. Cependant, des expérimentations ont montré que c'est la transmission de données qui est la plus consommatrice en énergie [16]. Le coût d'une transmission d'un bit d'information est approximativement le même que le coût nécessaire au calcul d'un millier d'opérations [19]. Notons que la consommation du module de détection dépend du type spécifique du nœud-capteur.

1.2.5. Surconsommation d'énergie d'un nœud capteurs

Nous appelons surconsommation d'énergie toute consommation inutile que l'on peut éviter afin de conserver l'énergie d'un nœud-capteur. Les sources de cette surconsommation sont nombreuses, elles peuvent être engendrées lors de la détection lorsque celle-ci est mal gérée (par exemple par une fréquence d'échantillonnage mal contrôlée) [20].

La surconsommation concerne également la partie communication. En effet, cette dernière est sujette à plusieurs phénomènes qui surconsomment de l'énergie surtout au niveau MAC où se déroule le contrôle d'accès au support sans fil. Certains de ces phénomènes sont les causes majeures de la perte d'énergie et ont été recensés dans [12],[21] comme suit:

1. *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. Tous les protocoles MAC essaient à leur manière d'éviter les collisions. Les collisions concernent plutôt les protocoles MAC avec contention.
2. *L'écoute à vide (idle listening)* : un nœud dans l'état "idle" 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.

3. *L'écoute abusive (overhearing)* : cette situation se présente quand un nœud reçoit des paquets qui ne lui sont pas destinés. 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, particulièrement dans les réseaux "mostly-on" [22].
4. *L'overmitting* : un nœud envoie des données et le nœud destinataire n'est pas prêt à les recevoir.
5. *L'overhead des paquets de contrôle* : l'envoi, la réception, et l'écoute des paquets de contrôle consomment de l'énergie. Comme les paquets de contrôle ne transportent pas directement des données, ils réduisent également le débit utile effectif.

1.2.6. Durée de vie d'un réseau de capteurs sans fil

Dans un réseau de capteurs sans fil, la durée de vie est la métrique la plus importante dans l'évaluation des performances. En effet, dans un environnement contraint, l'objectif principal consiste à déployer des nœuds dans une zone sans surveillance pendant des mois ou des années. La durée de la vie du réseau, comme mesure de la consommation d'énergie, occupe une place exceptionnelle puisqu'elle constitue la borne supérieure de l'utilité de ce réseau [23]. Maximiser la durée de de vie du réseau revient à réduire la consommation énergétique des nœuds.

La durée de vie est également considérée comme un paramètre fondamental dans un contexte de disponibilité et de sécurité dans les réseaux de capteurs sans fil [24].

Malgré les progrès qui ont été fait, la durée de vie de ses dispositifs miniatures, à batteries embarquées, continue d'être un défi majeur et un facteur clé, exigeant davantage de recherches sur l'efficacité énergétique des plates-formes et des protocoles de communication.

Il existe plusieurs définitions pour la durée de vie d'un réseau de capteurs. Toutes ces définitions dépendent de l'application ciblée du réseau de capteurs sans fil. En générale, on peut classer ces définitions en trois grandes catégories :

1. Les définitions basées sur le nombre des capteurs vivants [12], [25-26].
2. Les définitions basées sur la connectivité [27-28].
3. Les définitions basées sur la couverture [29-30].

Il convient de noter que la simulation de la durée de vie du réseau peut être un problème statistique difficile. De toute évidence, la durée de vie d'un réseau de capteurs sans fil dépend principalement de la durée de vie de chaque nœud et plus ces durées sont longues, meilleur est le fonctionnement du réseau.

1.3. Mécanismes de conservation de l'énergie

Comme nous l'avons souligné dans la section 1.2.5, c'est la transmission de données qui se révèle extrêmement consommatrice par rapport aux tâches du nœud-capteur. Cette caractéristique conjuguée à l'objectif de maximisation de la durée de vie du réseau a suscité de nombreux travaux de recherche. Avant de citer ces travaux dans les sections suivantes, nous introduisons dans ce qui suit les mécanismes de base de conservation de l'énergie:

1. *Mode d'économie d'énergie* : ce mode est possible quelle que soit la couche MAC adoptée. Cela consiste à éteindre le module de communication dès que possible. Par exemple, des protocoles MAC fondés sur la méthode TDMA (Time Division Multiple Access) offrent une solution implicite puisqu'un nœud n'échange des messages que dans les intervalles de temps qui lui sont attribués. Il peut alors garder sa radio éteinte durant les autres slots. Comme nous l'avons souligné précédemment (section 1.2.5), il faut toutefois veiller à ce que le gain d'énergie obtenu en mettant en veille le module radio ne soit pas inférieur au surcoût engendré par le redémarrage de ce module.
2. *Traitement local* : L'idée de cette technique est que la source peut se censurer. Ainsi une programmation événementielle semble bien adaptée aux réseaux de capteurs. Seuls les changements significatifs de l'environnement devraient provoquer un envoi de paquets au réseau. Dans le même état d'esprit, une grande collaboration est attendue entre les capteurs d'une même région en raison de leur forte densité et dans la mesure où les observations ne varient presque pas entre des voisins très proches. Ainsi les données pourront être confrontées localement et agrégées au sein d'un seul et unique message. Cette stratégie de traitement local permet de réduire sensiblement le trafic.
3. *Organisation des échanges* : ce procédé revient à limiter les problèmes de retransmission dus aux collisions. La solution extrême consiste à utiliser la technique d'accès au médium TDMA. Les collisions sont ainsi fortement réduites. Cette solution présente l'inconvénient

d'être peu flexible et de demander une synchronisation fine des capteurs. Des solutions intermédiaires ont vu le jour, par exemple S-MAC (Sensor MAC) [21] qui est une méthode d'accès au canal de type CSMA-CA avec le mécanisme RTS/CTS (Request to Send, Clear to Send) qui permet d'éviter les collisions et le problème de la station cachée. La principale innovation, apportée par ce protocole, est d'avoir un mécanisme de mise en veille distribué sur chaque nœud du réseau dans le but de réduire la consommation d'énergie. La principale difficulté de S-MAC est également de synchroniser les nœuds entre eux pour que la communication soit toujours possible.

4. *Limitation des accusés de réception* : l'acquittement systématique est mal adapté à des réseaux denses : il provoque une surcharge du réseau et donc des collisions et des interférences avec les données utiles échangées dans le réseau. Les acquittements par "piggy-backing" seront à privilégier.
5. *Répartition de la consommation d'énergie* : la formation de "clusters" permet d'envisager des réseaux comportant un très grand nombre de capteurs. Elle favorise une meilleure répartition de la consommation d'énergie. En effet, dans le cas d'une transmission directe vers l'observateur, les capteurs éloignés vont plus rapidement manquer d'énergie et les autres nœuds peuvent être sujets au phénomène d'overhearing dans le cas des réseaux "Mostly-On".

Au contraire, dans le cas d'une transmission par saut, les nœuds proches de l'observateur vont être vite en rupture de batterie car ils seront plus sollicités pour relayer les messages des autres. La solution consiste à hiérarchiser les échanges en divisant la zone d'observation en clusters. Un "clusterhead" est élu pour chaque cluster. Il s'occupe de récupérer les informations auprès des capteurs de son cluster et de les transmettre directement à l'observateur. En changeant régulièrement de clusterhead, on obtient un réseau dans lequel aucun capteur n'est prédisposé à arriver en rupture de batterie avant les autres. Mettre en place des clusters va également permettre de cloisonner le réseau et ceci dans l'objectif de réduire les interférences. On améliore ainsi la qualité du lien radio et par conséquent, on limite les retransmissions liées aux reprises sur erreur. L'exemple typique d'une solution avec des clusters est le protocole LEACH [31].

1.4. Techniques de conservation de l'énergie

Des mesures expérimentales ont montré que, généralement, c'est la transmission des données qui est la plus consommatrice en énergie, et de façon significative, les calculs, eux, consomment très peu [16]. La consommation d'énergie du module de détection dépend de la spécificité du capteur. Dans de nombreux cas, elle est négligeable par rapport à l'énergie consommée par le module de traitement et, par-dessus tout, le module de communication.

Dans d'autres cas, l'énergie dépensée pour la détection peut être comparable, ou supérieure à celle nécessaire à la transmission des données (Cas des capteurs d'images). En général, les techniques d'économie d'énergie se concentrent sur deux parties : la partie réseau (i.e., la gestion d'énergie est prise en compte dans les opérations de chaque nœud, ainsi que dans la conception de protocoles réseau), et la partie local (i.e. des techniques sont utilisées pour réduire le nombre ou la fréquence d'échantillonnage dans les nœuds).

La durée de la vie d'un réseau de capteurs peut être prolongée par l'application conjointe de différentes techniques [32]. Par exemple, les protocoles efficaces en énergie visent à réduire au minimum la consommation d'énergie pendant l'activité du réseau. Toutefois, une quantité considérable d'énergie est consommée par les composants d'un nœud (CPU, radio, etc), même s'ils sont inactifs. Un plan de gestion dédié à l'énergie peut alors être utilisé pour éteindre temporairement les composants du nœud lorsqu'ils ne sont pas sollicités.

Il existe plusieurs techniques de conservation de l'énergie, dédiées principalement aux réseaux de capteurs sans fil. Nous pouvons les résumer comme suit :

1.4.1. Techniques de Duty cycling

Cette technique est principalement utilisée dans l'activité réseau. Le moyen le plus efficace pour conserver l'énergie est de mettre la radio de l'émetteur en mode veille (low-power) à chaque fois que la consommation n'est pas nécessaire. Idéalement, la radio doit être éteinte dès qu'il n'y a plus de données à envoyer et/ou recevoir, et devrait être prête dès qu'un nouveau paquet de données doit être envoyé ou reçu. Ainsi, les nœuds alternent entre périodes actives et sommeil en fonction de l'activité du réseau. Ce comportement est généralement dénommé Duty-cycling. Un Duty-cycle est défini comme étant la fraction de temps où les nœuds sont actifs.

Comme les nœuds-capteurs effectuent des tâches en coopération, ils doivent coordonner leurs dates de sommeil et de réveil. Un d'ordonnement Sommeil/Réveil accompagne donc tout plan de Duty-cycling. Il s'agit généralement d'un algorithme distribué reposant sur les dates auxquelles des nœuds décident de passer entre l'état actif et l'état sommeil. Il permet aux nœuds voisins d'être actifs en même temps, ce qui rend possible l'échange de paquets, même si les nœuds ont un faible duty-cycle (c-à-d ils dorment la plupart du temps).

En générale, deux grandes catégories de protocoles de 'Duty cycle' existent dans la littérature. La première catégorie englobe les protocoles de 'Sleep/Wakeup' [33-34]. La deuxième catégorie englobe tous les protocoles de niveau MAC. De nombreux état de l'art sont disponibles dans la littérature [35-37].

1.4.2. Techniques de mobilité

Dans certains cas où les nœuds sont mobiles, la mobilité peut être utilisée comme outil pour réduire la consommation d'énergie (au-delà du duty-cycling et des techniques orientées données). Dans un réseau de capteurs statiques, les paquets provenant des nœuds suivent des chemins multi-sauts vers la station de base. Ainsi, certains chemins peuvent être chargés (sollicités plus que d'autres), et les nœuds proches de la station de base relayent plus de paquets [38] et sont plus sujets à l'épuisement prématuré de leurs batteries (*funneling effet*) [39]. Si certains nœuds (éventuellement, la station de base) sont mobiles, le trafic peut être modifié si les nœuds mobiles sont chargés de collecter des données directement à partir de nœuds statiques.

Les nœuds attendent le passage d'un dispositif mobile pour lui envoyer leurs messages de telle sorte que la communication ait lieu à proximité (directement ou au plus avec un nombre limité de sauts). Par conséquent, les nœuds ordinaires peuvent économiser de l'énergie parce que la longueur du chemin, la contention et les overheads de diffusion sont ainsi réduits. En outre, le dispositif mobile peut visiter le réseau afin de répartir uniformément la consommation d'énergie due à la communication. Lorsque le cout de la mobilité des nœuds de capteurs est prohibitif, l'approche classique consiste à attacher un capteur à des entités qui seront en itinérance dans le champ de détection, comme des autobus ou des animaux.

Les stratégies reposant sur la mobilité peuvent être classées en deux ensembles : les stratégies avec un sink mobile et les stratégies avec des relais mobiles, selon le type de l'entité mobile. Il est important de souligner ici que, lorsque nous examinons des systèmes mobiles, un problème

important est le type de contrôle de mobilité des nœuds qu'intègre la conception du réseau, ceci est détaillé dans [40]. Les nœuds mobiles peuvent être divisés en deux catégories : ils peuvent être spécifiquement conçus comme partie de l'infrastructure du réseau, ou faire partie de l'environnement.

Quand ils font partie de l'infrastructure, leur mobilité peut être entièrement contrôlée dans la mesure où ils sont, généralement, robotisés. Lorsque les nœuds mobiles font partie de l'environnement, ils pourraient ne pas être contrôlables. S'ils suivent un horaire strict, ils ont une mobilité complètement prévisible (par exemple, une navette pour les transports publics [41]). Sinon, ils peuvent avoir un comportement aléatoire de sorte qu'aucune hypothèse ne puisse être faite sur leur mobilité.

Enfin, ils peuvent suivre un schéma de mobilité, qui n'est ni prévisible, ni totalement aléatoire. Par exemple, c'est le cas d'un bus se déplaçant dans une ville, dont la vitesse est soumise à l'importante variation en raison des conditions de circulation. Dans un tel cas, les schémas de mobilité peuvent être tirés en se fondant sur des observations et des estimations d'une certaine précision.

1.4.3. Techniques orientées données

Généralement, les deux techniques citées précédemment à savoir : les techniques de "Duty-cycling" et les techniques de mobilité, ne tiennent pas compte des données prélevées par les nœuds. Par conséquent, des approches orientées données peuvent être utiles pour améliorer l'efficacité en énergie. En fait, la détection (ou prélèvement de données) affecte la consommation d'énergie de deux manières :

- *Des échantillons inutiles* : les données échantillonnées ont souvent de fortes corrélations spatiales et/ou temporelle [42], il est donc inutile de communiquer les informations redondantes à la Station de Base. Un échantillonnage inutile implique une consommation d'énergie à son tour inutile. En effet, même si le coût de l'échantillonnage est négligeable, cela induit aussi des communications tout le long de chemin qu'emprunte le message.
- *La consommation électrique du module de détection* : réduire la consommation ne suffit pas lorsque le capteur est lui-même très consommateur.

Des techniques orientées données sont conçues pour réduire la quantité d'échantillons de données en garantissant un niveau de précision acceptable dans la détection pour l'application.

1-Réduction des données

Réduire les données en termes de volume ou de nombre de paquets, dans le réseau peut avoir un impact majeur sur la consommation d'énergie due à la communication. Parmi les méthodes de réductions de données, nous trouvons le *In-network processing* qui consiste à réaliser de l'agrégation de données (par exemple, calculer la moyenne de certaines valeurs) au niveau des nœuds intermédiaires entre la source et le *sink*. Ainsi, la quantité de données est réduite tout en parcourant le réseau vers le *sink*. Une agrégation de données appropriée est spécifique à l'application. Un état de l'art sur les techniques *In-network processing* est disponible dans [43].

La compression de données peut être appliquée également pour réduire la quantité d'informations au niveau des nœuds qui engendrent des données, et le décodage au niveau de Sink. Un état de l'art sur les techniques de compression est disponible dans [44].

2-Acquisition de données efficaces en énergie

De nombreuses applications émergentes ont d'application a de réelles contraintes dues à la détection. Ceci va à l'encontre de l'hypothèse générale selon laquelle la détection n'est pas significative d'un point de vue consommation d'énergie. En fait, la consommation d'énergie du module de détection peut, non seulement être significative, mais encore supérieure à la consommation d'énergie de la radio ou même plus grande que la consommation d'énergie du reste du nœud capteurs [20]. Cela peut être dû à différents facteurs [45] :

- *Transducteur gourmand en énergie* : Certains capteurs ont intrinsèquement besoin d'une forte puissance pour s'acquitter de leur tâche d'échantillonnage. Par exemple, des capteurs d'images CMOS, voire des capteurs multimédia [15] ont généralement besoin de beaucoup d'énergie. Les capteurs chimiques ou biologiques peuvent aussi être gourmands en énergie.
- *Convertisseur A/D gourmands* : des capteurs tels que les transducteurs acoustiques et sismiques [46] nécessitent généralement des convertisseurs A/D à haut débit et à grande résolution. La consommation d'électricité des convertisseurs représente la part la plus importante de la consommation d'énergie du sous-système de de détection.
- *Capteurs actifs* : une autre classe de capteurs peut obtenir des données du phénomène perçu par l'utilisation de transducteurs actifs (par exemple, sonar, radar ou laser). Dans ce

cas, les capteurs doivent envoyer un signal de sondage afin d'obtenir des informations sur la grandeur observée, comme dans [47].

- *Temps d'acquisition long* : Le temps d'acquisition peut être de l'ordre de plusieurs centaines de millisecondes, voire de quelques. Par conséquent, l'énergie consommée par le sous système de détection peut être élevé, même si la consommation d'énergie du détecteur reste modérée.

Dans ce cas, réduire les communications peut s'avérer insuffisant, mais les stratégies de conservation d'énergie doivent réellement réduire le nombre d'acquisition (échantillons de données). Il faudrait également préciser les techniques d'acquisition de données efficaces en énergie ne visent pas exclusivement à réduire la consommation d'énergie du module de détection. En réduisant les données prélevées par des nœuds sources, elles diminuent aussi le nombre de communications. En fait, beaucoup de techniques d'acquisition de données efficaces en énergie ont été conçues pour réduire au minimum l'énergie consommée par radio, en supposant que la consommation de la radio est négligeable.

La classification des approches d'acquisition de données efficaces en énergie présentée dans [45] est comme suit :

- *Echantillonnage adaptatif* : Comme les échantillons mesurés peuvent être corrélés, les techniques d'échantillonnage adaptatif exploitent de telles similitudes pour réduire la quantité de données à acquérir par le transducteur. Par exemple, les données intéressantes peuvent changer lentement en fonction du temps. Dans ce cas, des corrélations temporelles peuvent être exploitées pour réduire le nombre d'acquisition. Une approche semblable peut être appliquée lorsque le phénomène étudié ne change pas brusquement entre les régions couvertes par des nœuds voisins. L'énergie due au prélèvement (et à la communication) peut être alors réduite en profitant des corrélations spatiales entre les données prélevées. Clairement, des corrélations temporelles et spatiales peuvent être conjointement exploitées pour réduire sensiblement la quantité de données à acquérir.
- *Echantillonnage hiérarchique* : L'approche d'échantillonnage hiérarchique suppose que les nœuds sont équipés de sondes (ou détecteurs) de différents types. Alors que chaque sonde est caractérisée par une résolution donnée et sa consommation d'énergie associée, cette technique choisit dynamiquement la classe à activer, afin d'obtenir un compromis entre la précision et l'économie d'énergie.

- *Echantillonnage actif* : l'échantillonnage actif fondé sur un modèle adopte une approche semblable à la prévision de donnée. Un modèle du phénomène mesuré est établi lors des prélèvements de données, de telle sorte que les valeurs futures puissent être prévues avec une certaine précision. Cette approche exploite le modèle obtenu pour réduire le nombre d'échantillons de données, et également la quantité de données à transmettre à la Station de Base bien que ce ne soit pas leur objectif principal.

1.5. Applications des réseaux de capteurs scalaires

Plusieurs applications envisagées dans les réseaux de capteurs font toujours l'objet d'une recherche et d'un développement actifs universitaires ou industriels [11]. On trouve des applications pour la détection et la surveillance des désastres, le contrôle de l'environnement et la cartographie de la biodiversité, le bâtiment intelligent, l'agriculture de précision, la surveillance et la maintenance préventive des machines, la médecine et la santé, la logistique et la télématique, etc. Nous décrivons ici brièvement quelques applications dans certains domaines pour donner une idée de cette étendue.

1.5.1. Exemples d'applications typiques

- *Supervision de l'habitat écologique* : Les études scientifiques des habitats écologiques (animaux, végétaux, micro-organismes) sont traditionnellement effectuées grâce à des activités sur le terrain par des enquêteurs. Un problème majeur dans ces études provient de ce qui est parfois appelé "l'effet de l'observateur". En effet, la présence et les activités potentiellement intrusives des enquêteurs sur le terrain peuvent affecter le comportement des organismes dans l'habitat supervisé et ainsi fausser les résultats des observations. Des réseaux de capteurs sans fil sans surveillance promettent une nouvelle approche écologique d'observation à distance pour la surveillance de l'habitat. En outre, les réseaux de capteurs, en raison de leur grande échelle potentielle et d'une haute densité spatio-temporelle, peuvent fournir des données expérimentales d'une richesse sans précédent.

Un des premiers déploiements expérimental de réseaux de capteurs sans fil pour la surveillance de l'habitat s'est déroulé, sur Great Duck Island dans le golfe du Maine [48]. Une équipe de chercheurs du laboratoire de recherche d'Intel à Berkeley, de l'Université de Californie à Berkeley, et du Collège de l'Atlantique à Bar Harbor ont déployé des nœuds-capteurs sans fil dans et autour des terriers de "L'océanité culblanc", un oiseau

qui vit en grandes colonies sur l'île, au cours de la saison de reproduction. Les données transmises par le réseau de capteurs ont été mise à disposition sur le web, via une station de base sur l'île reliée à un lien de communication par satellite.

- *Surveillance militaire et traque de cible* : Comme beaucoup d'autres technologies de l'information, les réseaux de capteurs sans fil proviennent principalement de la recherche militaire. Des réseaux de capteurs autonomes sont envisagés comme l'ingrédient essentiel dans cette lancée vers des systèmes de guerre centrés sur les réseaux. Ils peuvent être rapidement déployés et utilisés pour la surveillance des champs de bataille afin de fournir des renseignements concernant l'emplacement, le nombre, le mouvement, et l'identité des soldats et des véhicules, ou bien encore pour la détection des agents chimiques, biologiques et nucléaires.

Une grande partie de la croissance rapide dans la recherche et le développement des réseaux de capteurs sans fil a été apportée par des programmes financés par l'Agence américaine pour les Projets de Recherche Avancée de Défense (DARPA pour Defense Advanced Research Projects Agency), notamment grâce à un programme connu sous le nom de "SensIT" (Sensor Information Technology) [49] de 1999 à 2002.

- *Supervision des structures et des phénomènes sismiques* : Une autre classe d'applications pour réseaux de capteurs concerne le suivi de l'état des structures civiles. Ces structures peuvent être des bâtiments, des ponts et des routes, voire des aéronefs. A l'heure actuelle, la sûreté de ces structures est principalement apportée par le biais d'inspections manuelles ou visuelles ou occasionnellement par des technologies onéreuses en temps et en argent, telles que les rayons X et les ultrasons. Des techniques de détection réseau permettent d'automatiser le processus, en fournissant en temps opportun de riches informations sur un début de fissure ou d'autres dommages structuraux [50].

Les chercheurs envisagent le déploiement de ces capteurs avec une forte densité sur les structures, en les intégrant dans le matériau de construction comme le béton, ou en les mettant sur la surface. Ces réseaux de capteurs ont un potentiel pour le suivi à long terme de l'usure des structures ainsi que de leur état après des événements destructeurs, comme les tremblements de terre ou des explosions. Une vision futuriste, particulièrement pour

l'utilisation des réseaux de capteurs, implique le développement de structures contrôlables, qui contiennent des actionneurs qui réagissent en temps réel aux informations du capteur pour effectuer une "Annulation-d'écho" sur les ondes sismiques pour que la structure ne soit affectée par aucune perturbation externe.

- *Détection en réseau dans l'industrie et le commerce* : Dans la fabrication industrielle, des capteurs et des actionneurs sont utilisés pour les processus de suivi et de contrôle. Par exemple, dans une usine de traitement chimique à plusieurs étapes, il peut y avoir des capteurs placés en différents points dans le processus afin de surveiller la température, la concentration chimique, la pression, etc.

Les informations de cette surveillance en temps réel peuvent être utilisées pour diversifier les processus de contrôle, tels que l'ajustement des quantités d'un ingrédient particulier ou encore pour modifier les paramètres de chaleur. Le principal avantage de la création des réseaux de capteurs sans fil dans ces milieux est qu'ils peuvent améliorer de manière significative à la fois le coût et la souplesse inhérente à l'installation, mais encore l'entretien et la modernisation des systèmes filaires [51]. Il convient de noter qu'il existe déjà plusieurs sociétés de développement et de commercialisation de ces produits à l'image des technologies standards telles que le standard IEEE 902.15.4 [52], et de la collaboration industrielle telle que l'Alliance Zigbee [53].

1.6. Les réseaux de capteurs d'images

Le développement des micro-caméras et microphones a observé une forte évolution au cours de la dernière décennie, avec les évolutions des téléphones mobiles. 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 du signal. Jusqu'à il y a quelques années, l'usage d'un appareil photo impliquait la connexion d'un périphérique attaché au téléphone mobile, ou l'augmentation considérable de la taille de l'appareil. Aujourd'hui, nous trouvons ces micro-caméras embarquées dans pratiquement tous les téléphones cellulaires et les assistants numériques personnels, sans augmentation significative du coût de l'équipement, de son poids et de sa forme.

Les réseaux 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 nœuds sans fil. Cela a permis d'envisager concrètement un nouveau type d'applications utilisant des **“réseaux de capteurs sans fil multimédia”** [15][54].

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 :

1. *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 nœuds de capteurs sans fil malgré que la consommation énergétique reste considérable.
2. *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 nœuds 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. Notons que, 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.

1.6.1. Spécificités des réseaux de capteurs d'images

Bien évidemment, les travaux sur des images sont différents des travaux sur des données scalaires plus simples, comme ceux des premiers réseaux de capteurs. Ces différences sont dues évidemment 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, 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.

1. *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 photosensibles 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.
2. *Besoins de mémoire* : Comme nous l'avons dit, tandis que pour le codage d'un signal simple sollicite quelques bits d'information (de 1 à 9 octets, en fonction de la précision du capteur), 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. En effet, une image de 129×129 pixels utilisera en principe 4 fois plus de mémoire qu'une image de 64×64 . Maintenant, en fonction du format, une image peut être en noir et blanc, en niveaux de gris ou en couleur [55]. En principe, le format définit le nombre de bits nécessaires pour coder un pixel (une intensité capturée par l'un des photo-capteurs). Une image en niveaux de gris est normalement codée sur 9 bits par pixel (désigné 9bpp), même si cette règle n'est pas obligatoire. Pour coder un pixel en couleur, nous pouvons le faire sur trois plans en utilisant, soit un codage RGB (Red, Green, Bleu), soit un codage YCrCb. Ceci implique normalement l'utilisation d'un octet par plan de couleur (24bpp).
3. *Traitement du signal* : Dans les applications traditionnelles de vision, il est commun de vouloir faire des traitements sur les images à la source, afin d'extraire une information (par exemple : l'emplacement ou la classification d'un objet), ou de compresser l'image afin de diminuer la quantité de données nécessaires pour la représenter. Alors que ces traitements sont aisés à mettre en œuvre dans des dispositifs informatiques dotés de beaucoup de ressources, comme les ordinateurs portables et les assistants numériques personnels, 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.

4. *Transmission de données* : Comme le transmetteur radio est l'un des composants les plus gourmands en énergie, les protocoles de communication ont un rôle important à jouer pour faire des économies d'énergie. Dans les applications traditionnelles (par exemple : la température ou le mouvement), on peut envisager d'enregistrer plusieurs mesures et les embarquer dans un seul paquet pour augmenter le rendement de la communication. Une des techniques les plus utilisées est la fusion de données. Cela est possible parce que les mesures des différents capteurs sont généralement codées sur peu de bits et nous pouvons créer de paquets combinant des informations provenant de plusieurs sources. Dans le cas des images, la fusion de données n'est plus possible puisque les images sont transmises sur plusieurs centaines ou même milliers de paquets. Toutefois, les images naturelles ont des corrélations spatiales assez marquées et par conséquent la transmission d'images (et cela se produit également avec la transmission de la voix) offre une certaine tolérance aux pertes de paquets. En effet, on peut reconstruire une version approximative de l'image originale même si une partie des informations est perdue dans le réseau. Ces corrélations spatiales sont exploitées dans les algorithmes de compression, mais en contrepartie, les images compressées perdent leur tolérance aux pertes de paquets.

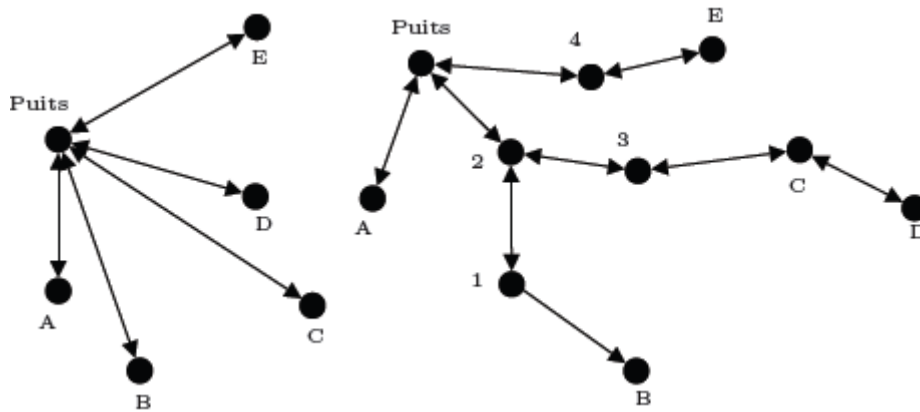
1.6.2. Applications des réseaux de capteurs d'image

Malgré le peu de temps qui s'est écoulé depuis l'émergence du domaine de recherche des réseaux de capteurs sans fil, il existe aujourd'hui un nombre significatif de travaux traitant des capteurs d'image incluant le développement de composants matériels et logiciels, de protocoles de communication (routage, contrôle de flux, . . .), et de traitements sur les données (compression, . . .). Les réseaux de capteurs d'image concernent toutes les applications qui touchent à la détection, la localisation, le dénombrement et le pistage d'objets par la vision. Dans cette section, nous allons classifier ces applications de manière générique et en présenter quelques-unes parmi les plus représentatives qui ont été expérimentés jusqu'à aujourd'hui.

Types d'applications

Nous pouvons distinguer deux grandes familles d'application pour les réseaux de capteurs de vision en fonction de l'architecture du réseau considérée [56]: réseaux à un saut (*single-hop network*) et réseau multi sauts (*multi-hop network*). Cette classification peut être généralisée pour tous les réseaux de capteurs sans fil. Dans le premier cas, un réseau de capteurs de vision est conçu comme un ensemble de nœuds caméra (et éventuellement d'autres types de capteurs) qui communiquent directement avec le puits comme montré à la figure 1.4 (a). Un

tel réseau a une couverture géographique limitée à la portée de communication du puits, quelques mètres à quelques centaines de mètres. Dans le deuxième cas, le réseau a une couverture géographique beaucoup plus étendue puisque les nœuds éloignés du puits vont pouvoir transporter leur information en passant par un ou plusieurs nœuds intermédiaires, comme montré dans la figure 1.4 (b).



(a) Réseau de capteur à un saut

(b) Réseau de capteur multi-sauts

Figure 1.4. Classification des réseaux de capteurs d'images en fonction de leur architecture de communication.

La disposition et le type de nœuds utilisé peut varier considérablement. On peut trouver des applications qui ne contiennent que les nœuds de capture d'images, mais il existe également des applications associant des nœuds de différents types qui se complètent mutuellement. Pour une application de détection des feux de forêt par exemple, on peut associer de capteurs de température et des capteurs d'image, les premiers fournissent l'évènement déclenchant la prise d'image par les seconds. Ces nœuds peuvent éventuellement être regroupés fonctionnellement, comme suggéré par les travaux de [57-58] sur l'établissement des architectures multicouches.

De même, les applications de capteurs d'images peuvent aussi être classifiées selon les modèles de surveillance ou de collecte des données des réseaux de capteurs traditionnels :

1. Surveillance périodique
2. Surveillance à la demande
3. Surveillance sur déclenchement d'évènement
4. Surveillance hybride

Les principes de ces classes d'application sont schématisés sur la figure 1.5

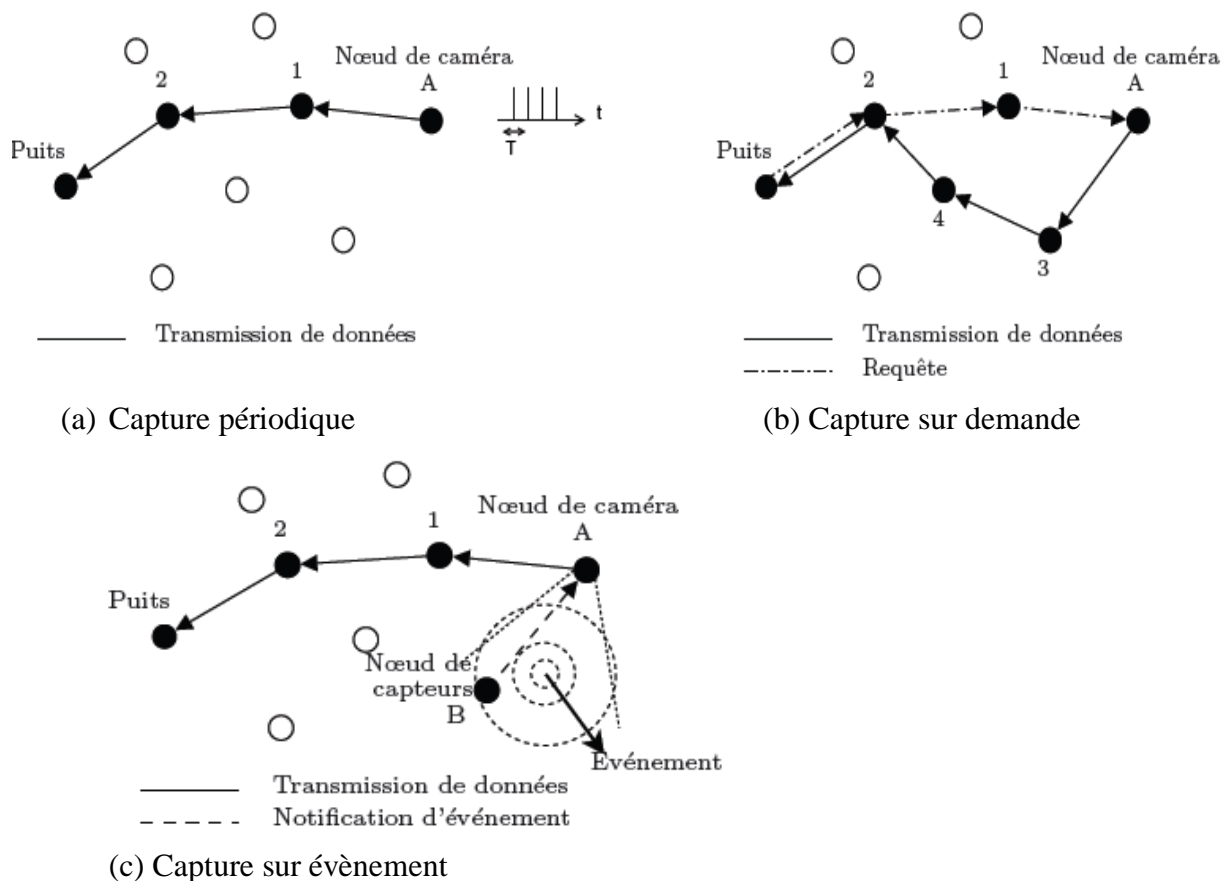


Figure 1.5 Classes d'application des réseaux de capteurs d'images

Les applications événementielles peuvent elles-mêmes être divisées en deux catégories en fonction du type d'évènement :

- Evènements internes : Les évènements sont détectés par le nœud de capture d'image.
- Evènements externes : Les évènements sont des messages reçus d'autres nœuds, qui notifient ainsi qu'un certain phénomène (l'évènement) s'est produit dans leur zone de perception.

Dans les applications événementielles, des évènements peuvent être détectés grâce à des capteurs de type scalaire (comme la température, la pression, les vibrations, etc), qui sont alors utilisés pour détecter un phénomène. Ils peuvent aussi être détectés par des capteurs d'image qui vont alors appliquer un algorithme (reconnaissance de contours, de couleurs, . . .), en vue d'identifier des informations intéressantes dans les images prises.

Exemples d'application

Il existe de nombreuses possibilités d'application des réseaux de capteurs d'image dans des scénarios réels. Elles concernent principalement les applications militaires, la surveillance environnementale, la sûreté et la sécurité des sites industriels, la surveillance des réseaux routiers et l'aide au déplacement de mobiles autonomes.

- *Applications militaires* : Les réseaux de capteurs de vision peuvent être très utiles pour l'espionnage militaire et la surveillance des champs de bataille. Dans l'expérience menée à la base McDill aux USA, « Une ligne dans le sable »[59], un réseau de capteurs de vision pourrait être utilisé pour la reconnaissance et la classification des cibles, par exemple. Le déploiement d'un réseau de capteurs sur les champs de bataille peut être réalisé manuellement ou aléatoirement. Dans le premier cas, des troupes de reconnaissance peuvent marcher dans le champ de bataille pour positionner et cacher stratégiquement des capteurs d'image.

Comme les capteurs sont, par définition, des dispositifs très petits, ils seront à priori faciles à dissimuler. L'orientation des caméras doit être effectuée rigoureusement si on veut couvrir visuellement toute la zone à observer. Cela peut être un problème lorsque le réseau de capteurs est déployé aléatoirement, par exemple en larguant les capteurs depuis un avion ou un drone. Comme les caméras ont un angle de vue limité, le risque est grand que des nœuds tombent dans une mauvaise position (pointant vers le sol, le ciel, d'un même côté qu'un autre, etc). Certains travaux [60] considèrent des caméras motorisées pour qu'elles puissent être orientées correctement après déploiement.

- *Vigilance environnemental* : Les réseaux de capteurs de vision sont aussi utiles pour la vigilance environnementale. Comme les nœuds consomment très peu d'énergie, ils peuvent être déployés dans des endroits stratégiques pour de longues périodes de temps (de l'ordre de plusieurs mois), afin d'obtenir des images d'intérêt scientifique sur de larges zones géographiques, par exemple près des nids, des abreuvoirs et réserves d'eau naturelles.

Grace à cela, les observateurs de la nature peuvent étudier le comportement et les habitudes des diverses espèces animales, en obtenant des scènes de lieux, qui peuvent être très éloignées, sans avoir besoin de se déplacer physiquement. En outre, des espèces naturellement timides qui rejettent la présence humaine pourraient être étudiées à travers

les caméras, permettant son étude et l'ouverture de nouvelles portes à la science. Des systèmes de support pour les tâches de gardes de parcs, comme la détection des incendies de forêt, pourraient être développés.

Quelques expériences peuvent être signalées pour ce type d'application. Par exemple, un réseau de capteurs infrarouges Cyclops a été déployé dans la réserve des Montagnes James San Jacinto (Californie), comme rapporté dans [61], dans les nids d'oiseaux pour les étudier au cours de la saison de nidification. Ils ont aussi utilisé ces capteurs pour des études herpétologiques. De même, dans [62], un système composé de caméras sans fil a été utilisé pour étudier le comportement des ours grizzly en Yukon, Canada, juste en-dessous du cercle arctique.

- *Sureté et protection des zones sensibles* : Evidemment, les réseaux de capteurs de vision peuvent être appliqués pour la sécurité des zones privées et publiques. Néanmoins, le véritable intérêt des réseaux de capteurs de vision sans fil n'est pas dans la surveillance des établissements fermés (par exemple, des industries, des bureaux, des magasins commerciaux, des résidences, etc.). Pour ce type de demande, une longue liste de produits est disponible sur le marché. Des webcams rotatives sans fil, micro-caméras et autres dispositifs existent déjà pour la vidéosurveillance. Ils s'appuient sur des méthodes de compression et des protocoles de communication normalisés, et des technologies de transmission à haut débit filaires ou sans fil.

Des systèmes de surveillance sans fil et limités en énergie pourraient être mis en place pour protéger des parcs, des zones sauvages, et d'autres zones liés à la protection des ressources naturelles, avec des caméras capables de dénoncer des chasseurs illégaux qui traversent les clôtures pour s'infiltrer dans les zones interdites, par exemple. D'autres applications pourraient être trouvées dans la surveillance des lieux privés ouverts, comme dans les industries forestières ou agricoles, par exemples.

- *Suivi du trafic routier* : Des réseaux de capteurs de caméras peuvent être déployés pour le suivi et le contrôle de la circulation routière. Des algorithmes d'analyse d'images peuvent être utilisés pour faire le dénombrement des véhicules ou des personnes pour estimer le niveau de trafic en fonction des heures de la journée.

1.6.3. Défis en matière de recherche

La vision est certainement le sens le plus puissant, mais aussi le plus complexe [63]. Les difficultés typiques sur les systèmes de vision, généralement associés à des problèmes de traitement à coût élevé, pourraient être multipliées lorsque nous devons faire face à d'énormes limitations en ressources, comme dans le domaine des réseaux de capteurs sans fil. Au-delà des défis traditionnels des réseaux de capteurs sans fil [64-65], les applications des réseaux de capteurs d'images posent des défis particuliers.

A part les défis spécifiques à la conception des matériels, nous identifions dans le champ des réseaux de capteurs d'images des défis analogues à ceux classés par [66] :

1. *Des protocoles de transmission et des algorithmes de compression d'images du monde réel* : La plupart des propositions actuelles sont évaluées soit par analyse mathématique, soit par simulation. Elles considèrent des hypothèses simplifiant à l'extrême le mode de communication des capteurs, la topologie du réseau, le positionnement des noeuds, les caractéristiques des noeuds, . . . , Ces hypothèses sont nécessaires pour simplifier les modèles mathématiques et les modèles de simulation utilisés pour évaluer les performances des propositions. Toutes les approches présentent d'excellents résultats par analyse mathématique et/ou simulation, mais qui y a-t-il de la réalité ? L'un des principaux défis pour les réseaux de capteurs de vision est de proposer des méthodes de compression et de communication réellement applicables, donc validées sur des plateformes réelles. Par exemple, la compression d'images selon la norme JPEG2000 a été largement discutée dans la bibliographie des réseaux de capteurs par ses indéniables qualités, en dépit de sa complexité. Au début, l'idée que le coût du traitement était négligeable prévalu, mais cette affirmation est certainement fautive [3][67]. Pour la plateforme mentionnée dans [3], JPEG2000 n'a pas donné de bons résultats. Quoi qu'il en soit, la faisabilité de JPEG2000 ou d'autres techniques de compression bien connues va dépendre des capacités des composants. La validation des modèles et des paramètres sur des plates-formes réelles doit donc être encouragée.
2. *Temps réel* : Le temps réel dans les réseaux de capteurs sans fil a été largement discuté dans la bibliographie [68-69]. Certaines applications des réseaux de capteurs de vision sont soumises à des contraintes de temps réel, et par conséquent, la mise en place de

mécanismes temps réel est nécessaire aussi bien en traitement d'images qu'en protocole de communication.

3. *Gestion de l'énergie* : Comme les nœuds ont par principe une source d'énergie très limitée, la gestion de l'énergie est le problème fondamental de la recherche dans les réseaux de capteurs. Ce problème est amplifié dans le cas des réseaux de capteurs d'image puisque les images forment des gros volumes de données.

Prenons l'exemple suivant : les auteurs de [70] ont évalué la puissance consommée et le temps d'exécution d'un mote Mica2 de Crossbow. Ils notent que le transcepteur radio d'un Mica2 consomme un courant de 3.72mA pour transmettre un octet à -20dBm (ce qui correspond à la puissance minimale de transmission), et cela prend environ 4.992E-004 secondes. Comme le Mica2 est alimenté avec une tension de 3V, nous avons une consommation d'énergie de 5,6µJ par octet transmis. Maintenant, pour transmettre une image de 129 × 129 pixels, la consommation d'énergie est d'environ 91mJ sans compter les en-têtes de paquets et les champs supplémentaires à insérer dans les paquets (numéro de l'image, offset des données, . . .). De plus, le coût de la capture d'image n'est pas négligeable, il coûte approximativement 90mJ pour une caméra Cyclops attachée à un mote Mica2. Cela est supérieur de plusieurs ordres de grandeurs au coût d'une mesure de lumière. Les capteurs d'images vont donc consommer beaucoup plus d'énergie que les capteurs traditionnels et donc, vont s'épuiser plus rapidement.

4. *Abstractions de la programmation* : Les défis sont les mêmes que pour les réseaux de capteurs sans fil traditionnels. Les développeurs ont besoin d'outils et de bibliothèques de programmation afin d'éviter l'exploration de la mise en œuvre de multiples niveaux.
5. *Sécurité et confidentialité* : La surveillance pour l'image est le mode de surveillance le plus intrusif qui soit. Les communications dans les réseaux de capteurs sans fil doivent faire face à des problèmes de sécurité parce que les nœuds sont généralement déployés dans des zones ouvertes faciles d'accès. Les attaques pourraient être réalisées aussi bien dans le matériel (par exemple en capturant et en prenant le contrôle d'un nœud) que dans les communications (attaques sur le protocole de communication). Pour cela, des

mécanismes de sécurité doivent être utilisés et adaptés aux contraintes des réseaux de capteurs sans fil.

Jusqu'à présent, la qualité de l'image n'a pas été considérée comme une exigence essentielle dans les applications de réseaux de capteurs de vision, mais seulement souhaitable. En fait, une grande quantité de travaux considèrent qu'il vaut mieux diminuer la qualité des images pour gagner sur la consommation d'énergie et sur le temps d'exécution qui sont des critères prioritaires. Nous pouvons citer par exemple les travaux de [71]. Les auteurs de [72] ont expliqué comment on pourrait obtenir des informations utiles pour certaines applications avec un capteur optique Agilent ADNS-3060 capable de prendre des images de 30×30 pixels seulement. Quoiqu'il en soit, le niveau minimal de la qualité d'image va dépendre des applications.

1.7. Conclusion

Dans ce chapitre nous avons procédé à l'étude des réseaux de capteurs sans fil. Nous avons exposé les fondements de base et fédéré quelques concepts nécessaires à la compréhension de nos problématiques dans la suite de ce manuscrit.

Cela fait des années que les réseaux de capteurs suscitent un engouement important dans la recherche. Nous avons remarqué à travers nos lectures que "minimiser la consommation d'énergie d'un nœud-capteur" est le problème fondamental de toutes les solutions et protocoles proposés. En effet, lorsque ce n'est pas l'objectif principal, alors c'est sûrement un critère de performance capital. Nous avons évoqué les formes de dissipation et les sources de surconsommation d'énergie par un nœud-capteur, les différentes techniques proposées pour réduire ces surconsommations.

Nous avons présenté un cas spécifique de réseau de capteurs sans fil : les réseaux de capteurs d'images. Comparés aux réseaux de capteurs classiques qui manipulent des données scalaires, les réseaux de capteurs d'images traitent un volume d'informations beaucoup plus grand. Cela nécessite des efforts supplémentaires pour adapter les différentes techniques de conservation d'énergie existantes.

Nous nous intéressent à la compression d'images pour diminuer le volume des données à transmettre afin de réduire la consommation de l'énergie.

Dans le prochain chapitre nous dresserons un panorama sur les différentes techniques de compression d'images dédiées aux réseaux de capteurs d'images proposées dans la littérature afin de prolonger la durée de vie des réseaux de capteurs.

Chapitre 2

La compression d'images dans les réseaux de capteurs sans fils

2.1. Introduction

Les progrès récents dans les systèmes micro-électromécaniques, les technologies de communication sans fils équipées de caméras à coûts abordables, ont donné un nouveau type de réseaux sans fils ad-hoc, appelés réseaux de capteurs d'images (WISN) [15]. Dans un WISN, chaque nœud dispose de la capacité d'acquérir, de compresser et de transmettre des trames vers la station de base, appelée également puits, à travers le trajet entre la source et le puits (voir figure 2.1). En règle générale, la station de base est un nœud puissant qui collecte les informations issues des autres nœuds (moins puissants). Ces réseaux jouissent d'une multitude d'applications potentielles, allant de la collecte de l'information visuelle des environnements difficiles d'accès, jusqu'au suivi et l'aide de populations âgées (voir chap 1)

Contrairement aux réseaux câblés classiques et les réseaux de capteurs sans fils (WSNs) qui traitent de données scalaires (humidité, température, etc...), les réseaux de capteurs d'images ont de nouveaux défis à relever : Par rapport aux réseaux câblés classiques, les réseaux de capteurs visuels rencontrent plus de problèmes en raison de leur nature sans fils intrinsèque et leurs ressources limitées (mémoire, traitement, batterie..etc.). Ils diffèrent de leurs prédécesseurs (réseaux de capteurs scalaires) essentiellement dans les points suivants :

1. La nature et le volume des flux visuels, qui sont des pixels, diffèrent des données scalaires simples manipulées par un réseau de capteurs classique
2. Un nœud, dans un réseau de capteur visuel, est équipé de caméra, d'où un champ de détection directionnelle donc une vue restreinte, contrairement au capteur de données scalaires.
3. Contrairement à un réseau scalaire, un nœud dans un réseau de capteur d'images exige des ressources importantes en termes de mémoire, de traitement et de puissance de communication pour pouvoir manipuler les flux visuels importants.
4. Des algorithmes de compression peu 'gourmands' en énergie sont obligatoires pour manipuler des images. Dans un capteur de données scalaires la compression n'est pas nécessaire.

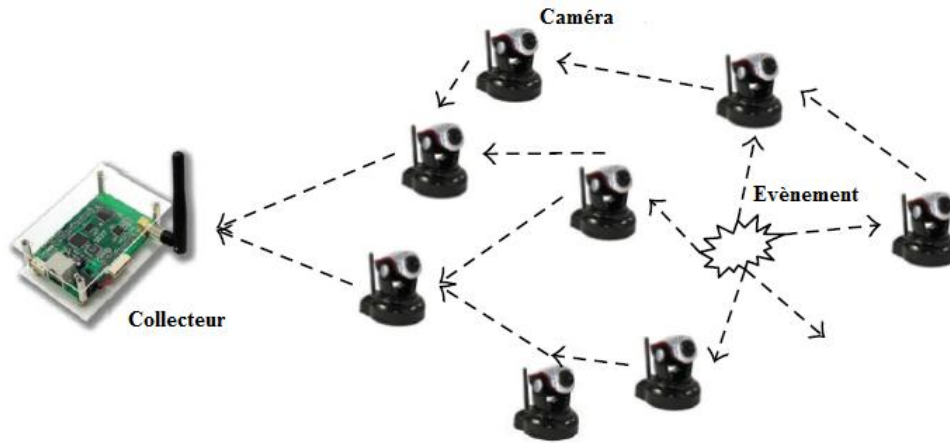


Figure 2.1 : Architecture typique d'un réseau de capteurs d'images

Comme le transmetteur radio est l'un des composants électroniques les plus 'gourmands' en énergie dans un nœud capteur [9], il est clair qu'un moyen d'économiser de l'énergie est de réduire la quantité de données à transmettre. la solution la plus évidente est la compression de données.

En règle générale, la compression est effectuée en exploitant la corrélation des données et la redondance. Dans un réseau de capteurs d'images, nous pouvons classer la redondance en trois catégories, comme suit :

- *La redondance inter-image* [73]. : c'est la redondance entre les trames successives capturées par le même capteur en un intervalle de temps donné. Cette redondance est connue comme redondance inter-image ou redondance temporelle. Elle est présente plus précisément dans les données de types vidéo.
- *la redondance entre les capteurs voisins qui suivent la même scène* : Cette redondance est aussi appelée redondance multi-vue. Dans le cas de la stéréovision, il s'agit de la corrélation qui existe entre deux images ou deux séquences vidéo de la même scène capturées simultanément par deux caméras à des angles de vues différents et se trouvant l'une de l'autre à une distance donnée. Dans les cas de la multi-vue il s'agit de la corrélation qui existe entre plusieurs scènes capturées par des caméras en même temps et avec des angles de vue différents. Plusieurs techniques ont été proposées pour réduire la redondance multi-vue. Un état de l'art sur ces différentes techniques est disponible sur la référence [74]

- *La redondance intra-image* [75]: c'est la redondance présente dans la même image. Il s'agit d'une corrélation qui existe entre un pixel donné dans une image par rapport à ses voisins dans la même image. C'est ce que nous appelons une redondance spatiale ou encore intra-image. Nous pouvons classer la redondance intra-image en trois types: la redondance spatiale, la redondance psycho-visuelle et la redondance statistique. Nous reviendrons, plus loin, sur ces différents types de redondance intra-image.

Le fait que 'moins il y aura de données à transmettre et moins le transmetteur radio consommera d'énergie' n'est pas toujours valable, car la compression a un coût d'énergie qui peut être très onéreux. Le problème est de disposer d'un algorithme de compression de données qui soit peu gourmand en énergie et qui ait un bon rapport débit-distorsion [3].

Une longue liste d'algorithmes de compression est aujourd'hui disponible [76], dont plusieurs, concernent la compression d'images. Il existe dans les littératures différentes méthodes de compression basées sur des approches vectorielles dans le domaine des ondelettes, [77-78] permettant d'atteindre à bas débit de meilleures performances en termes de qualité de reconstruction que les approches scalaires. Cependant, même lorsque leur complexité calculatoire est réduite comme dans [79], ces méthodes semblent encore être trop coûteuses pour une implantation dans un réseau de capteurs d'images.

La compression de données n'est pas un sujet nouveau, nous vivons aujourd'hui une grande évolution dans ce domaine. Cependant, la limitation de ressources des nœuds capteurs, comme la mémoire ou la vitesse des processeurs, rend inapplicables la plupart des algorithmes de compression existants utilisés dans l'informatique traditionnelle [3].

Dans la littérature des réseaux de capteurs sans fils, plusieurs algorithmes ont été proposés pour la compression de données. Citons par exemple quelques approches de compression distribuée [81-82], pipelined in-network compression [83] ou bien S-LZW [84]. Un état de l'art sur ces différentes techniques de compression est disponible sur [44].

Dans ce chapitre, nous concentrons notre étude sur les techniques de compression d'images spécifiques aux réseaux de capteurs et plus précisément la compression dite '**locale**'.

Pour que la compression de données soit rentable à la source, il faut que le traitement des données et la transmission des données compressées coutent moins d'énergie qu'un scénario en absence de compression. En effet, les auteurs de [3][67] ont démontré que la complexité de

certaines algorithmes de compression conduisent à des consommations d'énergie plus importantes que la transmission simple d'une image sans aucune compression. L'un des premiers travaux qui considèrent le compromis entre la consommation énergétique par traitement et par communication a été présenté par [85]. Les auteurs de ce travail, ont montré l'existence d'un nombre optimal de nœuds M impliqués dans une stratégie de compression qui minimise la consommation globale de puissance. Plus tard, les auteurs de [3] ont présenté un travail plus approfondi, dans lequel ils évaluent plusieurs algorithmes de compression traditionnels (tel que : JPEG, JPEG2000..etc), en analysant le compromis entre énergie consommée et traitement de données.

Dans ce qui suit, nous nous focaliserons sur l'état de l'art de la compression d'image allant des standards classiques (JPEG et JPEG2000 [86]), et leur applications dans les réseaux de capteurs d'images, jusqu'aux nouvelles méthodes de compression telles que l'échantillonnage compressé (Compressive Sampling : CS) [87]. Nous nous concentrerons sur la compression locale (à la source), les autres schémas de compression sortent du cadre de notre thèse. Nous donnerons un bref aperçu de chaque technique, ses avantages et ses inconvénients. Notre état de l'art respecte les points suivants:

- (1) nous étudierons et nous classerons les principales méthodes de compression locales proposées dans la littérature.
- (2) nous donnerons un bref aperçu de chaque méthode de compression.
- (3) nous conclurons par quelques directives qui concernent les paramètres pertinents qui doivent être pris en considération lors de la conception d'une méthode de compression pour un réseau de capteurs d'images.

2.2. La compression d'images dans les réseaux de capteurs d'images

La majorité des études qui ont été faites sur les réseaux de capteurs de données scalaires ont supposé que le coût de calcul, y compris l'acquisition et la compression de données, est insignifiant par rapport au coût de la communication des mêmes données [80]. Cette hypothèse est valide seulement quand il s'agit des capteurs de données scalaires, où le coût de la compression est négligeable devant le coût de la communication, nous dirons alors que la compression est rentable.

Dans le cas d'un réseau de capteurs sans fil traitant des données multimédia (les images dans notre cas), cette hypothèse n'est plus vérifiée. En effet, les flux visuels nécessitent toujours la compression. Dans cette section, nous montrons la relation entre le coût de compression et le coût de transmission dans le contexte d'un réseau de capteurs sans fils. Décider de faire précéder la transmission par une compression ou non, dépend essentiellement de l'algorithme de compression utilisé et éventuellement du processeur et de l'émetteur-récepteur radio.

Habituellement, compresser une image avant sa transmission est le choix idéal pour gagner en temps et en énergie. Il est clair que certains algorithmes de compression prennent plus de temps et consomment plus d'énergie que d'autres. Ces algorithmes sont, en général, utilisés pour le stockage, ou lorsqu'aucune exigence du temps réel n'est imposée. Par exemple, la plupart des algorithmes de compression reconnus tels que la compression fractales, JPEG, JPEG2000 sont très consommateurs de temps et d'énergie [56]. Par conséquent leurs applications aux réseaux de capteurs sans fils semblent être moins efficaces. Toutefois, lorsqu'ils sont appliqués aux réseaux filaires traditionnels, ces algorithmes donnent des taux de compression élevés indépendamment de l'énergie consommée.

Dans [3], les auteurs ont montré qu'une compression JPEG d'une image avant transmission consomme beaucoup plus d'énergie qu'une transmission directe de l'image sans compression !. Dans un tel cas, la compression n'est pas justifiée, nous dirons alors que la compression n'est pas rentable.

La figure 2.2 illustre une classification générale pour les algorithmes de traitement et compression d'images que nous avons trouvé dans la littérature. En principe, les méthodes de traitement traditionnelles ont été conçues pour travailler localement sur un seul et même processus. Dans les réseaux de capteurs, cela implique l'exécution de calculs au niveau du nœud 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, le deuxième sur la corrélation existante entre deux images ou plus, principalement quand elles contiennent des scènes voisines. Quelques approches hétérogènes pourraient combiner les deux stratégies.

Une nouvelle approche de compression, qui est l'échantillonnage compressé [87], a vu le jour ces dernières années grâce aux travaux d'Emmanuel Candès, Justin Romberg, Terence Tao [88]. Elle reste à ce jour théorique et de nouvelles approches 'hardwares' doivent être développées pour supporter ce nouveau paradigme de compression.

D'autres classifications de compression d'images se trouvent dans la littérature. Parmi tous ces algorithmes, nous essayerons d'examiner ceux dont l'application semble plus pratique tel que les algorithmes de codage de base.

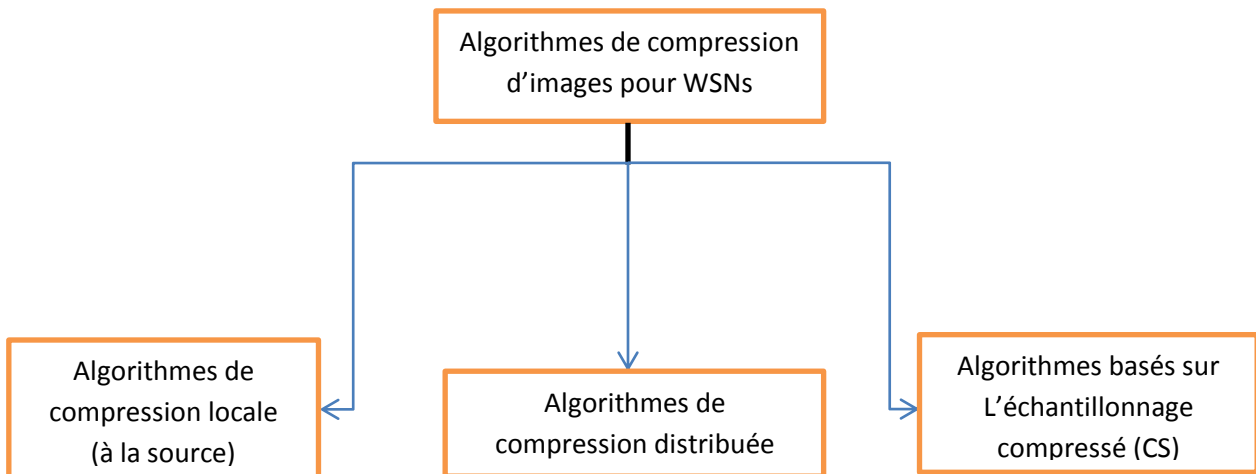


Figure 2.2. Classification générale des algorithmes de compression d'images pour les réseaux de capteurs d'images

Les techniques de compression d'images sont généralement classées en deux catégories : les compressions sans perte et les compressions avec perte. Les algorithmes de compression d'images sans perte permettent une reconstruction parfaite de l'image originale à partir de l'image compressée. On dit alors que ces méthodes de compression sont réversibles. Les algorithmes avec perte permettent seulement une approximation de l'image originale.

Les algorithmes de compression avec pertes sont fortement sollicités dans les réseaux de capteurs d'images comparés aux algorithmes de compression sans pertes. Toutefois, si les deux types de compression donnent les mêmes résultats en termes de consommation énergétique, les algorithmes de compression sans perte sont alors favorisés.

Nous pouvons regrouper les algorithmes de compression à la source en deux catégories (voir Figure 2.3): les algorithmes de compression à base de transformée, comme la transformée en

cosinus discrète (DCT) , la transformée en ondelettes discrète (DWT) et les algorithmes qui ne sont pas à la base de transformée tels que la quantification vectorielle [90-91] et la compression fractales [92]. Notons que la conception typique d'un algorithme basé transformé repose sur trois étapes: la dé-corrélation spatiale, suivie par la quantification, et un codeur entropique. Les autres schémas (algorithmes qui ne sont pas à la base de transformée) tels que la quantification vectorielle ou fractales ne suivent pas ce schéma.

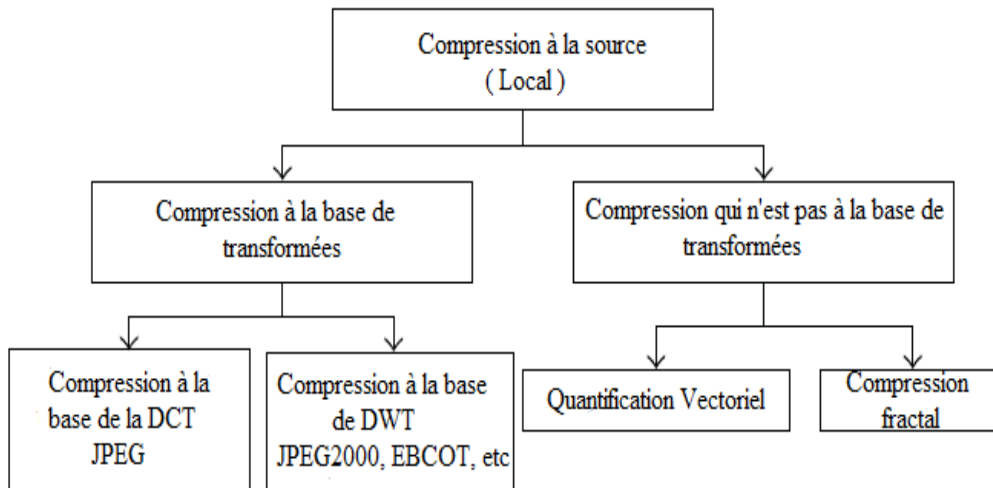


Figure 2.3. Classification des algorithmes de compression à la source

2.3. La compression à la base de transformées

2.3.1. Introduction

La compression d'images à base de transformées nécessite trois étapes de base. La première est la transformation des données de l'image pour obtenir des données moins corrélées. La seconde est la quantification ou le seuillage, c'est dans cette étape que la perte d'information se produit. Et enfin le codage où les données quantifiées sont réduites à des fins de transmission ou d'archivage (voir figure 2.4).



Figure 2.4 : Architecture typique d'une compression d'image à la base de transformée

Dans la phase de décompression ou de reconstruction de l'image, trois étapes sont nécessaires : le décodage, la déquantification et la transformation inverse (voir figure 2.5).

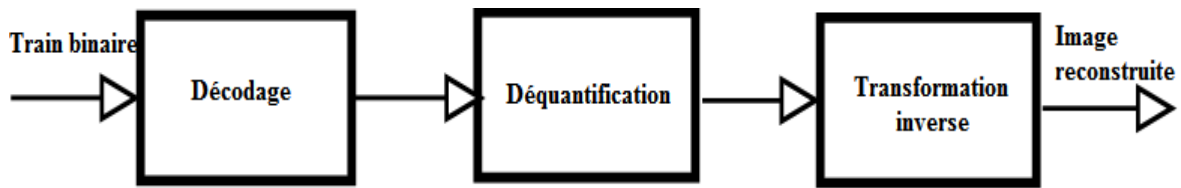


Figure 2.5 : Architecture typique d'une décompression d'image à la base de transformée

– Transformation

Le choix de la méthode de transformation doit permettre une réduction de la corrélation des données d'entrée. La sélection de la transformation est un facteur important dans la compression des données.

Les transformées les plus connues dans le domaine de la compression sont: La transformée de Fourier discrète (DFT), la transformée en cosinus discrète (DCT), la transformée de Hadamard discrète (DHT), la transformée de Karhunen Loève (KLT), la transformée de Walsh généralisée (WGT), la transformée en ondelettes discrète (DWT). DCT et DWT sont les plus utilisées dans la compression des données et en particulier dans la compression d'images [89].

– Quantification ou seuillage

La quantification permet de représenter numériquement chaque coefficient par un facteur de quantification (à partir d'une table) , le seuillage est l'élimination de coefficients quand leur valeur est inférieure à la valeur du seuil [93].

Par la quantification et le seuillage, on introduit une dégradation de la qualité et une réduction de la précision. Cette dégradation est basée sur le choix du facteur de quantification et la valeur du seuil. Si la valeur du seuil est élevée, la dégradation sera considérable mais si la valeur du seuil est faible, la valeur de la dégradation est négligeable ou faible avec un taux de compression aussi faible.

Le choix du facteur de quantification ou du seuil doit tenir compte du système visuel humain (HVS) pour atteindre une meilleure qualité de compression et un taux de compression élevé [83], [93]. Notons que le système visuel humain est davantage sensible aux fréquences basses qu'aux fréquences hautes.

La quantification peut être effectuée échantillon par échantillon, dans ce cas elle est dite 'scalaire'. Bloc par bloc, la quantification est 'vectorielle'. Cette dernière consiste à traiter conjointement n coefficients

En revanche, la plupart des approches vectorielles demeurent coûteuses en calcul même si la quantification vectorielle algébrique permet de réduire considérablement ce coût. Elles nécessitent, par ailleurs, beaucoup de ressources mémoire. Pour toutes ces raisons, elles ne sont pas actuellement éligibles dans le contexte des réseaux de capteurs [94].

– Codage entropique

Après quantification ou seuillage, la matrice obtenue est disponible pour le codage. Pour la transmission ou l'archivage, il est très avantageux de minimiser le nombre de bits du signal.

Les méthodes utilisées sont les méthodes de compressions réversibles. Nous pouvons citer les plus importants comme suit : le codage Huffman [95], le codage arithmétique [96], le codage de Golomb [97].

Selon le type de transformation, les deux normes de compression les plus connues sont la norme JPEG et la norme JPEG 2000.

Le système de compression le plus connu à la base de DCT est le standard JPEG [86]. Dans ce qui suit, nous détaillerons cette norme avec un exemple illustratif.

2.3.2. Norme JPEG

JPEG est l'acronyme de Joint Photographic Expert Group. Ce comité a donné son nom à la norme de compression d'images numériques. La norme JPEG fait référence à la norme ISO/CEI 10918-1 ou UIT-T recommandation T.81. Cette norme est basée sur la technique de transformation orthogonale DCT en découpant l'image en blocs de 16×16 , 8×8 , 4×4 , 4×8 ou 8×4 .

Il y a quatre modes de compression par JPEG [86]:

- Encodage séquentiel
- Encodage progressif
- Encodage sans perte
- Encodage hiérarchique

Nous présenterons dans ce qui suit le principe général de l'encodage séquentiel qui est la base des autres modes de compression. Comme le découpage de 8×8 est le plus adopté dans la communauté scientifique, nous nous contenterons dans le reste de ce chapitre de prendre cette taille.

– **Encodage séquentiel JPEG**

Les différentes étapes de l'encodage séquentiel JPEG se présentent comme suit :

- Un prétraitement est appliqué sur l'image à compresser
- L'image est coupée en bloc de taille 8×8 pixels
- La DCT est appliquée sur chaque bloc de gauche à droite et de haut en bas.
- Chaque bloc de coefficients obtenu après DCT est compressé par quantification
- Codage de la matrice quantifiée en zigzag
- Compression par l'algorithme de Huffman

Le schéma bloc de la chaîne de compression JPEG de base est représenté sur la figure suivante. Nous détaillons les différentes étapes dans ce qui suit.

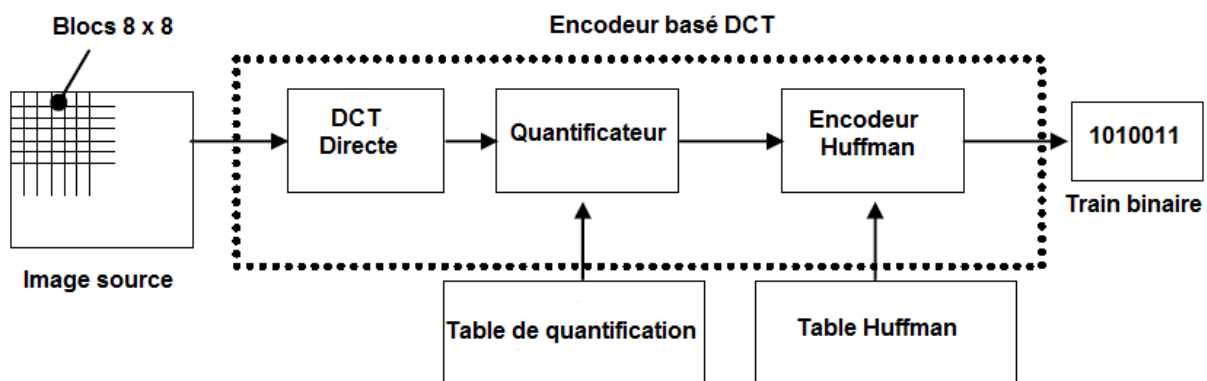


Figure 2.6 : Schéma bloc de la chaîne de compression JPEG séquentiel

– **DCT d'une image de taille 8×8**

Soit une image à deux dimensions $x(n_1, n_2)$ de taille 8×8 ($N_1 = N_2 = 8$). La transformation de cette image vers le domaine fréquentiel se fait en lui appliquant la transformation en cosinus discrète directe comme suit :

$$X[k_1, k_2] = \frac{1}{4} \alpha_{k_1} \alpha_{k_2} \sum_{n_1=0}^7 \sum_{n_2=0}^7 x[n_1, n_2] \cos \left[\frac{\pi(2n_1 + 1)k_1}{16} \right] \cos \left[\frac{\pi(2n_2 + 1)k_2}{16} \right] \quad (2.1)$$

La transformation inverse sera comme suit :

$$x[n_1, n_2] = \frac{1}{4} \sum_{k_1=0}^7 \sum_{k_2=0}^7 \alpha_{k_1} \alpha_{k_2} x[k_1, k_2] \cos \left[\frac{\pi(2n_1 + 1)k_1}{16} \right] \cos \left[\frac{\pi(2n_2 + 1)k_2}{16} \right] \quad (2.2)$$

- Matrice de forme de DCT

Pour le calcul rapide on utilise la matrice de forme dont les éléments sont donnés par l'expression suivante :

$$T(k_1, k_2) = \begin{cases} \frac{1}{\sqrt{8}} & \text{si } k_1 = 0 \\ \frac{1}{2} \cos \left[\frac{\pi(2k_2 + 1)k_1}{16} \right] & \text{si } k_1 > 0 \end{cases} \quad (2.3)$$

Le développement de l'équation précédente donne la matrice suivante :

$$\mathbf{T} = \begin{pmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.4904 & 0.4157 & 0.2778 & 0.0975 & -0.0975 & -0.2778 & -0.4157 & -0.4904 \\ 0.4619 & 0.1913 & -0.1913 & -0.4619 & -0.4619 & -0.1913 & 0.1913 & 0.4619 \\ 0.4157 & -0.0975 & -0.4904 & -0.2778 & 0.2778 & 0.4904 & 0.0975 & -0.4157 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 & -0.3536 & 0.3536 \\ 0.2778 & -0.4904 & 0.0975 & 0.4157 & -0.4157 & -0.0975 & 0.4904 & -0.2778 \\ 0.1913 & -0.4619 & 0.4619 & -0.1913 & -0.1913 & 0.4619 & -0.4619 & 0.1913 \\ 0.0975 & -0.2778 & 0.4157 & -0.4904 & 0.4904 & -0.4157 & 0.2778 & -0.0975 \end{pmatrix}$$

- Prétraitement

Pour une image en niveau de gris, les valeurs des pixels sont rangées de 0 à 255 avec un pas de 1. Un noir pur est représenté par zéro et un blanc pur est représenté par 255. Or, la DCT est applicable sur des pixels qui sont rangés de -128 à 127, alors le bloc original est nivelé par la soustraction de 128 pour chaque coefficient de la matrice d'entrée.

- Application de la DCT sur les blocs de taille 8×8

La DCT va être appliquée sur chaque bloc après prétraitement par la fonction :

$$D = T \times M \times T^T \quad (2.4)$$

M est la matrice nivelée,

T est la matrice de forme orthogonale et **T^T** sa transposée,

D est la forme matricielle équivalente de l'équation (3.14).

La matrice **M** est multipliée par **T** du côté gauche, (**TM**). Alors nous avons la transformation des lignes puis le résultat est multiplié par la **T^T** pour réaliser la transformation des colonnes.

– Quantification

Comme l'œil humain est moins sensible aux hautes fréquences, les éléments de la matrice de quantification sont représentés par un coefficient de poids correspondant à leur importance dans l'image. La matrice de quantification **Q** n'est pas unique; elle dépend du facteur de qualité choisi.

Ce choix nous informe sur la qualité et le niveau de compression désiré.

Si le facteur de qualité de la matrice de quantification est très petit alors la qualité de l'image reconstruite est médiocre et son taux de compression est élevé. Par contre, si ce facteur de qualité est très élevé alors la qualité de l'image reconstruite est très bonne mais avec un taux de compression faible.

Des expériences subjectives utilisant le système visuel humain ont fait jaillir la matrice de quantification standard de JPEG [86] [98]. La matrice ci-dessous donne un taux de compression très élevé et une qualité de décompression d'image excellente [86] [98].

$$Q = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

C'est dans l'étape de quantification que l'information se perd et la précision de l'image reconstruite est déterminée. Chaque coefficient de l'image va être divisé par le coefficient correspondant dans la matrice de quantification et le résultat sera arrondi à une valeur entière.

Le nombre de zéros contenus dans la matrice résultante après quantification varie suivant le choix du facteur de qualité.

– **Exemple**

Pour bien visualiser les différentes étapes de la compression par DCT, nous avons pris un bloc \mathbf{M} 8×8 et nous avons appliqué les étapes citées précédemment. La matrice \mathbf{M} à compresser est la suivante :

$$\mathbf{M} = \begin{pmatrix} 17 & 23 & 17 & 23 & 27 & 23 & 23 & 30 \\ 56 & 55 & 59 & 63 & 59 & 60 & 60 & 65 \\ 65 & 63 & 69 & 65 & 69 & 69 & 72 & 72 \\ 69 & 72 & 76 & 76 & 82 & 84 & 95 & 108 \\ 105 & 112 & 122 & 134 & 142 & 153 & 159 & 160 \\ 149 & 146 & 136 & 131 & 116 & 108 & 90 & 80 \\ 64 & 53 & 47 & 36 & 31 & 29 & 33 & 39 \\ 45 & 57 & 72 & 79 & 96 & 103 & 118 & 126 \end{pmatrix}$$

Nous appliquons un prétraitement on obtient :

$$M' = M - 128 \tag{2.5}$$

$$\mathbf{M}' = \begin{pmatrix} -111 & -105 & -111 & -105 & -101 & -105 & -105 & -98 \\ -72 & -73 & -69 & -65 & -69 & -68 & -68 & -63 \\ -63 & -65 & -59 & -63 & -59 & -59 & -56 & -56 \\ -59 & -56 & -52 & -52 & -46 & -44 & -33 & -20 \\ -23 & -16 & -6 & 6 & 14 & 25 & 31 & 32 \\ 21 & 18 & 8 & 3 & -12 & -20 & -23 & -48 \\ -64 & -75 & -81 & -92 & -97 & -99 & -95 & -89 \\ -83 & -71 & -56 & -49 & -32 & -25 & -10 & -2 \end{pmatrix}$$

Nous appliquons la DCT 8×8 (voir formule (2.3), nous obtenons la matrice suivante :

$$\mathbf{D} = \begin{pmatrix} -406.875 & -32.7756 & 3.7003 & -6.9407 & 2.625 & -2.114 & 0.7674 & 1.0272 \\ -122.7516 & 1.1753 & -2.9122 & -0.1802 & 3.3924 & -1.7106 & -0.9811 & -2.008 \\ -188.8345 & -5.8142 & 2.1062 & -1.8838 & 2.2092 & -3.0997 & -1.638 & 0.9699 \\ 32.7689 & 60.4568 & -7.0628 & 8.0392 & 0.5468 & -0.2609 & -2.0338 & -7.2928 \\ 39.875 & -88.2675 & -2.156 & -6.0612 & -0.125 & -4.9191 & -3.5718 & 1.532 \\ -152.2 & 37.4349 & 17.8185 & -0.9156 & 3.2796 & -2.9053 & 0.0242 & 2.3619 \\ 53.9993 & 18.7591 & -13.388 & 4.6601 & -3.4858 & -1.8305 & -1.6062 & -0.0365 \\ -18.8039 & -27.1627 & 2.02516 & 2.3055 & -1.5409 & 1.6776 & 0.3056 & 2.1908 \end{pmatrix}$$

En observant cette matrice, nous soulignons les points suivants:

- Les coefficients dont l'amplitude est la plus élevée en valeur absolue sont situés en haut de la matrice et du côté gauche
- L'amplitude la plus élevée de la matrice est représentée par le premier coefficient de la matrice
- L'amplitude du premier coefficient de la matrice est presque la somme de tous les autres coefficients de la matrice. Il s'agit du coefficient représentant la composante continue du bloc ou DC. Les 63 autres coefficients sont les composantes alternatives ou AC.

Nous appliquons la quantification à la matrice D, Chaque coefficient DCT de la matrice D est divisé par le coefficient correspondant dans la table de quantification Q et le résultat sera arrondi à la valeur entière selon la formule suivante :

$$A = \text{Arrondi} \left[\frac{D_{ij}}{Q_{ij}} \right] \quad (2.6)$$

$$A = \begin{pmatrix} -25 & -3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -13 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & -4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Nous avons beaucoup de zéros comme prévu et la valeur la plus élevée en valeur absolue est localisée dans le premier coefficient de la matrice.

Le résultat de la matrice A est ensuite codé en zigzag puis encodé par le codeur Huffman pour réaliser le codage JPEG.

– Codage

Après quantification, les 64 coefficients de chaque bloc sont traités pour l'encodage entropique. L'encodage entropique utilise des statistiques sur la source d'entrée pour former un code. Les propriétés de ce code dépendent de la source. L'encodage Huffman et l'encodage arithmétique sont deux principaux types de l'encodage entropique. Le traitement avant encodage est présenté dans la figure 2.7 et les différentes étapes de ce traitement sont données comme suit:

Dans chaque bloc nous avons 64 coefficients; le premier coefficient C_{00} est la composante DC du bloc et les 63 autres coefficients sont les composantes AC ($AC_{ij} = C_{ij}, i, j = 0,1 \dots 7$) du bloc. Les coefficients DC et les coefficients AC sont codés différemment. Si on prend une

image $N \times N$ avec $N=256$ on aura 32 blocs de 8×8 donc 32 coefficients DC. Pour l'encodage des coefficients DC, on codifie la différence des coefficients DC des différents blocs successivement suivant la formule suivante $Diff_b = DC_b - DC_{b-1}$ avec $b = 1, \dots, N/8$. Ainsi, le coefficient DC précédent prédit le coefficient suivant. Pour l'encodage des coefficients AC on utilise pour chaque bloc le codage en zigzag. Cette méthode permet d'augmenter la taille des suites de zéro.

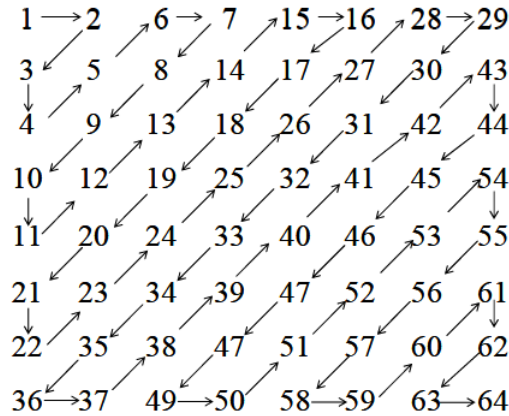


Figure 2.7 : Encodage en Zig zag

Reconstruction de l'image

La phase de reconstruction de l'image suit le chemin inverse de la phase de compression. Après le décodage des données de l'image comprimée, chaque élément de la matrice obtenue est multiplié par le coefficient qui lui correspond dans la matrice de quantification, précédemment utilisée. C'est uniquement la partie entière de chaque coefficient qui est considérée en sortie. La transformée en cosinus discrète inverse est appliquée sur la sortie du quantificateur inverse. Du moment que nous avons appliqué un prétraitement au début de la compression en appliquant la soustraction du nombre 128 pour chaque coefficient, alors un ajout du nombre 128 est apposé pour chaque coefficient de la matrice de transformation inverse. Les différentes étapes de la reconstruction sont représentées dans la figure suivante :

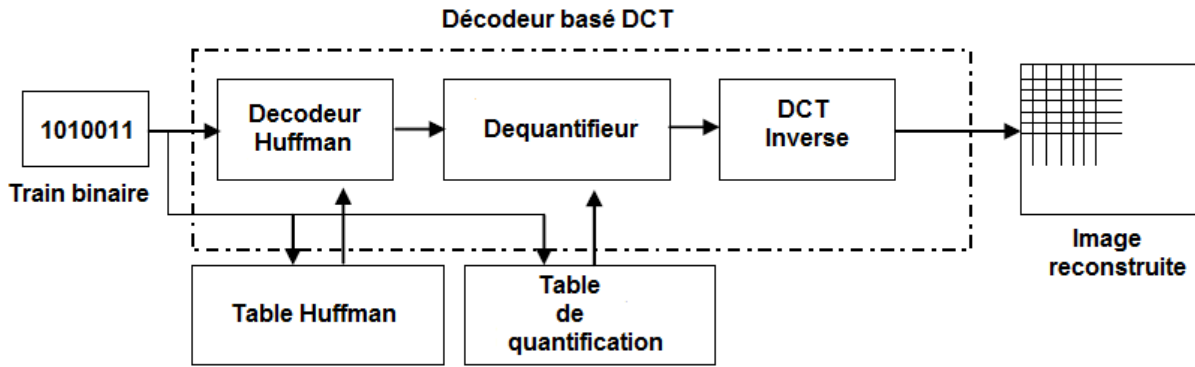


Figure 2.8: Schéma bloc de la chaîne de Décompression JPEG séquentielle

En appliquant les étapes de la décompression sur la matrice A (C'est la matrice obtenu après décodage de Huffman) selon les expressions suivantes :

$$A^{\wedge} = \text{Arrondi} [A_{ij} \times Q_{ij}] \quad (2.7)$$

$$B = IDCT(A^{\wedge}) - 128 \quad (2.8)$$

Nous obtenons :

$$B = \begin{pmatrix} 23 & 23 & 24 & 25 & 26 & 27 & 27 & 28 \\ 57 & 58 & 58 & 59 & 60 & 61 & 61 & 62 \\ 67 & 68 & 69 & 71 & 73 & 75 & 77 & 77 \\ 58 & 61 & 68 & 76 & 85 & 94 & 100 & 104 \\ 118 & 121 & 126 & 133 & 141 & 148 & 153 & 156 \\ 140 & 136 & 129 & 120 & 111 & 102 & 95 & 91 \\ 68 & 65 & 59 & 51 & 43 & 35 & 29 & 26 \\ 47 & 53 & 64 & 79 & 95 & 109 & 121 & 127 \end{pmatrix}$$

Si nous comparons la matrice B (matrice décompressée) avec la matrice d'origine M, nous pouvons constater que les valeurs des matrices sont très proches les unes des autres.

2.3.3. Adaptation de JPEG aux contraintes des réseaux de capteurs d'images

Plusieurs variantes des méthodes de compression à la base de DCT sont proposées dans la littérature pour améliorer les caractéristiques de JPEG, pour minimiser les effets de blocs, pour améliorer la qualité et pour augmenter le taux de compression.

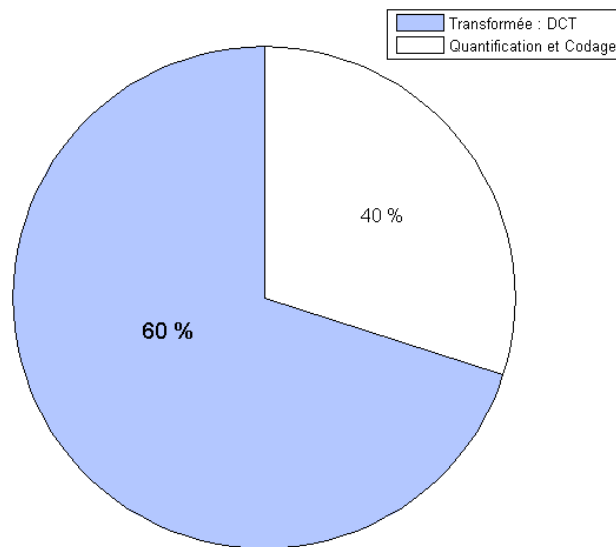


Figure 2.9. *Coût énergétique de la DCT dans une chaîne JPEG*

L'adoption de JPEG comme outil de compression dans les réseaux de capteurs d'images n'est pas très bénéfique en termes de consommation d'énergie [3][57]. Cela est dû essentiellement à la complexité du codeur, et précisément à la DCT qui représente l'étape la plus gourmande en énergie.

Au sein d'une chaîne de compression JPEG, la DCT est la plus coûteuse en matière d'énergie. Elle représente plus de 60% du coût de calcul de l'algorithme [5] (voir Figure 2.9). De nombreuses tentatives pour réduire sa consommation énergétique ont été proposées dans la littérature. Certains d'entre elles sont utiles pour un réseau de capteurs d'images. Nous pouvons classer les travaux les plus importants comme suit :

- 1) *Niveau algorithmique*: La réduction du nombre des opérations gourmandes en énergie, telles que les multiplications. Il a été démontré théoriquement que le nombre minimum pour calculer la DCT 8 points est 11 multiplications [99]. L'un des algorithmes les plus célèbres dans la littérature, qui atteint cette limite est l'algorithme de Loeffler [100]. D'autres algorithmes existent dans la littérature pour réduire la complexité de calcul de la DCT. Les auteurs de [101-102] proposent de convertir les multiplications en des opérations simples, telles que les additions et les décalages. En effet, la DCT peut être implémentée en utilisant seulement des opérations d'additions et de décalages. Cependant, ces dernières sont des approximations et par conséquent elles dégradent la qualité des images finales. En outre, ces systèmes permettent un faible

coût et une implémentation rapide par rapport à la DCT exacte. Une liste détaillée de ces différents algorithmes est disponible dans [103].

2) *Niveau architectural* : Plusieurs travaux ont été proposés pour implémenter la DCT.

Nous citons par exemple :

- les architectures qui utilisent le parallélisme et le pipeline de la DCT 2D: les auteurs dans [104] utilisent la méthode de décomposition ligne\colonne basée sur deux processeurs DCT 1D et une mémoire tampon intermédiaire en utilisant le parallélisme et le pipeline. L'architecture proposée permet aux éléments de traitement et aux unités arithmétiques de fonctionner en parallèle. Ceci réduit à la fois la complexité de calcul et la mémoire interne, et permet un débit élevé [105].
- L'utilisation d'une DCT à point fixe à la place de la DCT à virgule flottante: la DCT à virgule flottante présente une forte consommation d'énergie. À titre d'illustration, prenons l'exemple de [105]. Le Codage JPEG d'une image en niveaux de gris, à 1 bit par pixel (bpp) au format QCIF, en utilisant le processeur StrongARM SA1110 nécessite 2.87 mJ avec la DCT entière. La même opération en utilisant la DCT à virgule flottante nécessite plus de 22 mJ. Ceci justifie le choix d'une DCT à point fixe dans les réseaux de capteurs d'images.
- Une autre technique utilisée pour réduire les éléments de traitement est l'arithmétique distribuée (DA) [106]. Elle consiste à remplacer les multiplieurs constants par des tables "lookup" et des accumulateurs pour réduire la complexité de l'implémentation.

D'autres travaux du niveau architectural ont été proposés pour réduire davantage la consommation énergétique de la DCT. Nous pouvons citer : [107-108]. Une liste plus détaillée est disponible sur [109].

En résumé, plusieurs adaptations, qui ont été faites sur JPEG, montrent la possibilité de son application comme un outil de compression d'images dans les réseaux de capteurs d'images [110].

2.3.4. Compression locale par JPEG2000

JPEG2000 est un algorithme de compression d'images basées sur un processus très complexe qui inclut une transformée en ondelettes dyadique (voir figure 2.10), une allocation de bits, une quantification et un codage entropique. L'un des premiers travaux qui ont appliqué les

techniques de compression basé sur la DWT sur les réseaux de capteurs d'images est [67]. Les auteurs de [67] ont introduit une technique de faible consommation énergétique qui incorpore le standard JPEG2000 pour la compression d'images depuis un nœud caméra sans fil. C'est l'un des premiers travaux qui a considéré un scénario multi-saut pour réseaux de capteurs d'images. Ils ont formulé la transmission d'images comme un problème d'optimisation et ils ont proposé une heuristique appelé MTE (Minimize Total Energy). Ici, il est supposé qu'un nœud source a une connaissance du nombre (estimé) de nœuds intermédiaires entre lui et le puits, et chaque nœud est capable de connaître son état de batterie.

L'algorithme travaille comme suit : premièrement, le nœud source vérifie une table avec des paramètres de quantification et niveaux d'ondelette pré-calculés pour un réseau déterminé et une qualité d'image requise. Le nœud de capteur compresse l'image capturée en appliquant la compression par ondelettes, puis, il calcule un taux de compression et une dissipation d'énergie. Une nouvelle compression est donc réalisée pour calculer des nouveaux taux de compression et dissipation d'énergie. Ces valeurs sont comparées avec les anciennes. Ces étapes sont répétées tant que les nouveaux taux de compression et de dissipation d'énergie sont meilleures ou jusqu'à un certain nombre d'itérations.

Quelques alternatives sont discutées pour obtenir encore plus d'économies d'énergie, comme l'application de tables de recherche recalculées ou le calcul des paramètres optimaux dans un centre de traitement externe. On peut quand même voir que cette méthode pousse à compresser l'image plusieurs fois et on a du mal à voir comment la source va économiser l'énergie. En fait, cette méthode vise à optimiser l'énergie dans les nœuds de transit mais elle n'est pas bonne pour la source.

Les auteurs de [111] ont proposé une autre stratégie de transmission d'images basées sur JPEG2000 pour des réseaux de capteurs point-a-point. En partant d'une distorsion d'image attendue, une unité de contrôle est capable de déterminer le nombre de niveaux de qualité (couches) à être transmises et de trouver le schéma le plus efficace en énergie.

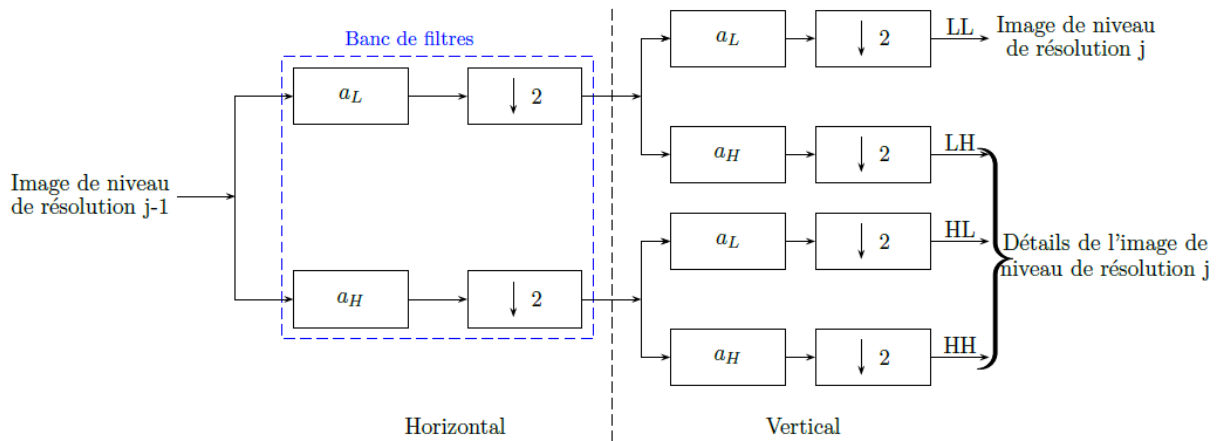


Figure 2.10. Transformée en ondelettes 2D. Schéma de décomposition classique en sous bandes

2.3.5. Schéma basé sur le codage SPIHT

L'une des premières propositions pour la compression d'images dans les réseaux de capteurs a été introduite dans [112]. Les auteurs de [112] ont proposé un schéma basé sur le codage SPIHT, des blocs de données sont générés par relation parent-enfant de coefficients d'ondelettes. Cette relation parent-enfant est effectuée afin de renforcer la robustesse de SPIHT aux erreurs de transmission. L'algorithme fonctionne comme suit : premièrement, l'image capturée est décomposée en multiple résolutions en appliquant une transformée en ondelettes discrètes. Puis les coefficients d'ondelettes sont regroupés en fonction de leur relation parent-enfant. Chacun de ces groupes est codé indépendamment par l'algorithme SPIHT. De cette manière, les erreurs possibles lors de la transmission affecteront seulement le bloc erroné, en permettant alors la reconstruction de l'image avec des pertes possibles d'information.

Les expériences ont été réalisées avec un processeur Intel StrongARM SA 1110 et un émetteur radio LMX3162. Les auteurs de [112] proposent un schéma de transmission basé sur RCPC/CRC pour permettre la protection aux erreurs. Les résultats montrent des économies d'énergie en termes de traitement de données et une réduction effective de la propagation des erreurs.

2.4. Compression distribuée

2.4.1. Introduction

Un bon nombre de travaux a considéré la compression distribuée des images, en raison de la nature distribuée des réseaux de capteurs. Notons que plusieurs propositions ont été publiées sous le qualificatif de compression distribuée, même si elles ne réalisent pas vraiment du traitement distribué, mais plutôt du traitement collaboratif.

Dans la bibliographie des réseaux de capteurs d'images, deux approches différentes ont été cataloguées comme méthodes « distribuées ». Le premier groupe (le plus développé) profite de la corrélation existante entre deux ou plusieurs images capturées par deux caméras différentes ou plusieurs (principalement voisines), en cherchant à diminuer la quantité de données à transmettre depuis chacune.

Une deuxième catégorie considère une approche plus exacte à la définition formelle du traitement distribué, en envoyant quelques zones d'une image à des nœuds différents pour qu'ils fassent, chacun, une partie du traitement.

Dans le reste de cette section nous allons décrire et discuter quelques-uns des travaux trouvés dans la littérature sur le traitement distribué des images appliqué spécifiquement aux réseaux de capteurs sans fil.

2.4.2. Compression distribuée d'images corrélées

En 1973, Slepian et Wolf [113] ont présenté leur théorème sur le codage de sources corrélées codées indépendamment, mais décodées toutes ensemble. Ces propositions de la théorie de l'information classique ont inspiré plusieurs des approches proposées pour le traitement collaboratif- distribué d'images sur réseaux de capteurs sans fil. La principale idée, ici, est de tirer avantage de la corrélation entre deux ou plusieurs images d'une même scène mais originaires de plusieurs caméras. Cette corrélation est plus prononcée quand les caméras capturent des scènes similaires, et plus encore quand elles proviennent de sources voisines.

Même si le théorème de Slepian-Wolf [113] a inspiré la plupart des travaux publiés sur la compression distribuée dans les réseaux de capteurs d'images, son application n'est pas exempte de difficultés. En effet, les auteurs de [114] ont proposé un schéma pour des caméras multiples qui capturent différents points de vue de la même scène. Ici, un premier

nœud de caméra prend une image d'une scène générale et un ensemble de caméras secondaires ont chacune une vision chevauchée de la région de la première scène. La méthode est basée sur l'approche de contexte de forme présenté par [115]. Elle permet la détermination des similarités entre deux images en représentant les formes des objets capturés comme un ensemble de points échantillonnés depuis ces contours, et par le calcul de la correspondance des points de fonction, en envoyant alors des versions de faible résolution des zones des images chevauchées déterminées. Du côté du décodeur, un algorithme de super-résolution est utilisé afin de reconstruire une image de plus haute résolution de la région partagée.

Le cas des images corrélées a été largement étudié sur [116]. Ils proposent une méthode distribuée pour profiter de la corrélation entre plusieurs vues capturées par plusieurs caméras adjacentes en utilisant une certaine information géométrique afin de réduire le débit de transmission

2.4.3. Compression distribuée sur une architecture clustérisée

Dans les travaux présentés sur [117], la compression distribuée d'images en utilisant le standard JPEG2000 est proposée. L'idée de base est de répartir la charge de travail du calcul de la transformée en ondelettes entre les différents nœuds. Deux méthodes sont proposées :

- 1) *Méthode de transformée d'ondelettes parallèles* : Dans la première méthode, l'application d'une transformée d'ondelettes parallèles est proposée. L'image capturée est divisée en n blocs de données R_1, \dots, R_n , consistant en une ou plusieurs lignes. Ces blocs sont transmis à certains nœuds voisins de la source. Ces nœuds effectuent une transformée en ondelettes unidimensionnelles (1-D) pour chaque bloc de données qui lui a été envoyé, puis transmet le résultat à un nœud agrégat. Celui-ci divise les données obtenues, cette fois en m blocs I_1, \dots, I_m composé de colonnes. Puis il distribue ces blocs en les transmettant à des nœuds voisins. Ceux-ci effectuent alors la transformée en ondelettes 1-D et les renvoient au nœud agrégat. Celui-ci récupère tous les blocs et obtient aussi le résultat de transformation bidimensionnelle (2-D) en ondelettes de l'image.
- 2) *Méthode de carrelage* : Dans cette deuxième méthode, l'image capturée est divisée en tuiles (de blocs de données), puis ces tuiles sont distribuées à un certain nombre de

nœuds voisins. Ces nœuds effectuent la transformation en ondelettes 2-D de façon indépendante sur ces tuiles. Enfin, les résultats sont transmis à un nœud agrégat.

Un autre schéma de compression d'images pour les réseaux de capteurs est présenté dans [118]. Ce schéma est basé sur la LBT (Lapped biorthogonal transform), l'algorithme zero-tree, la quantification multiple et le codage de Golomb. La LBT était préférée à la DWT car elle nécessitait beaucoup moins de calculs et d'espace mémoire. De même, le codage de Golomb était préféré au codage de Huffman ou au codage arithmétique pour les mêmes raisons. Les auteurs se sont inspirés de la méthode clustérisée décrite précédemment pour distribuer la charge de calcul et la consommation d'énergie d'un nœud source individuel.

Grosso modo, la distribution de l'algorithme de compression marche comme suit : premièrement, le nœud source envoie un message de requête à l'un de ses voisins (S), qui la renvoie à la tête de cluster. Le nœud tête de cluster choisit quelques nœuds dans son cluster qui ont une quantité d'énergie minimale et envoie un message d'acceptation vers S, qui le renvoie vers le nœud caméra. Le nœud caméra envoie 8 lignes de données vers S qui les distribue aux nœuds choisis. Ces nœuds réalisent la compression et envoient les données compressées vers la tête du cluster.

2.5. Échantillonnage compressé (Compressive Sampling : CS)

L'échantillonnage compressé (CS) est un nouveau paradigme qui échantillonne un signal en un nombre d'échantillons bien inférieurs au nombre qui le stipule le théorème de Nyquist. Ce nouveau paradigme a attiré l'attention de beaucoup de chercheurs parce qu'il utilise la propriété de parcimonie (sparsity) des signaux. La parcimonie est une caractéristique inévitable de plusieurs signaux naturels. Elle permet aux signaux d'être stockés dans quelques échantillons et ensuite d'être récupérés avec précision. Nous pouvons citer par exemple, une image dans le domaine transformé. La quasi-totalité de l'énergie de l'image est compactée dans les 10 % premiers coefficients [89].

Les origines de l'échantillonnage compressé (CS) datent de quatre décennies. Il a été utilisé pour la première fois en sismologie, en 1970 [119]. L'idée de l'échantillonnage compressé vient des travaux de David Donoho, Emmanuel Candes, Justin Romberg et Terence Tao [88][120] qui ont donné les fondements mathématiques de cette théorie.

2.5.1 Théorème de Nyquist

En 1949, Shannon énonce que l'échantillonnage d'un signal en bande limitée, c'est-à-dire sa représentation sous une forme discrète, par une liste de valeurs prélevées régulièrement dans ce signal, exige une fréquence d'échantillonnage supérieure au double de l'écart entre les fréquences minimale et maximale qu'il contient. Par conséquent un nombre N d'échantillon minimal doit être respecté. Dans les techniques de traitement du signal traditionnel, nous prenons ' N ' échantillons qui respecte le taux de Nyquist, avant la transmission. Ces échantillons sont ensuite compressés vers un nombre ' M ' d'échantillons, par conséquent ' $N-M$ ' échantillons seront jetés. Au niveau du récepteur, la décompression de données aura lieu pour récupérer ' N ' échantillons à partir de ' M ' échantillons. Le paradigme de la théorie de l'échantillonnage de Shannon possède beaucoup d'inconvénients [121] surtout quand il s'agit des signaux large bandes.

Après avoir examiné le théorème d'échantillonnage classique on peut se demander: pourquoi devrions-nous passer par N échantillons alors que nous avons seulement besoin de ' M ' échantillons à la fin de la transmission? , Comment pouvons-nous acquérir un nombre d'échantillons bien inférieur à ' N ' ?. La théorie alternative de L'échantillonnage compressé répond à ces questions. La CS a apporté une contribution importante à la littérature de traitement de signal, en donnant à la théorie de l'échantillonnage classique une autre dimension. Les différents blocs constructifs de la CS sont représentés dans la figure 2.11

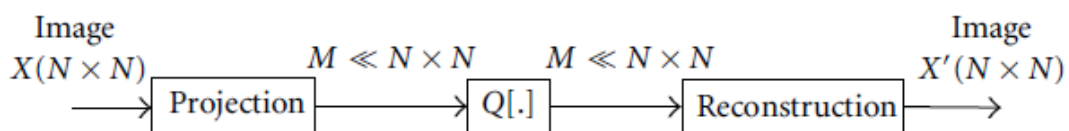


Figure 2.11 : Compression par échantillonnage compressé

2.5.2 Application de l'échantillonnage compressé aux réseaux de capteurs d'images

Au cours des dernières années, des progrès considérables ont été faits dans la résolution de nombreux problèmes dans les réseaux de capteurs sans fil. En ce qui concerne les techniques de compression d'images, deux problèmes majeurs demeurent : la complexité algorithmique du codeur et la faible résistance aux erreurs du canal de transmission.

- *la complexité algorithmique du codeur* : le codage classique nécessite des algorithmes de traitement complexes, qui conduisent à une forte consommation d'énergie. De nouveaux paradigmes de codage sont donc nécessaires pour réduire la complexité des codeurs traditionnels, qui ne sont pas adaptés à des capteurs d'images embarqués. Récemment, les algorithmes de codage distribué [122-123] (aka Wyner - Ziv) exploitent les statistiques de la source au décodeur, déplaçant ainsi la complexité au décodeur. Les codeurs Wyner - Ziv ont besoin de chaîne de retour de bout en bout [124], ce qui introduit un retard dans la communication. En outre, les gains des codecs en termes de PSNR sont limités (2 - 5 dB) [124-125]. Des encodeurs distribués qui ne nécessitent pas de retransmission de bout en bout ont été récemment proposés [126], mais au prix d'une réduction supplémentaire de la performance.
- *La faible résistance aux erreurs du canal de transmission* : les techniques de compressions classiques telles que JPEG résistent peu aux erreurs de canaux de transmission [4]

L'échantillonnage compressé peut offrir une solution pour les problèmes mentionnés ci-dessus [127]. La CS détecte et compresse les données simultanément à très faible complexité. Elle permet aussi une résistance sans précédent aux erreurs de canaux [121]. La CS a touché beaucoup de domaine d'applications, cependant afin de supporter ce nouvel paradigme, des équipements dédiés doivent être conçus.

2.6. Conception d'une méthode de compression pour un réseau de capteur visuel

En général, la conception d'un algorithme de compression efficace au sens de l'énergie dépend de l'ensemble des étapes de la chaîne de compression. Rappelons que les algorithmes à base de transformation sont préférés aux algorithmes qui ne sont pas à base de transformation. En d'autres termes, cela dépend de la transformation sélectionnée, telles que DCT ou DWT ; la sélection d'une matrice de quantification appropriée, le codeur entropique, tel que Huffman ou Golomb-Rice, et l'interconnexion entre ces étapes. En outre, selon le domaine d'application, une compression avec ou sans pertes doit être sélectionnée, sachant que la compression avec perte est généralement préférable par rapport à une compression sans perte, en termes d'efficacité énergétique. Nous devons également mentionner qu'il est obligatoire de prendre en considération la phase d'acquisition avant la compression. En fait, à l'exception de CS, toutes les méthodes de compression ne prennent pas en considération l'acquisition d'image lors de l'encodage. Joignant ensemble la phase d'acquisition et de

compression permet de réduire considérablement l'énergie d'un capteur visuel. Un autre point connexe est de savoir si les nœuds intermédiaires, dans le chemin établi entre la source et la destination, peuvent coder et décoder des images. Le décodage et l'encodage des images par des nœuds intermédiaires nécessitent de l'énergie supplémentaire. Dans un tel cas (encodage et décodage), le processus de décodage doit être de faible complexité afin d'être efficace en termes d'énergie.

En général, un algorithme de compression dédié pour des réseaux de capteurs d'images doit présenter les propriétés suivantes :

1. Un taux de compression acceptable
2. Une faible consommation énergétique
3. Une faible complexité calculatoire
4. Faible utilisation de la mémoire dynamique
5. Un encodage embarqué

2.7. Conclusion

Dans ce chapitre nous avons abordé en détails l'état de l'art sur les techniques de compression d'une manière générale. Cependant, nous avons beaucoup plus insisté sur les normes de compression connues et aussi les techniques de compression adaptées pour le cas des réseaux de capteurs sans fil. Ces derniers exigent que les techniques de compression adoptées doivent, bien sûr, répondre aux critères classiques, à savoir un compromis débit-distorsions, mais aussi et surtout garantir une réduction de la complexité algorithmique. Ceci, dans le but d'alléger le fonctionnement du processeur du nœud de capteur et par conséquent augmenter l'autonomie énergétique.

Chapitre 3

Transformations de faible complexité

3.1 Introduction

La DCT « Discrete Cosine Transform » (TCD « Transformée en Cosinus Discrète ») est une transformation orthogonale discrète. Elle est utilisée dans sa forme unidimensionnelle (1D) dans de nombreuses applications (voir [128] par exemple), etc. Mais le plus souvent, la DCT est utilisée sous sa forme bidimensionnelle (2D) en traitement d'images pour des applications diverses telles que la détection de formes [129], le tatouage numérique « watermarking » [130], l'estimation de mouvement [131] et plus particulièrement la compression d'images [132].

En outre, elle est utilisée dans la majorité des standards de compressions d'images telles que le JPEG et le JPEG XR et de vidéos telles que MPEG, H264 et HEVC. En effet, la DCT contribue à la réduction de la redondance spatiale en transformant le domaine spatial en domaine spectral. Cependant, la forme et la taille des blocs de la DCT 2D changent en fonction de la norme à implanter et aussi en fonction de l'application spécifique.

Dans le chapitre précédent, nous avons présenté les trois étapes de la chaîne de compression. Notre choix s'oriente vers la compression JPEG qui est la méthode la plus adaptée. Même si la complexité de la DCT est inférieure à celle de la transformées en ondelettes [132], la DCT demeure coûteuse pour une implantation dans le contexte des réseaux de capteurs.

Dans le schéma de compression JPEG, l'étape de la DCT représente à elle seule 60% du coût total en énergie nécessaire pour l'exécution de la technique de compression [133]. Cependant, afin de rendre l'utilisation de la DCT compatible avec les contraintes des réseaux de capteurs, nous avons étudié les différents algorithmes de DCT de faible complexité existant dans la littérature et proposé une approche adaptée au cas des réseaux de capteurs.

Dans ce chapitre, nous allons commencer par définir la DCT sous ses deux formes 1D et 2D. Par la suite, nous étudierons différents algorithmes de calcul de la DCT 1D en mettant l'accent sur les algorithmes les plus utilisés. Nous présenterons ensuite les deux principales architectures de la DCT 2D. Finalement, nous développerons une architecture optimisée adaptée aux cas des réseaux de capteurs afin de réduire la complexité calculatoire. Nous verrons bien que les résultats obtenus montrent l'efficacité de l'architecture proposée.

3.2 Les variantes de la DCT

Il y a huit variantes de la transformée discrète en cosinus, transformée en cosinus discrète. Seulement, les plus utilisées sont au nombre de quatre. La plus courante parmi ces quatre est la DCT type II qui est appelée DCT. Son inverse est la DCT type III qui est appelée IDCT [6]. Nous allons présenter ces quatre variantes DCT I, DCT II, DCT III et DCT IV dans ce qui suit.

- **Transformée en cosinus discrète type I (DCT I)**

Les formules des transformations directe et inverse de DCT I sont présentées ci-dessous :

Transformation directe:

$$X[k + 1] = \sqrt{\frac{2}{N}} \alpha_k \sum_{n=0}^{N-1} \alpha_n x[n] \cos \left[\frac{\pi nk}{N} \right] \quad k = 0, \dots, N - 1 \quad (3.1)$$

Transformation inverse :

$$x[n + 1] = \sqrt{\frac{2}{N}} \alpha_n \sum_{k=0}^{N-1} \alpha_k X[k] \cos \left[\frac{\pi nk}{N} \right] \quad n = 0, \dots, N - 1 \quad (3.2)$$

Avec, α_k constante définie comme suit :

$$\alpha_k = \begin{cases} \frac{1}{\sqrt{2}} & \text{si } k = 0 \text{ ou } N \\ 1 & \text{sinon} \end{cases} \quad (3.3)$$

Cette forme de DCT I est définie pour les valeurs N entières supérieures à 2.

La forme orthonormée matricielle de la transformation directe DCT type I peut être exprimée comme suit :

$$[C_{N+1}^I]_{k,n} = \sqrt{\frac{2}{N}} \left[\alpha_k \alpha_n \cos \frac{\pi nk}{N} \right] \quad k, n = 0, 1, \dots, N - 1 \quad (3.4)$$

et celle de sa transformée inverse peut être exprimée comme $[C_{N+1}^I]_{k,n}^{-1} = [C_{N+1}^I]_{k,n}^T$

Ou $[\cdot]_{k,n}^{-1}$ représente la matrice inverse, et $[\cdot]_{k,n}^T$ représente la matrice transposée

- **Transformée en cosinus discrète type II (DCT II)**

Les expressions des transformations directe et inverse de la DCT II sont présentées ci-dessous

Transformation directe:

$$X[k] = \sqrt{\frac{2}{N}} \alpha_k \sum_{n=0}^{N-1} x[n] \cos \left[\frac{\pi(2n+1)k}{2N} \right] \quad k = 0, \dots, N-1 \quad (3.5)$$

Transformation inverse

$$x[n] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} \alpha_k X[k] \cos \left[\frac{\pi(2n+1)k}{2N} \right] \quad n = 0, \dots, N-1 \quad (3.6)$$

Cette forme est définie pour toutes les valeurs N positives. Elle est généralement désignée par DCT c'est la forme la plus utilisée.

La forme orthonormée matricielle de la transformation directe DCT type II peut être exprimée comme suit :

$$[C_N^{II}]_{k,n} = \sqrt{\frac{2}{N}} \left[\alpha_k \cos \frac{\pi(2n+1)k}{2N} \right] \quad k, n = 0, 1, \dots, N-1 \quad (3.7)$$

sa transformée inverse peut être exprimée comme $[C_N^{II}]_{k,n}^{-1} = [C_N^{II}]_{k,n}^T = C_N^{III}$

Ou $[\cdot]_{k,n}^{-1}$ représente la matrice inverse, et $[\cdot]_{k,n}^T$ représente la matrice transposée

- **Transformée discrète en cosinus type III (DCT III)**

Les expressions des transformations directe et inverse de DCT III sont présentées ci-dessous :

Transformation directe:

$$X[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} \alpha_n x[n] \cos \left[\frac{\pi(2n+1)n}{2N} \right] \quad k = 0, \dots, N-1 \quad (3.8)$$

Transformation inverse

$$x[n] = \sqrt{\frac{2}{N}} \alpha_n \sum_{k=0}^{N-1} X[k] \cos \left[\frac{\pi(2n+1)n}{2N} \right] \quad n = 0, \dots, N-1 \quad (3.9)$$

Cette forme est définie pour toutes les valeurs de N positives.

La forme orthonormée matricielle de la transformation directe DCT type III peut être exprimée comme suit :

$$[C_N^{III}]_{k,n} = \sqrt{\frac{2}{N}} \left[\alpha_n \cos \frac{\pi(2n+1)k}{2N} \right] \quad k, n = 0, 1, \dots, N-1 \quad (3.10)$$

Sa transformée inverse peut être exprimée comme suit : $[C_N^{III}]_{k,n}^{-1} = [C_N^{III}]_{k,n}^T = C_N^{II}$

Ou $[\cdot]_{k,n}^{-1}$ représente la matrice inverse, et $[\cdot]_{k,n}^T$ représente la matrice transposée

- **Transformée en cosinus discrète type IV (DCT IV)**

Les expressions des transformations directe et inverse de DCT IV sont présentées ci-dessous :

Transformation directe:

$$X[k] = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x[n] \cos \left[\frac{\pi(2n+1)(2k+1)}{4N} \right] \quad k = 0, \dots, N-1 \quad (3.11)$$

Transformation inverse

$$x[n] = \sqrt{\frac{2}{N}} \sum_{k=0}^{N-1} X[k] \cos \left[\frac{\pi(2n+1)(k+1)}{4N} \right] \quad n = 0, \dots, N-1 \quad (3.12)$$

La forme orthonormée matricielle de la transformation directe DCT type IV peut être exprimée comme suit :

$$[C_N^{IV}]_{k,n} = \sqrt{\frac{2}{N}} \left[\alpha_k \cos \frac{\pi(2n+1)(2k+1)}{4N} \right] \quad k, n = 0, 1, \dots, N-1 \quad (3.13)$$

Sa transformée inverse peut être exprimée comme suit : $[C_N^{IV}]_{k,n}^{-1} = [C_N^{IV}]_{k,n}^T = C_N^{IV}$

Où $[\cdot]_{k,n}^{-1}$ représente la matrice inverse, et $[\cdot]_{k,n}^T$ représente la matrice transposée

L'analyse de ces formes montre les points suivants:

- Les transformées en cosinus de type I et de type IV sont représentables par des matrices involutives dont les inverses sont des matrices de la transformation directe elle-même,
- La transformée inverse de type II est similaire à la transformée directe de type III.

La DCT II est très utilisée en traitement du signal et de l'image et spécialement en compression [6]

3.3 DCT 1D conventionnelle

Pour une séquence d'entrée $x(n)$, n un nombre entier naturel $\in [0, N - 1]$, les coefficients de la DCT (u), u nombre entier naturel $\in [0, N - 1]$, sont donnée par :

$$X(u) = C(u) \sum_{n=0}^{N-1} x(n) \cos \frac{(2n+1)u\pi}{2N} \quad (3.14)$$

Où N est un nombre entier naturel qui représente la taille de la DCT. Cette transformée est réversible permettant de restituer le signal d'origine (temporel). Pour cela il suffit d'appliquer la Transformée en Cosinus Discrète Inverse : TCDI (IDCT « Inverse Discrete Cosine Transform ») au spectre du signal X . Cette transformation inverse est donnée par :

$$x(n) = \sum_{u=0}^{N-1} C(u) X(u) \cos \frac{(2n+1)u\pi}{2N} \quad (3.15)$$

Les constantes de pondération $C(u)$ utilisées dans les équations 3.1 et 3.2 sont données par :

$$C(u) = \begin{cases} \frac{1}{\sqrt{N}} & \text{si } u = 0 \\ \frac{2}{\sqrt{N}} & \text{sinon} \end{cases} \quad (3.16)$$

En utilisant cette définition (équation 3.16) et en remplaçant $C(0)$ par sa valeur dans l'équation 3.14, le coefficient DCT à la fréquence 0 est donné par $X(0) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n)$. Ainsi, la première valeur de la transformée représente la somme de toutes les entrées pondérées par $\frac{1}{\sqrt{N}}$. Cette valeur est appelée DC (Direct Component) et représente l'offset de la transformée. Toutes les autres valeurs sont nommées coefficients AC (Alternative Component)

Le calcul des coefficients de la DCT peut se faire avec une combinaison linéaire d'un ensemble de fonctions définies par :

$$F_n(u) = C(u) \cos \frac{(2n+1)u\pi}{2N} \quad (3.17)$$

Où $u \in [0, N-1]$,

Ces fonctions sont appelées les fonctions de base. L'équation 3.14 peut donc s'écrire sous la forme :

$$X(u) = \sum_{n=0}^{N-1} x(n) F_n(u) \quad (3.18)$$

L'avantage d'une telle décomposition réside dans le fait que ces fonctions, indépendantes de $x(n)$, peuvent être pré-calculées à l'avance. Ces fonctions sont appelées également noyaux de la transformation. Ainsi, le calcul des coefficients de la DCT s'effectue en réalisant une combinaison linéaire entre le signal d'entrée $x(n)$ et ces fonctions. Notons que toutes ces fonctions sont orthogonales. Les fonctions orthogonales sont indépendantes, ainsi aucune de ces fonctions ne peut être obtenue par une combinaison linéaire des autres. La forme matricielle associée à l'équation (3.18) est la suivante :

$$X_{N \times 1} = F_{N \times N} \cdot x_{N \times 1} \quad (3.19)$$

$F_{N \times N}$ est la matrice associée à (3.17), elle est définie comme suit :

$$F_{N \times N} = [C_N^H]_{u,n} = \left[C(u) \cos \frac{(2n+1)u\pi}{2N} \right] \quad u, n = 0, 1, \dots, N-1 \quad (3.20)$$

Enfin, notons que si la taille de la séquence d'entrée est supérieure à N , elle peut être divisée en plusieurs sous-séquences chacune de taille N . Par la suite la DCT peut être obtenue en utilisant les sous-DCT calculées avec les sous-séquences.

3.4 DCT 2D conventionnelle :

3.4.1. Définitions

La DCT 2D est une transformation très utilisée dans les applications de traitement d'images et de vidéo comme la compression et le codage. Le calcul des coefficients $Y(u, v)$ d'une matrice d'entrée $y(i, j)$ peut se faire par l'équation 3.21.

$$Y(v, u) = \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} \frac{C(v)}{2} \frac{C(u)}{2} y(i, j) \cos \frac{(2i+1)u\pi}{2N} \cos \frac{(2j+1)v\pi}{2N} \quad (3.21)$$

$C(u)$ et $C(v)$ sont définis par l'équation (3.16) et u, v, i, j sont des entiers naturels $\in [0, N-1]$.

La matrice d'entrée $y(i, j)$ peut être recalculée à partir de $Y(u, v)$ en utilisant la DCT 2D inverse IDCT 2D définie par :

$$y(i, j) = \sum_{v=0}^{N-1} \sum_{u=0}^{N-1} \frac{C(v)}{2} \frac{C(u)}{2} Y(v, u) \cos \frac{(2i+1)u\pi}{2N} \cos \frac{(2j+1)v\pi}{2N} \quad (3.22)$$

De la même manière que pour la DCT 1D, la DCT 2D peut être écrite sous la forme d'une combinaison linéaire d'un ensemble de fonctions appelées fonctions de base. Le calcul de ces fonctions est réalisé par l'équation (3.23).

$$F_{i,j}(u, v) = \frac{C(v)}{2} \frac{C(u)}{2} \cos \frac{(2i+1)u\pi}{2N} \cos \frac{(2j+1)v\pi}{2N} \quad (3.23)$$

Par conséquent, l'équation permettant de calculer les coefficients de la DCT 2D est donnée par :

$$Y(v, u) = \sum_{j=0}^{N-1} \sum_{i=0}^{N-1} F_{i,j}(u, v) y(i, j) \quad (3.24)$$

Les fonctions de base de la DCT 2D peuvent être générées à partir des fonctions de base 1D (voir (3.17)). Ceci est assuré grâce à la propriété de séparabilité du noyau de la transformation de la DCT2D.

3.4.2 Propriétés de la DCT

Dans cette section nous soulignons quelques propriétés de la transformée en cosinus discrète et en particulier celles qui sont importantes pour la compression d'image.

Dé-corrélation

La DCT-II bidimensionnelle est la transformée de dé-corrélation utilisée dans les standards de codage international de l'image et de la vidéo. Le but de la compression est de minimiser la redondance de l'information ceci revient à minimiser la corrélation des coefficients du signal.

La transformation de Karhunen-Loève (KLT) est la transformation dite optimal. C'est à dire elle dé-corrèle les coefficients complètement. La DCT est la transformation la plus proche de la transformation KLT (optimal) dans la dé-corrélation des données [132].

Concentration de l'énergie

L'efficacité d'une transformée est mesurée par sa capacité de concentrer l'énergie des données dans le plus petit ensemble de coefficients possible. La DCT est capable de concentrer l'énergie du signal dans les premiers coefficients DCT donc dans les basses fréquences. Ceci permet au quantificateur d'éliminer les coefficients de haute fréquence généralement de faibles amplitudes sans, ou en introduisant de très faibles, distorsions lors de la reconstruction de l'image. Même avec de faibles distorsions elles seront non visible à l'œil nu (en utilisant un simple filtrage passe-bas).

Séparabilité

Une transformée orthogonale est séparable si son noyau vérifie les conditions suivantes:

$$F(k_1, k_2, n_1, n_2) = F_1(k_1, n_1)F_1(k_2, n_2) \quad (3.25)$$

Où F est le noyau de la transformation 2D et F_1 le noyau de la transformation 1D

Ainsi, l'équation (3.14) de la transformée en cosinus discrète peut aussi s'écrire de la façon suivante:

$$\begin{aligned} X[k_1, k_2] \\ = \sqrt{\frac{2}{N_1 N_2}} \alpha_{k_1} \alpha_{k_2} \sum_{n_1=0}^{N_1-1} \cos \left[\frac{\pi(2n_1 + 1)k_1}{2N_1} \right] \sum_{n_2=0}^{N_2-1} x[n_1, n_2] \cos \left[\frac{\pi(2n_2 + 1)k_2}{2N_2} \right] \end{aligned} \quad (3.26)$$

Donc, la transformée en cosinus discrète est séparable

Cette propriété a pour avantage de calculer la DCT 2D en utilisant des DCT 1D. Ceci, en commençant par des transformations 1D de toutes les lignes, puis des transformations 1D de toutes les colonnes successivement comme nous allons voir plus loin dans ce chapitre.

L'équation (3.15) de la transformée discrète en cosinus inverse peut aussi être calculée de la même manière:

$$\begin{aligned} x[n_1, n_2] \\ = \sqrt{\frac{2}{N_1 N_2}} \sum_{k_1=0}^{N_1-1} \alpha_{k_1} \cos \left[\frac{\pi(2n_1 + 1)k_1}{2N_1} \right] \sum_{k_2=0}^{N_2-1} \alpha_{k_2} X[n_1, n_2] \cos \left[\frac{\pi(2n_2 + 1)k_2}{2N_2} \right] \end{aligned} \quad (3.27)$$

Donc, la transformée discrète en cosinus inverse est aussi séparable.

Symétrie

La transformation est symétrique si elle est séparable et satisfait la condition suivante:

$$X_1(k_1, n_1) = X_2(k_2, n_2) \quad (3.28)$$

Grace aux propriétés de symétrie et de séparabilité, les DCT 2D, directe (3.21) et inverses (3.22) peuvent être exprimées respectivement comme suit:

$$Y_{N \times N} = F_{N \times N} \cdot x_{N \times N} \cdot F_{N \times N}^T \quad (3.29)$$

$$x_{N \times N} = F_{N \times N}^T \cdot Y_{N \times N} \cdot F_{N \times N} \quad (3.30)$$

Où $F_{N \times N}$ étant la matrice noyau de la DCT, donnée par l'équation (3.20). $x_{N \times N}$ est la matrice de données et $(.)^T$ est la transposée.

Ces propriétés permettent de calculer au préalable, la matrice noyau de la transformée en offline, puis de l'appliquer à l'image, ce qui produirait une amélioration de l'efficacité de calcul.

Orthogonalité

La transformation est dite orthogonale, si et seulement si la matrice noyau de la transformée est orthogonale. Autrement dit si toutes les colonnes de la matrice F sont orthogonales les unes aux autres et si toutes les lignes sont orthogonales les unes aux autres. Ceci revient à écrire en représentation matricielle :

$$F^T = F^{-1} \quad (3.31)$$

La transformée DCT est aussi orthonormée, cela revient à écrire :

$$F^T F = I \quad (3.32)$$

Ces propriétés sont importantes. Comme le calcul de la matrice inverse se réduit au calcul du simple transposée, cela réduit énormément la complexité de calcul.

Une autre conséquence de la propriété de l'orthogonalité est la conservation de l'énergie. Cette dernière propriété est formulée par le **théorème de Parseval** comme suit :

$$\sum_i^{N-1} |X_i|^2 = \sum_j^{N-1} |x_j|^2 \quad (3.33)$$

3.5 Intérêt de l'utilisation de la DCT :

La DCT comme la DFT permet de transformer un signal temporel ou plan spatial en un signal fréquentiel ou plan spectral respectivement. Evidemment, il s'agit de deux algorithmes différents et par conséquent le calcul des coefficients fréquentiels diffère selon l'algorithme utilisé : la DFT ou DCT. Une des principales différences entre la DFT et la DCT réside au

niveau du nombre d'opérateurs nécessaires pour le calcul de l'une ou de l'autre transformation. Cependant, dans les deux cas, toutes les multiplications utilisées sont des multiplications par des constantes. Dans le premier cas, pour $N = 8$, le calcul des coefficients de la DFT nécessite entre autres deux multiplications par la constante $\cos \frac{\pi i}{4}$ et deux multiplications par la constante $-\cos \frac{\pi i}{4}$ [44]. Cependant, les architectures optimisées de la DCT requièrent entre 11 et 16 multiplications. Par conséquent, le calcul des coefficients de la DCT est plus complexe que celui de la DFT dans le sens où il nécessite plus d'opérateurs. Malgré cela, pour des applications de compression : les standards JPEG, MPEG et H.26x ont privilégié l'utilisation de la DCT plutôt que la DFT. Afin de comprendre les raisons de ce choix une comparaison succincte entre l'efficacité de restitution d'un signal avec les deux transformations DCT/ IDCT « Inverse Discrete Cosine Transform » et DFT/IDFT « Inverse Discrete Fourier Transform » a été présentée dans [133] et reproduite ici. Soit un signal sous forme d'une rampe défini sur 8 points de 8 bits chacun.

Les valeurs de sortie de la DCT et de la DFT appliquées à ce signal sont listées dans le tableau 3.1.

indice	x(n)	DCT(x(n))	DFT(x(n))
1	0	356.38	1008
2	36	-231.92	-144+347j
3	72	0	-144+144j
4	108	-24.24	-144+59j
5	144	0	-144
6	180	-7.23	-144-59j
7	216	0	-144-144j
8	252	-1.82	-144-347j

Tableau 3.1 Valeurs des transformées d'une rampe

Avec j un nombre complexe vérifiant $j^2 = -1$

Pour pouvoir comparer l'efficacité de restitution de la DCT et de la DFT, nous n'utilisons que quelques valeurs des coefficients (Dans le domaine spectral) de ces deux transformations. Les autres valeurs sont fixées à zéro. Ceci constitue une modélisation très grossière de l'opération

de quantification utilisée dans les applications de compression. Ainsi, dans cet exemple nous n'avons considéré que 3 valeurs dans le domaine spectral (les valeurs les plus importantes en intensité) comme indiqué dans le tableau 3.2.

indice	$x(n)$	$\widehat{DCT}(x(n))$	$\widehat{IDCT}(\widehat{DCT}(x(n)))$	$\widehat{DFT}(x(n))$	$\widehat{IDFT}(\widehat{DCT}(x(n)))$
1	0	356.38	2.28	1008	90+61j
2	36	-231.92	31.92	-144+347j	64
3	72	0	73.34	-144+59j	100-36j
4	108	-24.24	110.04	0	126-25j
5	144	0	141.95	0	126-25j
6	180	0	178.65	0	151-36j
7	216	0	220.07	0	187
8	252	0	249.71	0	162+61j

Tableau 3.2 Valeurs des transformées et des transformées inverses d'une rampe

Nous pouvons, facilement, remarquer que la DCT/IDCT donne une très bonne approximation du signal original. En effet, avec la DCT/IDCT seulement avec 3 coefficients spectraux, nous avons réussi à restituer le signal d'origine avec une erreur de reconstruction (différence entre le signal $x(n)$ et le signal reconstruit après transformation) comprise entre $[-12, 12]$. Cependant, la restitution du signal d'origine avec DFT/IDFT introduit une erreur souvent complexe qui varie de $[-108, 78]$.

En effet, La DCT permet de localiser l'information utile dans les basses fréquences. Par conséquent, il est possible d'utiliser seulement cette information utile pour retrouver le signal d'origine. Cette propriété est très utile dans les applications de compression où nous essayons de minimiser la quantité de l'information à traiter, à stocker et/ou à transmettre. Ainsi, l'intérêt de la DCT comparé à celui de la DFT réside dans la capacité à restituer un signal avec peu de coefficients spectraux. Ceci peut expliquer son utilisation massive dans les applications de compression d'images.

La DCT comme la DFT sont des transformations orthogonales discrètes. Autrement dit, pour une séquence d'entrée discrète de N échantillons dans le domaine direct (temporel pour les signaux et spatial pour les images) ces transformations vont nous donner une séquence de

sortie discrète également de N coefficients dans le domaine fréquentiel. En tenant compte de la propriété qui dit qu'une discrétisation dans un domaine (direct ou fréquentiel) conduit à une périodisation dans l'autre domaine (fréquentiel ou direct) alors les transformations discrètes de type DCT et DFT vont périodiser les séquences d'entrée et de sortie simultanément.

Dans le cas de la DFT cette périodisation dans le domaine direct va s'accompagner inévitablement par des discontinuités entre le dernier échantillon d'une période et le premier échantillon de la période suivante. Ceci se traduira donc par des hautes fréquences relativement significatives. Autrement dit, le compactage de l'énergie ne sera pas optimal.

Par contre dans le cas de DCT il y'aura une symétrisation du signal avant sa périodisation. Ceci enlève toute discontinuité entre les différentes périodes. Ainsi, les coefficients de la DCT seront mieux compactés dans les basses fréquences (voir figure 3.1).

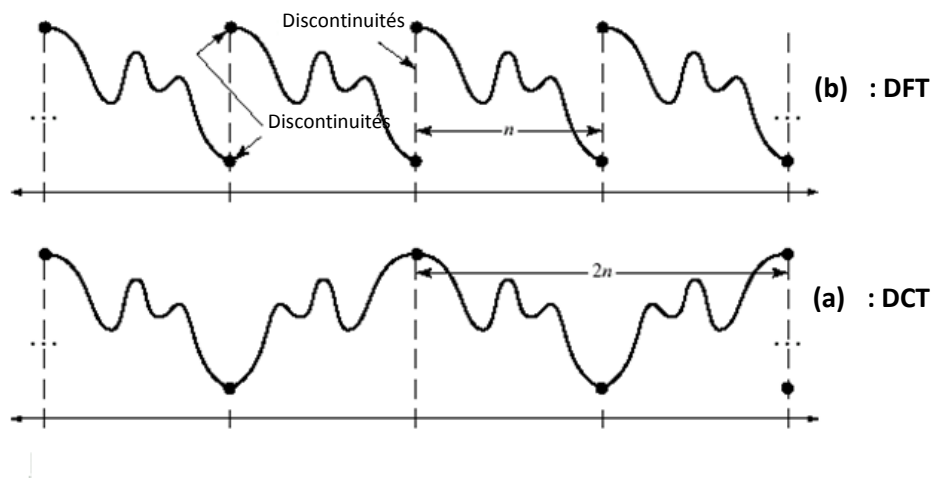


Figure 3.1: *DFT vs DCT*

(a) : *Effet de la périodisation dû à la DFT*

(b) : *Effets de la symétrisation et périodisation dus à la DCT*

3.6 Algorithmes rapides pour le calcul de la DCT 1D exacte:

La réalisation directe de la DCT nécessite un nombre important d'opérateurs arithmétiques ($N \times (N - 1)$) additions et N^2 multiplications [132]. Afin de proposer une architecture efficace de la DCT qui se prête bien dans les systèmes numériques embarqués (nœuds

capteurs par exemple), plusieurs algorithmes ont été proposés pour réduire la complexité calculatoire de la DCT.

Nous nous sommes intéressés plus particulièrement à des architectures permettant l'approximation de la DCT afin de réduire la consommation énergétique, le temps de traitements et par conséquent augmenter la durée de vie du nœud capteurs. Dans le but d'atteindre cet objectif, nous étudions les différents algorithmes proposés et validés dans la littérature. Ces algorithmes peuvent être classés en plusieurs catégories selon les méthodes de calcul employées. La première catégorie englobe tous les algorithmes réduisant le nombre d'opérations pour calculer les coefficients DCT. Ces algorithmes calculent la DCT exacte. La deuxième catégorie englobe les algorithmes qui approximent la DCT. Deux types de cette catégorie existent: le premier remplace la transformée originale avec une autre de faible précision [101]. Le deuxième type c'est l'approche zonale ('Pruned') [134-135].

Dans cette section nous présenterons les principaux algorithmes de référence pour le calcul de la DCT exacte rapide.

3.6.1 Calcul récursif : [136-137]

La récursivité peut être utilisée pour calculer les coefficients d'une DCT de grande taille à partir de plusieurs blocs de DCT de petites tailles. Plusieurs algorithmes basés sur cette méthode ont été proposés. Nous pouvons citer par exemple l'algorithme proposé par Hou [136]. Cet algorithme consiste à utiliser une méthode récursive permettant le calcul des coefficients d'une DCT de taille N points à partir de deux blocs de DCT de taille $\frac{N}{2}$. En effet, pour un nombre d'entrées, sous forme d'une puissance de deux, les coefficients, les coefficients de la DCT se calculent sur 2 parties. Une première utilise les éléments d'indices pairs tandis que la seconde partie utilise les éléments d'indices impairs. Ce principe peut se produire jusqu'à $N = 2$.

Pour $N = 8$, cet algorithme permet de calculer les coefficients de la DCT en utilisant 12 multiplieurs et 29 additionneurs [136]. Cependant, cette méthode nécessite plusieurs éléments de stockage ainsi que des éléments de contrôle. En effet, le résultat obtenu à la fin de chaque étage doit être sauvegardé dans des mémoires ou des registres supplémentaires. De plus, elle nécessite l'utilisation de plusieurs éléments de contrôle (voir figure 3.2).

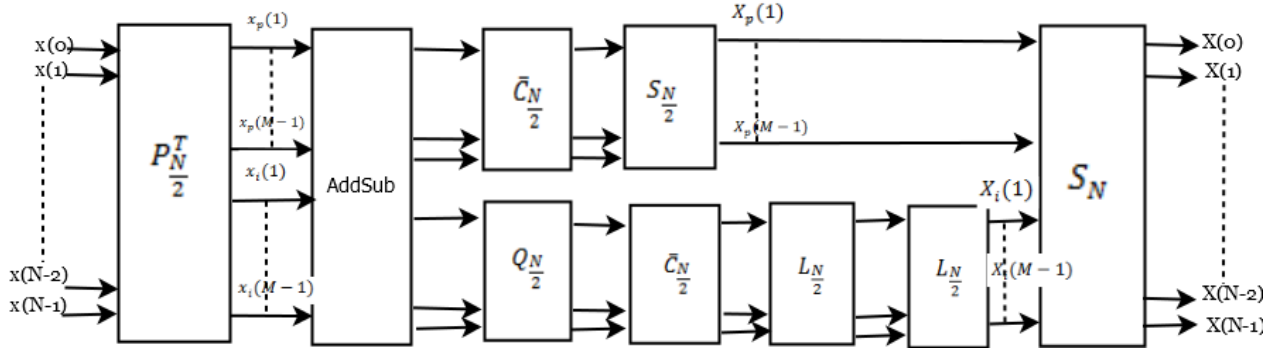


Figure 3.2 : Principe de la méthode récursive.

P, C, Q et S sont des matrices définies selon [136]

3.6.2 Calcul indirect : [138-141]

Cette méthode utilise la transformée de Fourier rapide (FFT) et la transformée WHT « Walsh-Hadamard Transform » pour le calcul des coefficients de la DCT. En effet, les auteurs de [142] ont montré que le calcul de la DCT peut être effectué en utilisant deux blocs de calcul de FFT N points. Dans [139], les auteurs ont proposé un algorithme permettant d'obtenir la DCT en calculant uniquement la partie réelle des premiers N coefficients de la DFT de $2N$ points.

Considérant un signal $x(u)$, avec $m \in [0, N - 1]$, Alors nous pouvons calculer les coefficients de la DCT 1D de ce signal (de taille N) à partir d'une DFT de taille $2N$ points comme suit :

$$X(u) = C'(u)X'(u) \quad (3.34)$$

$C'(u)$ est donné comme suit :

$$C'(u) = \frac{1}{2} e^{-j\frac{\pi}{2N}u} C(u) \quad (3.35)$$

$C(u)$ est donné par l'équation (3.16)

$X'(u)$ est la DFT de taille $2N$ appliquée au signal $x'(m)$. Elle est donnée par :

$$X'(u) = \sum_{m=0}^{2N-1} x'(m) e^{-j\frac{2\pi}{2N}um} \quad (3.36)$$

$x'(m)$ est le signal dupliqué. Il est donné par l'expression suivante :

$$x'(m) = \begin{cases} x(m_1) & \text{pour } 0 \leq m_1 \leq N - 1 \\ x(2N - m_1 - 1) & \text{pour } N \leq m_1 \leq 2N - 1 \end{cases} \quad (3.37)$$

Le schéma bloc présenté dans la Figure 3.2 illustre le synoptique de cette méthode.

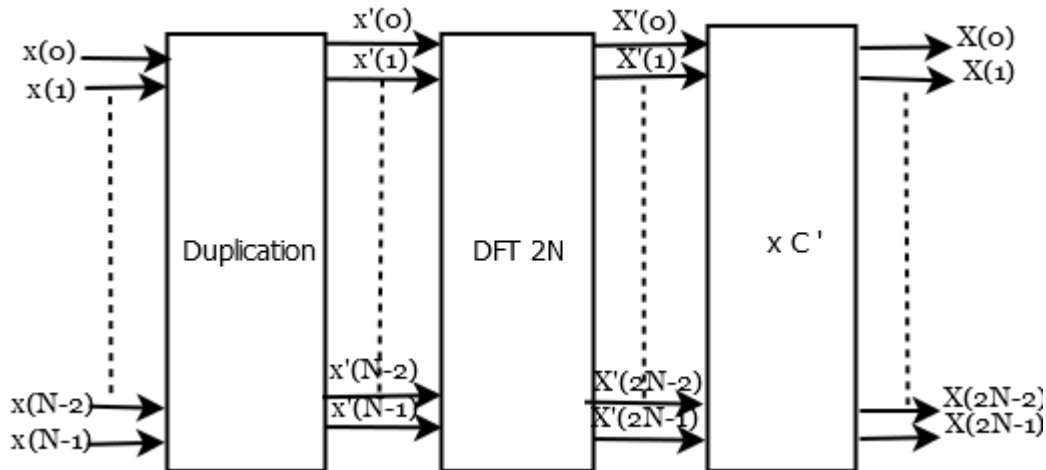


Figure 3.3 Principe de la méthode de calcul indirecte

En utilisant cette méthode, les coefficients d'une DCT de taille 8 points peuvent être obtenus en utilisant 12 multiplieurs et 29 additionneurs [138]. Cependant, l'inconvénient majeur de cette méthode est l'utilisation aussi des mémoires supplémentaires. Ces derniers sont nécessaires pour sauvegarder le signal en entrée de la DFT de taille $2N$.

3.6.3 DCT à base de convolution : [143-144]

Cette méthode est basée sur le principe de reformulation des nombres premiers en un certain nombre de convolutions cycliques. Elle ne s'applique que pour un nombre N impair. Dans cette optique, les auteurs de [143] ont proposé une architecture de calcul de la DCT à base de deux convolutions cycliques similaires. Une deuxième architecture a été proposée dans [144]. Elle permet de calculer une DCT de N points à base de 4 convolutions cycliques de taille $\frac{N}{4}$.

Le nombre d'opérateurs utilisés par cette méthode est de $\frac{N}{2} + 3$ multiplieurs et $\frac{N}{2} + 6$ additionneurs [144]. En effet, les auteurs de la référence [144] ont utilisé le principe des multiplieurs à base de mémoires afin de réduire d'avantage le nombre d'opérateurs requis. Ce nombre est ainsi inférieur à celui utilisé par les autres méthodes. Cependant, cette méthode n'est applicable que pour un nombre de points N impair. Ceci peut constituer un inconvénient pour des applications de compression où N est généralement égale à 4, 8, 16 ou 32. Les

standards de compression fixent un nombre pair pour des tailles de blocs = $4 \times 4, 8 \times 8, 16 \times 16$, etc.

3.6.4 Factorisation directe [145-149]

Le calcul de la DCT, en utilisant la factorisation directe, se base sur la propriété de symétrie de la matrice des coefficients de la DCT. L'avantage principal de cette méthode réside dans l'utilisation d'un nombre réduit d'additionneurs ou de multiplieurs. Cette méthode permet de calculer la DCT de tout vecteur de taille N qui s'écrit sous la forme d'une puissance de 2. Parmi les algorithmes basés sur cette méthode, on peut citer l'algorithme de Lee [146], de Wang [148], de Chen [145] et de Loeffler [100]. En effet, Wang [148] a proposé un algorithme optimisé pour le calcul des coefficients de la DCT 1D. Cet algorithme permet aussi d'obtenir la transformée en sinus discrète (DST), La transformée en ondelettes discrète DWT « Discrete Wavelet Transform » et la transformée de Fourier discrète (DFT) à partir de son algorithme pour la DCT. Une version optimisée de cet algorithme a été proposée par Suehiro [147]. Cette optimisation a permis de réduire le nombre de multiplieurs utilisés pour le calcul de la DCT ainsi que pour les autres transformées déduites.

Par ailleurs, dans cette famille d'algorithmes à base de factorisation directe, l'algorithme de Chen [145] est l'un des algorithmes les plus utilisés dans la littérature. Cet algorithme se base sur le principe de la factorisation matricielle afin de réduire le nombre d'opérateurs arithmétiques, tout en gardant un nombre réduit d'éléments de contrôle et de mémorisation. En effet, l'algorithme de Chen permet d'obtenir les coefficients d'une DCT de taille 8 points en utilisant 16 multiplieurs et 26 additionneurs et un nombre réduit d'éléments de contrôle. Il présente ainsi le nombre le plus réduit d'additionneurs.

Un autre algorithme, très utilisé dans la littérature, basée sur la méthode de factorisation directe est l'algorithme de Loeffler [100]. La principale raison du succès de cet algorithme est le nombre réduit de multiplieurs requis. En effet, l'algorithme de Loeffler permet de calculer les coefficients de la DCT de taille 8 points en utilisant uniquement 11 multiplieurs et 29 additionneurs. Le nombre de multiplieurs ainsi proposé représente la limite théorique pour le calcul des coefficients de la DCT introduite par Duhamel [151]. Un autre avantage de cet algorithme vient du fait que le nombre d'éléments de contrôle utilisé par cet algorithme est réduit.

Les auteurs dans [152] ont présenté une comparaison entre ces différents algorithmes. Le tableau 3.3 résume cette comparaison.

Auteur	Chen[145]	Wang[148]	Lee[146]	Suehiro[147]	Hou[136]	Vetterli[138]	Loeffler[100]
Mult	16	13	12	12	12	12	11
Add	26	29	29	29	29	29	29

Tableau 3.3 Comparaison entre les différents algorithmes en termes de nombre d'opérations

Cette comparaison montre que la méthode basée sur la factorisation directe et celle basée sur le calcul récursif. De plus, bien que le nombre d'opérateurs utilisés par la méthode basée sur le calcul indirect soit équivalent à celui utilisé par la méthode basée sur la factorisation directe, cette dernière présente l'avantage de ne pas utiliser d'éléments de mémorisations et de logique séquentielle. En effet, la première étape de la méthode basée sur le calcul indirect consiste à dupliquer le signal d'entrée afin d'appliquer la DFT. Cette duplication nécessite l'utilisation de mémoires supplémentaires. En outre, la division du signal d'entrée en deux parties et les différents post-traitements nécessitent plusieurs éléments séquentiels. Malgré le fait que la méthode à base de convolution utilise un nombre réduit d'opérateurs, l'utilisation de cette méthode est limitée en vue qu'elle n'est applicable que pour un nombre N impair. Cette contrainte rend son utilisation ambiguë pour les nouveaux standards de compression.

De plus, comme nous l'avons cité précédemment, les algorithmes de Loeffler [100] et de Chen [145] sont des algorithmes basés sur la méthode de calcul direct. Ces algorithmes sont les plus utilisés dans le calcul des coefficients de la DCT nécessitant qu'un nombre réduit d'éléments de contrôle et de mémorisation.

Le schéma bloc de la DCT de taille 8 points basée sur l'algorithme de Loeffler est illustré dans la Figure 3.4.

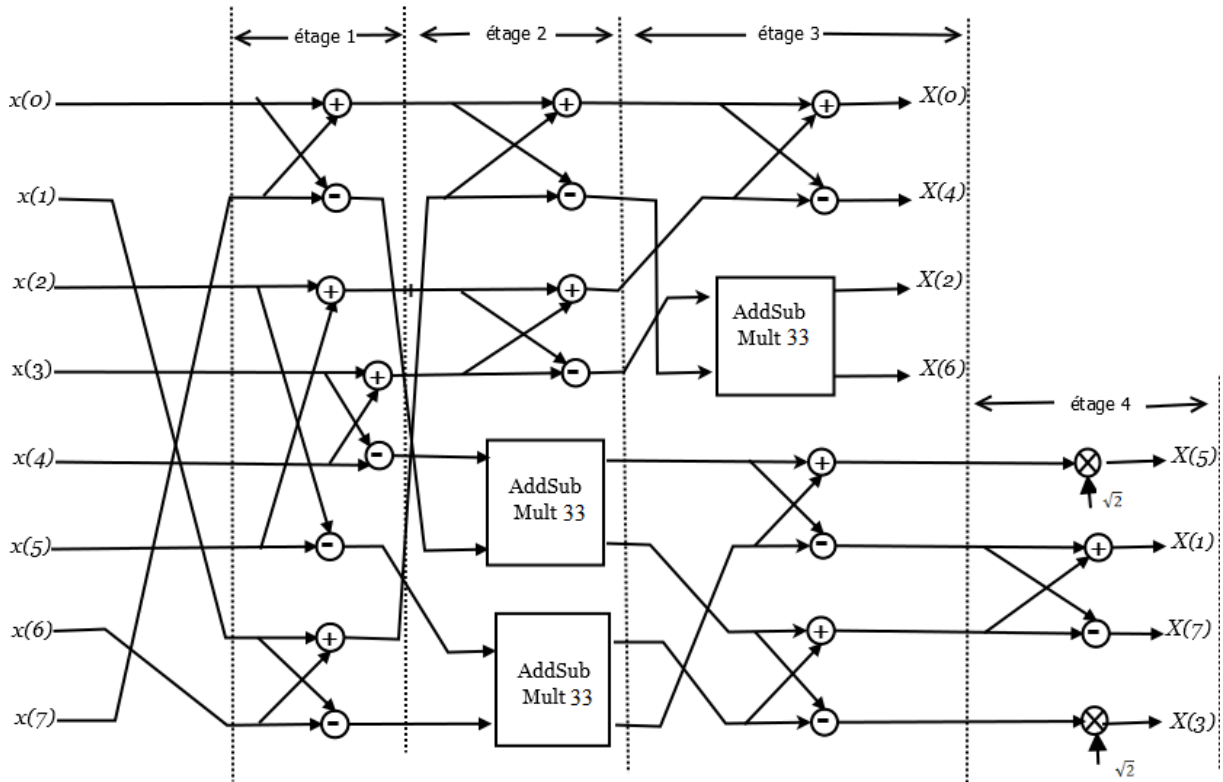


Figure 3.4 : Schéma bloc de l'algorithme de Loeffler 8 points

Le bloc AddSubMult33 utilise 3 multiplieurs et 3 additionneurs

Au total, l'algorithme de Loeffler consomme 10 blocs composé chacun d'un additionneur et un soustracteur, 3 blocs de AddSubMult33 et 2 blocs de multiplications par constante. Par conséquent, le nombre total d'opérateurs arithmétiques utilisés par l'algorithme de Loeffler est 11 multiplieurs et 29 additionneurs.

L'algorithme de Chen [145] est parmi les premiers algorithmes qui ont été proposés pour le calcul de la DCT. Tout comme l'algorithme de Loeffler, l'algorithme de Chen permet la réalisation de la DCT de façon simple et économique en termes de ressources. Cet algorithme possède une architecture structurée et utilise 26 additionneurs. Ce dernier représente le nombre minimum d'additionneurs pour une DCT de taille 8 points.

Le schéma bloc de la DCT de taille 8 points basée sur l'algorithme de Chen est illustré dans la Figure suivante.

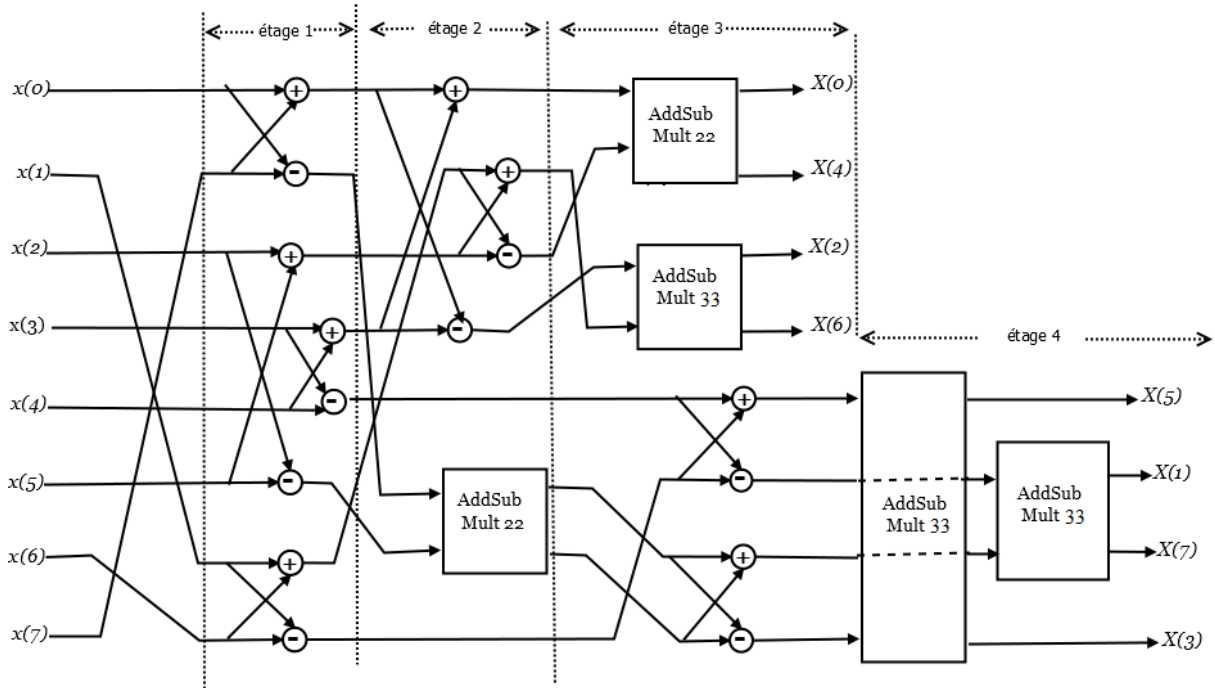


Figure 3.5 Schéma bloc de l'algorithme de Chen 8 points

Notons que le bloc AddSubMult22 utilise 2 additionneurs et 2 multiplieurs

3.7 Architectures pour la DCT 2D :

Les transformées DCT 2D directe et inverse permettent de passer du domaine bidimensionnel spatial au domaine bidimensionnel fréquentiel et vice versa. Pour une matrice d'entrée y de taille $N \times N$ la DCT 2D est donnée par l'équation 3.21. Cette équation peut être écrite sous la forme matricielle comme suit :

$$Y = AyA^T \tag{3.38}$$

Où $A_{ij} = C(i) \cos \frac{(2j+1)i\pi}{2N}$

La réalisation de la DCT 2D est basée sur l'algorithme de la DCT 1D. Comme nous l'avons vu précédemment, les algorithmes de la DCT 1D ont été largement étudiés dans les années 1970 par Loeffler [100] et dans les années 1980 par Chen [145]. A l'époque, l'enjeu était de réaliser la transformée sur une surface très réduite puisque le silicium était très cher. Aujourd'hui, la technologie a évolué et les enjeux se sont déplacés vers la rapidité, la réduction de la puissance et la flexibilité tout en gardant une surface acceptable. Pour cette raison, nous avons étudié les algorithmes de Chen et de Loeffler qui sont les plus

économiques en surface et nous nous efforçons à trouver les moyens algorithmiques et architecturaux pour une réalisation efficace de la DCT 2D.

3.7.1 Réalisation à base de séparation ligne colonne :

Le nombre important d'opérations requis pour des grandes valeurs de N a amené les chercheurs à diviser la matrice en des blocs de tailles plus petites (2×2 , 4×4 , 8×8 ou 16×16). Pour une image d'entrée de taille $N \times N$, et après division en plusieurs blocs de petites tailles, le principe de la DCT 2D basée sur la séparation ligne colonne consiste à :

- (1) appliquer l'algorithme de la DCT 1D sur chaque ligne
- (2) appliquer la DCT 1D par la suite sur chaque colonne. Une illustration de ce principe est donnée sur la Figure 3.6.

En effet, la transformation en colonne peut être exprimée par :

$$Y(v, u) = \sum_{j=0}^{N-1} X(v, j) \cos \frac{(2j+1)u\pi}{2N} \quad (3.39)$$

Où $X(v, j)$ est la transformation en ligne. Cette transformation est exprimée par :

$$X(v, j) = \sum_{i=0}^{N-1} y(v, i) \cos \frac{(2i+1)v\pi}{2N} \quad (3.40)$$

Où $X(v, j)$ représente la transformée ligne de la matrice d'entrée.

La réalisation de la DCT à base de la séparation ligne colonne est la plus adaptée pour une implantation matérielle de la DCT 2-D. L'implémentation de cette architecture présentée sur la Figure 3.14 est composée de trois étages. Le premier étage consiste à utiliser une première DCT qui sera appliquée à chaque vecteur ligne de la matrice d'entrée. Dans le deuxième étage, le résultat obtenu sera sauvegardé dans une mémoire intermédiaire afin d'effectuer la transposée de la matrice résultante. Le dernier étage est composé d'une DCT 1D à appliquer sur chaque ligne de la mémoire.

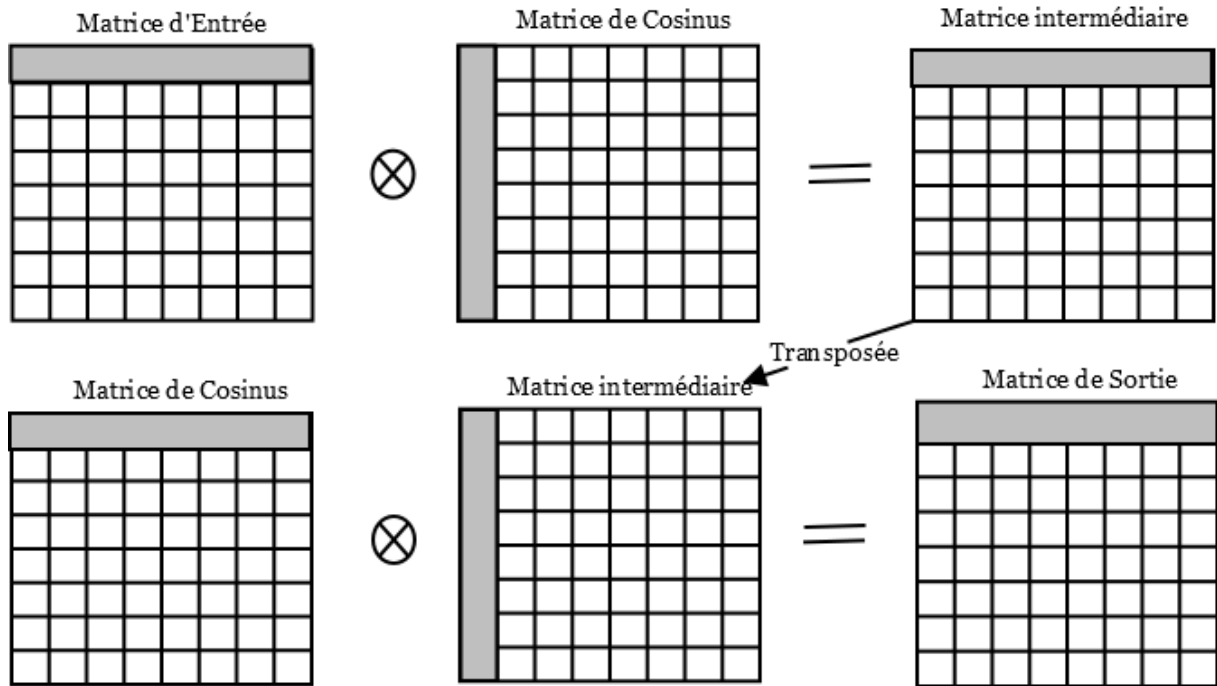


Figure 3.6 Principe de la réalisation ligne colonne de la DCT

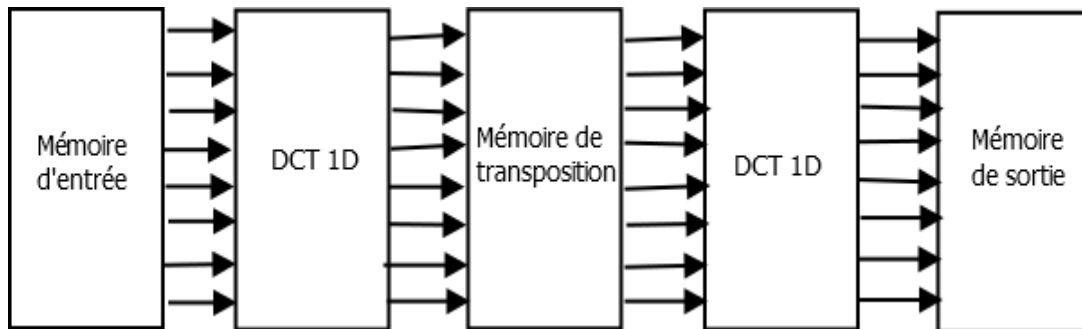


Figure 3.7 Schéma bloc de la réalisation matérielle de la séparation ligne colonne

3.7.2 Réalisation directe :

La deuxième approche permettant l'implémentation de la DCT 2D consiste à effectuer une réalisation directe en se basant sur la multiplication matricielle présente dans l'équation 3.38. Pour simplifier la représentation matricielle, prenons l'exemple où $N = 4$, l'équation 3.38 devient :

$$Y = AyA^T = \begin{pmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{pmatrix} \begin{pmatrix} y(0,0) & y(0,1) & y(0,2) & y(0,3) \\ y(1,0) & y(1,1) & y(1,2) & y(1,3) \\ y(2,0) & y(2,1) & y(2,2) & y(2,3) \\ y(3,0) & y(3,1) & y(3,2) & y(3,3) \end{pmatrix} \begin{pmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{pmatrix} \quad (3.41)$$

La matrice A est représentée par 3 éléments à savoir : $a = \frac{1}{3}$, $b = \frac{1}{\sqrt{2}} \cos(\frac{\pi}{8})$ et $c = \frac{1}{\sqrt{2}} \cos(\frac{3\pi}{8})$. La réalisation directe de cette multiplication matricielle nécessite un nombre élevé d'opérations arithmétiques (additions et multiplications). En effet, pour une matrice de taille N, le nombre d'opérations nécessaires pour réaliser l'équation 3.38 est de $2N^2(N - 1)$ additions. Par conséquent, pour $N = 4$, cette équation nécessite 128 multiplications et 96 additions.

3.8 Réduction des coefficients à calculer : DCT Zonale

La taille typique d'une transformation est de 8×8 pixels. Une transformée de plus grande taille, par exemple 16×16 , n'est pas aussi efficace qu'une de 8×8 dans sa capacité de concentration d'énergie [89]. Le tableau 3.4 donne le pourcentage des coefficients DCT contenant plus de 99 % de l'énergie totale de la transformée pour plusieurs tailles : 4×4 , 8×8 et 16×16 pour l'image 'Cameraman' [89].

Taille des blocs	Pourcentage des coefficients contenant > 99% de l'énergie totale
4×4	62.5
8×8	65.63
16×16	68.75

Tableau 3.4 : Pourcentage de coefficients contenant plus de 99% de l'énergie totale

La DCT 4×4 est la plus efficace dans sa capacité de concentration d'énergie. En plus, elle possède la complexité de calcul la plus réduite. Le tableau 3.5 donne le nombre d'opérations nécessaires pour calculer la DCT réelle en 2-D et en utilisant la décomposition ligne colonne pour les différentes tailles standards.

Taille des blocs	Complexité de calcul	
	Add	Mul
4 × 4	96	128
8 × 8	896	1024
16 × 16	7680	8192

Tableau 3.5 : Complexité de calcul de la DCT pour différentes tailles

La DCT 4 × 4 possède la complexité de calcul la plus réduite. Des économies considérables d'environ 89 % et de 98 % (pour chaque type d'opérations) comparé à la DCT 8 × 8 et 16 × 16 respectivement.

Cependant, les effets de blocs sont beaucoup plus prononcés lors de l'utilisation des tailles de 4 × 4, ce qui réduit l'efficacité en termes de PSNR. Par conséquent, la taille 8 × 8 est préférable et plus adoptée dans les différents standards de compression d'image et vidéo.

Les algorithmes de calcul de la DCT basés sur l'approche de factorisation de matrice, génèrent à partir de N entrées le même nombre de sorties (N coefficients DCT). Mathématiquement, cela revient à calculer N coefficients DCT selon l'équation :

$$Y_{N \times 1} = A_{N \times N} \cdot X_{N \times 1} \quad (3.42)$$

où $X_{N \times 1}$ est le vecteur de données, $A_{N \times N}$ est la matrice transformation et $Y_{N \times 1}$ est le vecteur transformé. Dans une image, l'information importante de l'image est essentiellement concentrée dans les coefficients de basses fréquences (voir tableau 3.5). Compresser revient donc à ne garder que L coefficients DCT tel que $L < N$. Ainsi, $(N - L)$ coefficients vont être rejetés (dans l'étape de quantification) sans introduire une erreur notable.

Les coefficients les plus significatifs (contiennent 99% de l'énergie totale) sont peu nombreux et se trouvent essentiellement dans la partie supérieure à gauche du bloc transformé par la DCT (top left). Ainsi, c'est seulement ces coefficients qui pourraient être calculés. Cette méthode de calcul partiel est généralement mentionnée dans la littérature par "Zonale DCT" ou bien "Pruned DCT [134-135]". Mathématiquement, cela revient à calculer directement les L coefficients DCT tel que $L < N$ selon l'équation :

$$Y_{L \times 1} = A_{L \times N} \cdot X_{N \times 1} \quad (3.43)$$

Où :

$X_{N \times 1}$ est le vecteur de données, $Y_{L \times 1}$ est le vecteur transformé correspondant.

$A_{L \times N}$ est la matrice transformation réduite. Notons que cette matrice est une matrice semi orthogonale c.-à-d. : $A_{L \times N} \times A_{L \times N}^T = I_L$, avec I_L est la matrice identité d'ordre L. un exemple de la matrice DCT réduite pour $L = 4$ est présenté ci-après :

$$\begin{bmatrix} e & e & e & e & e & e & e & e \\ a & b & c & d & -d & -c & -b & -a \\ f & g & -g & -f & -f & -g & g & f \\ b & -d & -a & -c & c & a & d & -b \end{bmatrix}$$

L'équation associée à la version 2-D de la transformation réduite est la suivante [6]:

$$Y_{L \times L} = A_{L \times N} \cdot X_{N \times 1} \cdot A_{L \times N}^T \quad (3.44)$$

Le tableau 3.6 donne le nombre d'opérations nécessaires pour calculer la DCT zonale en utilisant l'algorithme de Loeffler, en 1-D et en 2-D, pour les différentes valeurs de L.

L	Coût de la DCT 1-D		Coût de la DCT 2-D	
	Mult	Add	Mult	Add
8 (référence)	11	29	176	464
7	11	28	165	420
6	10	26	140	364
5	9	25	117	325
4	9	24	108	288
3	8	23	88	253
2	6	20	60	200

Tableau 3.6 : Complexité de calcul de la DCT zonale

Comparé à l'algorithme de Loeffler original, les économies de calcul de la DCT 1-D paraissent assez faibles (d'environ 9% et 18%). Cependant, lorsqu'on projette les résultats à la DCT 2-D, la réduction du coût de calcul devient significative. En effet, cela nous fait une

économie d'environ 20 % et 38% sur chaque type d'opérations (respectivement pour $L = 6$ et $L = 4$).

L'approche zonale consiste donc à opérer sur un sous-ensemble des coefficients, c'est-à-dire les plus importants, pour calculer, quantifier, encoder et transmettre seulement ceux-là [135-136]

Le sous-ensemble peut être de deux formes : une forme carrée et une forme triangulaire (voir figure 3.8). Les auteurs dans [94] ont comparé ces deux modèles de sélection zonale et ont démontré que la forme carrée fournit une meilleure efficacité énergétique qu'une forme triangulaire. En effet, pour une valeur d'énergie constante (égale à 0, 04 mJ) le PSNR de la forme carrée est meilleur que celui de la forme triangulaire (voir tableau 3.7)

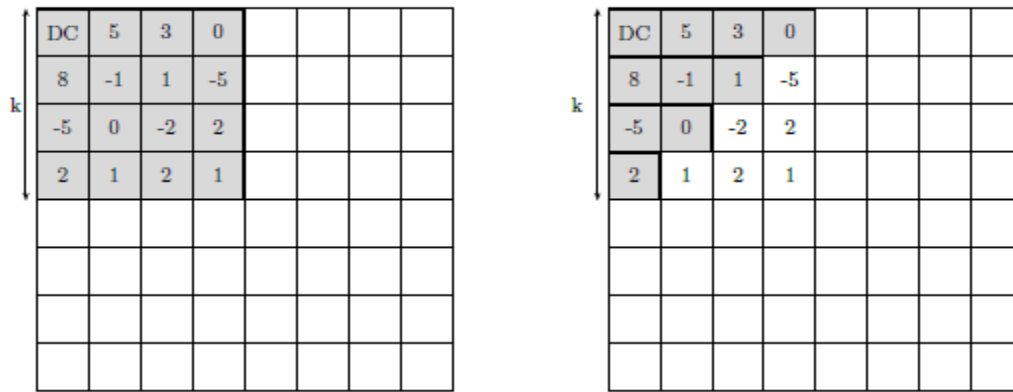


Figure 3.8 : Formes de la DCT zonale : carrée et triangulaire pour $L = 4$

Images standards	PSNR (Carrée), db	PSNR (Triangulaire), db
Lena	33.34	33.28
Goldill	30.65	30.30
Barbara	25.92	25.84

Tableau 3.7 : PSNR pour les deux formes zonale avec une consommation énergétique de 0.04 mJ

3.9 Combinaison des deux approches : Réduction des coefficients et des opérations à calculer

Les deux approches présentées, approche par factorisation de la matrice et approche par sélection zonale des coefficients, sont complémentaires. Nous avons proposé de les combiner. Le résultat d'une telle combinaison est une DCT de faible complexité. Le principe de la DCT de faible complexité proposée dans cette thèse consiste donc à sélectionner la zone des coefficients de la DCT à coder dans le bloc et les calculer en se basant sur des algorithmes de DCT 1-D rapides et disponibles dans la littérature. Ces algorithmes sont les suivants : Loeffler, Cordic Loeffler, BDCT, CB-DCT et intDCT. Cette approche réduit le nombre de coefficients à calculer, à quantifier et à coder dans chaque bloc, entraînant mécaniquement des économies d'énergie sur toute la chaîne de compression.

3.9.1 La DCT binaire (BDCT)

Les auteurs dans [153], se sont intéressés à la recherche d'une matrice binaire qui se rapproche de la matrice DCT. L'idée la plus simple consiste à appliquer une fonction 'signe' sur la matrice DCT pour obtenir une nouvelle matrice dont les entrées sont seulement 1 ou -1. La matrice résultante de la fonction signe est appelée 'SDCT' [154]. Cependant, cette matrice n'est pas orthogonale et par conséquent, la matrice inverse contient d'autres éléments autres que le 1 et le -1. Elle nécessite donc des multiplications. En outre, cette approche ne peut pas générer une DCT binaire (BDCT). Afin d'assurer l'orthogonalité, nous utilisons la matrice de Hadamard pour approcher la matrice DCT. La procédure proposée dans [153] pour la construction est de comparer une ligne donnée de la matrice SDCT avec toutes les lignes de la matrice de Hadamard et la remplacer par le plus proche. La procédure est répétée pour toutes les lignes de la matrice SDCT.

Nous illustrons ci-après cette procédure pour le cas $N = 8$. Donnons la matrice DCT (C_8) comme suit :

$$C_8 = \begin{bmatrix} e & e & e & e & e & e & e & e \\ a & b & c & d & -d & -c & -b & -a \\ f & g & -g & -f & -f & -g & g & f \\ b & -d & -a & -c & c & a & d & -b \\ e & -e & -e & e & e & -e & -e & e \\ c & -a & d & b & -b & -d & a & -c \\ g & -f & f & -g & -g & f & -f & g \\ d & -c & b & -a & a & -b & c & -d \end{bmatrix}$$

a, b, c, d, e, f et g sont des nombres positifs. La matrice SDCT est obtenue comme suit [154].:

$$\check{C}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

La matrice SDCT n'est pas orthogonale, c.-à-d. sa matrice inverse est différente de sa matrice transposée.

$$\check{C}_8^{-1} = \frac{1}{8} \begin{bmatrix} 1 & 2 & 1 & 2 & 1 & 0 & 1 & 0 \\ 1 & 2 & 1 & 0 & -1 & -2 & -1 & 0 \\ 1 & 0 & -1 & -2 & -1 & 0 & 1 & 2 \\ 1 & 0 & -1 & 0 & 1 & 2 & -1 & -2 \\ 1 & 0 & -1 & 0 & 1 & -2 & -1 & 2 \\ 1 & 0 & -1 & 2 & -1 & 0 & 1 & -2 \\ 1 & -2 & 1 & 0 & -1 & 2 & -1 & 0 \\ 1 & -2 & 1 & -2 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Dans ce qui suit, nous comparons chaque ligne de la matrice SDCT avec toutes les lignes de la matrice de Hadamard (WHT). La matrice WHT est la suivante :

$$\mathbf{H}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$$

Le résultat de cette comparaison en termes d'éléments de lignes différents de la matrice SDCT par rapport aux éléments de lignes de la matrice de Hadamard est résumé dans le tableau 3.8.

Lignes de la SDCT	Lignes de la matrice H							
	1	2	3	4	5	6	7	8
1	0	4	4	4	4	4	4	4
2	4	4	4	4	0	4	4	4
3	4	4	4	4	4	4	0	4
4	4	2	2	4	6	4	4	2
5	4	4	4	0	4	4	4	4
6	4	2	6	4	2	4	4	2
7	4	4	4	4	4	0	4	4
8	4	0	4	4	4	4	4	4

Tableau 3.8 : Comparaisons des lignes de la matrice SDCT avec ceux de la matrice de HADAMARD pour $N = 8$ en termes de nombre d'entrées qui sont différentes

Après les remplacements convenables nous arrivons à la matrice BDCT [153] présentée ci-après :

$$\hat{C}_8 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

En appliquant la procédure précédente sur un N arbitraire puissance de 2, nous obtenons la formule générale de la matrice binaire BDCT comme suit :

$$BDCT_{n,k} = (-1)^{\sum_{i=0}^{r-1} (n_i + n_{i+1})k_{r-1-i}}, 0 \leq n, k \leq N - 1 \quad (3.45)$$

Propriétés et complexité

L'algorithme rapide pour le calcul de la BDCT pour $N = 8$ est présenté dans la figure 3.9 :

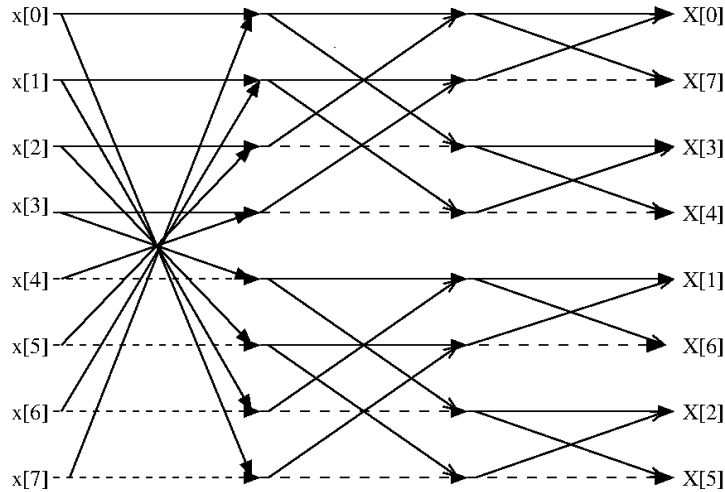


Figure 3.9 : L'algorithme rapide pour le calcul de la BDCT

D'après le diagramme de flux de la BDCT nous pouvons ainsi calculer le nombre des opérations arithmétiques nécessaires au calcul de la transformation 1D. La DCT 2-D peut être obtenue en appliquant d'abord la DCT 1-D sur chaque ligne du bloc de pixels puis sur chacune des colonnes du bloc résultant (voir section 3.7.1). Le tableau suivant résume la complexité calculatoire pour $N = 8$.

	DCT 1-D		DCT 2-D	
	Add	Décal et Mult	Add	Décal et Mult
DCT binaire (BDCT) [153]	24	0	384	0

Tableau 3.9 : Complexité calculatoire de la DCT binaire

La BDCT réduit considérablement la complexité calculatoire par rapport à la DCT réelle. Elle ne nécessite aucune opération de multiplication. Une comparaison de complexité avec la DCT réelle est présentée dans le tableau suivant.

	DCT 1-D		DCT 2-D	
	Add	Mult	Add	Mult
DCT binaire (BDCT) [153]	24	0	384	0
DCT réelle	56	64	896	1024

Tableau 3.10 : Comparaison des complexités calculatoire : BDCT vs DCT réelle

Première DCT zonale proposée :

Nous avons appliqué l’approche zonale à la BDCT [6]. Le tableau 3.11 donne le nombre d’opérations nécessaires pour calculer la DCT zonale basée sur la DCT binaire pour les différentes valeurs de L. La version 2-D est obtenue en appliquant la décomposition ligne colonne. Un exemple du diagramme de flux (pour L = 4) est montré dans la figure 3.10.

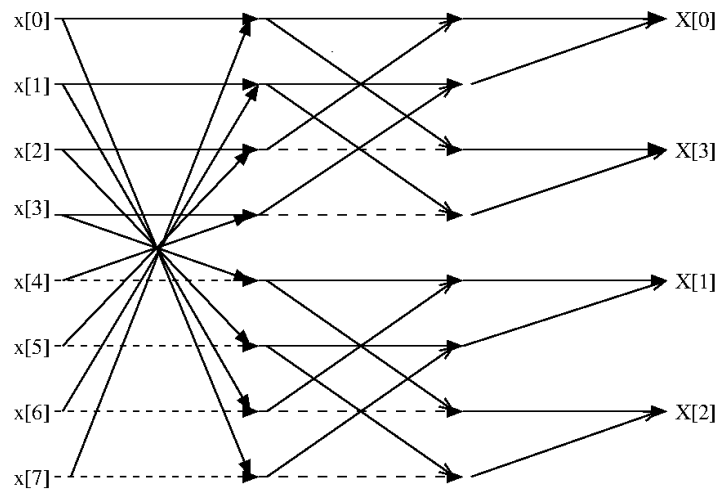


Figure 3.10 : L’algorithme rapide pour le calcul de la version zonale de la BDCT (L= 4)

	DCT 1-D		DCT 2-D zonale	
	Add	Mult+Décal	Add	Mult + Décal
(référence)	24	0	384	0
K = 7	23	0	345	0
K = 6	22	0	308	0
K = 5	21	0	273	0
K = 4	20	0	240	0
K = 3	17	0	187	0
K = 2	14	0	140	0

Tableau 3.11 : Complexité calculatoire de la version zonale de la DCT binaire

Comparé à l'algorithme BDCT original, les économies de calcul de la DCT 1-D paraissent assez faibles (d'environ 16%). Cependant, lorsqu'on projette les résultats à la DCT 2-D, la réduction du coût de calcul devient significative puisque cela nous fait une économie d'environ 37.5% (pour $L = 4$).

3.9.2 La DCT arrondie (CB-DCT)

L'approximation proposée dans [155] modifie la matrice DCT standard C par l'application d'une fonction entière. Initialement, la matrice est multipliée par un facteur de deux, ensuite une opération d'arrondie est appliquée sur chaque élément de la matrice résultante. L'opération d'arrondie est la suivante :

$$C_0 = [2 \cdot C] \tag{3.46}$$

Ainsi, la matrice résultante, est présentée ci-après:

$$C_0 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & -1 \\ 0 & -1 & 1 & 0 & 0 & 1 & -1 & 0 \\ 0 & -1 & 1 & -1 & 1 & -1 & 1 & 0 \end{bmatrix}$$

Cette matrice a les propriétés suivantes :

- Elle ne contient que des 0, des 1 et des -1, cela revient à dire que la complexité multiplicative est nulle
- Elle ne contient que des opérations d'addition
- Une approximation de la matrice DCT \hat{C} est obtenue en multipliant la matrice C_0 par deux, c.-à-d. : $\hat{C} = \left(\frac{1}{2}\right) \times C_0$.

Cependant, la matrice C_0 n'est pas orthogonale parce que $C_0^{-1} \neq C_0^T$, ceci implique deux choses :

- L'énergie peut ne pas être conservée
- La faible complexité de la transformation inverse n'est pas garantie

Pour rendre cette matrice orthogonale, nous faisons appel à la théorie des décompositions polaires des matrices [156]. Pour cela, on considère une matrice d'ajustement S donné par la formule (3.47) comme suit :

$$S = \sqrt{(C_0 \cdot C_0^T)^{-1}} \quad (3.47)$$

Le calcul de la formule (3.50) donne la matrice diagonale suivante :

$$S = \text{Diag} \left(\frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}}, \frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}} \right) \quad (3.48)$$

En termes de complexité, la matrice S ne peut pas introduire un calcul supplémentaire. Pour la compression d'images, l'opération DCT est une étape de prétraitement pour une procédure de quantification ultérieure. Par conséquent, les coefficients de la matrice diagonale peuvent être fusionnés dans l'étape de quantification. Cette procédure est suggérée et adoptée dans plusieurs travaux notamment [157-159]. En conséquence, la complexité de calcul de la transformation est limitée à C_0 .

La matrice DCT peut être mieux approximée par la matrice orthogonale suivante :

$$\hat{C}_{orth} = S \cdot C_0 \quad (3.49)$$

Propriétés et complexité

La matrice \hat{C}_{orth} possède les propriétés suivantes :

- Elle est orthogonale
- La matrice S est une matrice diagonale
- Elle possède la même complexité que la matrice C_0

L'algorithme rapide de C_0 est présenté dans la figure 3.11 :

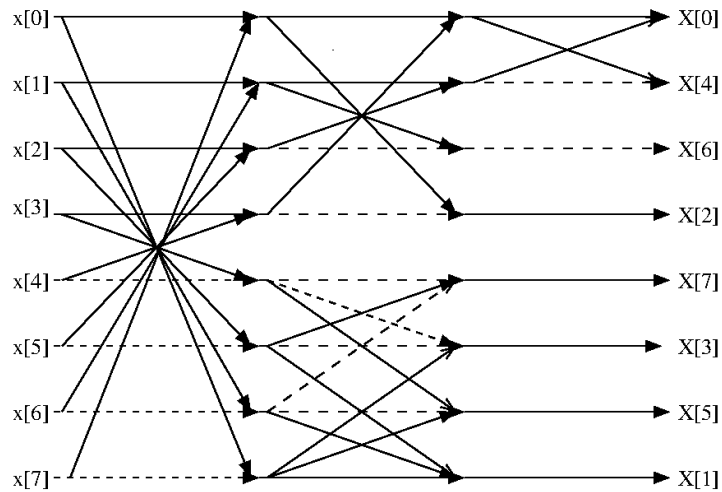


Figure 3.11 : L'algorithme rapide pour le calcul de la CB-DCT

D'après le diagramme de flux de la CB-DCT nous pouvons ainsi calculer le nombre d'opérations arithmétiques nécessaires au calcul de la transformation 1D. La version en 2D est obtenue en utilisant la décomposition ligne colonne. Le tableau suivant résume la complexité calculatoire pour $N = 8$.

	DCT 1-D		DCT 2-D	
	Add	Décal + Mult	Add	Décal+ Mult
CB-DCT [155]	22	0	352	0

Tableau 3.12 : Complexité calculatoire de la CB-DCT

La CB-DCT réduit considérablement la complexité calculatoire par rapport à la DCT réelle. Elle ne nécessite aucune opération de multiplication. Une comparaison de complexité avec la DCT réelle est présentée dans le tableau suivant.

	DCT 1-D		DCT 2-D	
	Add	Mult	Add	Mult
CB-DCT [155]	22	0	352	0
DCT réelle	56	64	896	1024

Tableau 3.13 : Comparaison des complexités calculatoire : CB-DCT vs DCT réelle

Deuxième DCT zonale proposée

Nous avons appliqué l’approche zonale à la CB-DCT. Le tableau suivant donne le nombre d’opérations nécessaires pour calculer la DCT zonale basée sur la CB-DCT pour les différentes valeurs de L. La version 2-D est obtenue en appliquant la décomposition ligne colonne. Un exemple du diagramme de flux (pour L = 4) est montré dans la figure 3.12.

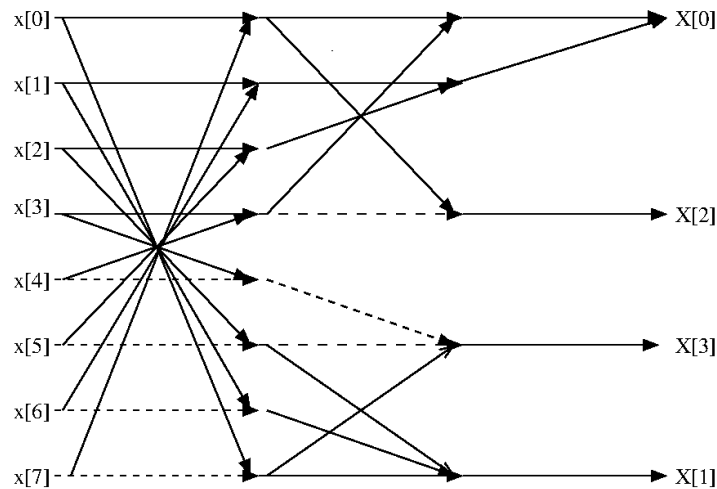


Figure 3.12 : L’algorithme rapide pour le calcul de la version zonale de la CB-DCT (L= 4)

La matrice associée à cette transformation (C_0^\wedge) (pour L = 4) est la suivante

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 & -1 & -1 \\ 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 1 & 0 & -1 & -1 & 1 & 1 & 0 & -1 \end{bmatrix}$$

Ainsi, nous obtenons une matrice semi-orthogonale à partir de C_0^\wedge comme suit :

$$C_{semi-orth}^\wedge = S_4 \cdot C_0^\wedge \tag{3.50}$$

$$\text{Où, } S_4 = \text{Diag}\left(\frac{1}{2\sqrt{2}}, \frac{1}{\sqrt{6}}, \frac{1}{2}, \frac{1}{\sqrt{6}}\right)$$

Notons que la matrice $C_{\text{semi-orth}}^{\wedge}$ est une matrice semi-orthogonale, ainsi les lignes forment un ensemble orthogonal.

La version 2-D (L = 4) est obtenue en appliquant la formule (3.44).

	DCT 1-D		DCT 2-D zonale	
	Add	Décal	Add	Décal
(référence)	22	0	352	0
K = 7	20	0	300	0
K = 6	19	0	266	0
K = 5	17	0	221	0
K = 4	16	0	192	0
K = 3	12	0	132	0
K = 2	11	0	110	0

Tableau 3.14 : Complexité calculatoire de la version zonale de la CB-DCT

Comparé à l'algorithme CB-DCT original, les économies de calcul de la DCT 1-D est d'environ 16% (pour L = 4). Cependant, lorsqu'on projette les résultats à la DCT 2-D, la réduction du coût de calcul devient significative puisque cela nous fait une économie d'environ 45.5 % (pour L = 4).

3.9.3 La DCT entière 4 × 4

La transformation directe 2D adoptée dans le standard de compression vidéo H264 [160] est la suivante :

$$Y_{2D} = (H \cdot X \cdot H^T) \otimes E_f \quad (3.51)$$

La transformation directe 1D est la suivante :

$$Y_{1D} = (H \cdot X) \otimes E \quad (3.52)$$

La matrice H est une approximation entière de la matrice DCT réelle. Cette approximation est donnée à partir de [110] comme suit :

$$H = \text{round} (\alpha H_{DCT}) \quad (3.53)$$

$(.)^T$ et \otimes sont la transposée et la multiplication point par point respectivement.

Les matrices H, H_{DCT}, E_f et E sont respectivement données par :

$$H = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}, \quad H_{DCT} = \begin{pmatrix} 0.5000 & 0.5000 & 0.5000 & 0.5000 \\ 0.6533 & 0.2706 & -0.2706 & -0.6533 \\ 0.5000 & -0.5000 & -0.5000 & 0.5000 \\ 0.2706 & -0.6533 & 0.6533 & -0.2706 \end{pmatrix}$$

$$E_f = \begin{pmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{pmatrix}, \quad E = \begin{pmatrix} a & b/2 & a & b/2 \\ b/2 & b/2 & b/2 & b/2 \\ a & b/2 & a & b/2 \\ b/2 & b/2 & b/2 & b/2 \end{pmatrix}$$

$$a = \frac{1}{2}, \quad b = \sqrt{1/2} \times \cos\left(\frac{\pi}{8}\right)$$

La matrice E_f peut-être fusionnée dans l'étape de quantification ce qui implique que cette matrice n'a aucune influence sur la complexité. La complexité calculatoire de la DCT entière (ICT) est égale à la complexité calculatoire de la transformation $(H \cdot X \cdot H^T)$.

Le diagramme de flux correspondant à la matrice noyau H est présenté dans la figure 3.13 :

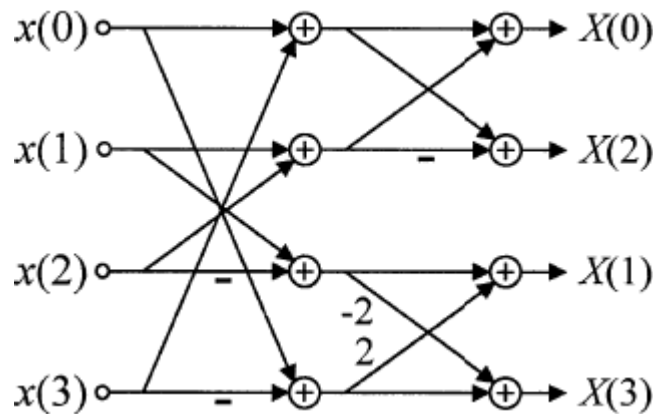


Figure 3.13 : L'algorithme rapide pour le calcul de la DCT entière (ICT)

D'après le diagramme de flux nous pouvons ainsi calculer le nombre d'opérations arithmétiques nécessaires au calcul de la DCT entière 1D et 2D. Le tableau suivant résume la complexité calculatoire pour $N = 4$.

	Addition		Décalage	
	1D	2D	1D	2D
DCT entière (ICT) [160]	8	64	2	16

Tableau 3.15 : Complexité calculatoire de la DCT entière

La DCT entière réduit considérablement la complexité calculatoire par rapport à la DCT réelle. Une comparaison entre la DCT entière et la DCT réelle (pour $N = 4$) est présentée dans le tableau suivant :

	Addition		Décalage		Multiplication	
	1D	2D	1D	2D	1D	2D
DCT entière (ICT)	8	64	2	16	0	0
DCT réelle	12	96	0	0	16	128

Tableau 3.16 : Comparaison des complexités calculatoire : DCT entière vs DCT réelle

Troisième DCT zonale proposée

Le tableau suivant donne le nombre d'opérations nécessaires pour calculer la DCT zonale basée sur la DCT entière, en 1-D et en 2-D, pour les différentes valeurs de L [161-162].

	DCT 1-D		DCT 2-D zonale	
	Add	Décal	Add	Décal
K = 4 (référence)	8	2	64	16
K = 3	7	1	56	8
K = 2	6	1	48	8

Tableau 3.17 : Complexité calculatoire de la version zonale de la DCT entière

Comparée à la DCT entière originale, les économies de calcul de la DCT 1-D paraissent assez faibles (d'environ 12.5%). Cependant, lorsqu'on projette les résultats à la DCT 2-D, la réduction du coût de calcul devient d'environ 26% (pour $L = 2$).

Conclusion :

Dans ce chapitre nous avons présenté, dans un premier temps, les différents algorithmes de calcul des coefficients de la DCT 1D ainsi qu'une comparaison de la complexité de ces algorithmes. Ensuite, après avoir étudié les algorithmes de réalisation de la DCT 2D nous avons présenté une nouvelle architecture afin de réduire le nombre d'opérateurs arithmétiques nécessaires. Les résultats obtenus ont montré que l'architecture proposée diminue considérablement la complexité et par conséquent elle est souhaitable pour une utilisation en compression d'images dans les nœuds de capteurs sans fil. Une évaluation en termes de PSNR, temps d'exécution et consommation énergétique de l'architecture proposée a été réalisée et les résultats obtenus sont détaillés dans le chapitre suivant.

Chapitre 4

Evaluation des performances

4.1 Introduction

Nous avons étudié dans les chapitres 2 et 3 des solutions de compression d'images basées sur des DCT de faibles complexités qui réduisent le nombre des opérations nécessaires aux calculs. Cette réduction va agir directement sur le temps d'exécution et donc sur la quantité d'énergie consommée par le microcontrôleur. Nous avons en effet prouvé que l'utilisation de l'algorithme CB-DCT et/ou l'algorithme BDCT (Transformations sans multiplication) combinée avec une approche zonale (Pruned) contribuent fortement à réduire le nombre des opérations de calculs nécessaires au microcontrôleur pour exécuter l'algorithme de compression de l'image.

Ce chapitre a pour objectif d'évaluer les performances de nos propositions, en particulier le temps d'exécution, la consommation énergétique et aussi et surtout la qualité des images reconstruites.

Les performances des DCT rapides proposées ont été évaluées en considérant une chaîne de compression JPEG, et en utilisant la table de quantification standard et du codage de Huffman qui sont recommandées dans l'annexe le standard JPEG [163]. La consommation d'énergie et le temps d'exécution ont été calculées en adoptant le nœud capteur 'Mica2' comme référence. Ce dernier possède un microcontrôleur 'Atmega128L' qui est l'un des plus cités dans la littérature scientifique portant sur cette thématique.

Dans ce chapitre, nous étudierons et nous analyserons les résultats obtenus. Enfin, Pour prouver l'efficacité de cette approche, une comparaison des résultats obtenus avec ceux récemment rapportés dans [7] sera également effectuée.

4.2 Nombre de cycles

Nous avons basé la simulation en considérant le microcontrôleur de la famille ATMEL 'ATmega128L'. Ce microcontrôleur est le plus adopté dans la littérature [7][164-165]. Nous avons appliqué les différents algorithmes de la DCT sur un bloc de 8×8 pixels et nous avons obtenu le nombre de cycles que doit exécuter ce microcontrôleur. Le nombre de cycle a été déterminé en utilisant l'outil logiciel Atmel studio, il s'agit d'un outil de développement de la série 'Atmel AVR' des microprocesseurs RISC.

Les résultats pour chaque méthode de DCT et pour chaque valeur de L sont donnés dans le tableau 4.1.

	L = 8	L = 7	L = 6	L = 5	L = 4	L = 3	L = 2
Loeffler	732873	676494	588269	503150	485468	410264	339892
BDCT	1850	1666	1450	1274	1113	890	700
CB-DCT	1760	1560	1303	1105	1036	673	583
intDCT	-	-	-	-	1600	1280	1142

Tableau 4.1 : Nombre de cycles des différents algorithmes pour le calcul de la DCT appliqués sur un bloc 8×8 obtenu par le simulateur Atmel Studio avec le microcontrôleur ATmega128L.

D'après les résultats présentés dans le tableau 4.1, nous pouvons constater que l'algorithme CB-DCT est meilleur en termes de nombres de cycles. L'algorithme BDCT donne aussi des résultats compétitifs en comparant avec CB-DCT.

Comme l'algorithme de Loeffler est basé sur des nombres réels (il calcule la DCT exacte), les résultats en termes de nombres de cycles sont nettement moins bons par rapport aux résultats obtenus par CBDCT, BDCT et intDCT, qui sont basés sur des nombres entiers.

4.3 Consommation énergétique et temps d'exécution

Le coût d'énergie est proportionnel au nombre de cycles exécutés par le microcontrôleur. En effet, en utilisant le nombre de cycle obtenu précédemment, nous pouvons calculer la consommation énergétique et le temps d'exécution du microcontrôleur des différentes DCT 2-D sur un bloc de 8×8 pixels.

Au niveau du microcontrôleur, le temps d'exécution et la consommation énergétique sont fonction du nombre de cycles (C), de la puissance (P) et de la fréquence de son horloge (F).

Le temps d'exécution du microcontrôleur (T_{MCU}), est donné par la formule (4.1) :

$$T_{MCU} = \frac{C}{F} \quad (4.1)$$

La consommation énergétique (E_{MCU}) est donnée par la formule (4.2) :

$$E_{MCU} = T_{MCU} \times P \quad (4.2)$$

Les valeurs numériques de la puissance (P) et de la fréquence horloge (F) caractérisent le nœud capteur "Mica2" basé sur un microcontrôleur ATmega128L. Elles sont tirées de [166] et elles sont validées par plusieurs travaux dans la littérature [7] [164-165]. Ainsi, ces valeurs sont : $P = 23$ mW et $F = 7,3728$ MHz.

Les résultats en termes de consommation énergétique et temps d'exécution pour les algorithmes de calcul de la DCT : Loeffler, BDCT, CBDCT et intDCT pour chaque valeur de L sont donnés dans les tableaux: 4.2, 4.3, 4.4 et 4.5 respectivement.

	Temps d'exécution (μ s)	Consommation énergétique (μ J)
L = 8 (réf)	99402	2286
L = 7	91755	2110
L = 6	79789	1835
L = 5	68244	1569
L = 4	65845	1514
L = 3	55645	1279
L = 2	46100	1060

(a)

	Temps d'exécution (μ s)	Consommation énergétique (μ J)
L = 8 (réf)	250	5.77
L = 7	225	5.18
L = 6	196	4.52
L = 5	172	3.97
L = 4	150	3.47
L = 3	120	2.78
L = 2	94	2.18

(b)

	Temps d'exécution (μ s)	Consommation énergétique (μ J)
L = 8 (réf)	238	5.47
L = 7	211	4.85
L = 6	176	4.05
L = 5	150	3.45
L = 4	140	3.22
L = 3	91	2.09
L = 2	79	1.81

(c)

Tableau 4.2: Consommation énergétique et temps d'exécution pour le calcul des algorithmes sur un bloc 8×8 obtenu par le simulateur Atmel Studio avec le microcontrôleur ATmega128L. (a)-Loeffler zonale, (b)BDCT zonale, (c)-CB-DCT zonale

	Temps d'exécution (μ s)	Consommation énergétique (μ J)
L = 4 (réf)	217	4.99
L = 3	174	4.00
L = 2	155	3.56

Tableau 4.3: Consommation énergétique et temps d'exécution pour le calcul de l'algorithme *intDCT* zonale sur un bloc 8×8 obtenu par le simulateur Atmel Studio avec le microcontrôleur ATmega128L.

Lecuire et Makkaoui dans [7] [94], ont évalué plusieurs algorithmes de calculs de la DCT. Ils ont prouvé que l'utilisation de la DCT zonale basée sur l'algorithme de Cordic-Loeffler (ZCL) [7] fournit le meilleur compromis entre complexité de calcul et distorsion d'images. Autrement dit, la réduction du temps d'exécution et de la consommation énergétique est au dépit d'une réduction acceptable de la qualité des images reconstruites. Les résultats en termes de consommation énergétique et temps d'exécution pour l'algorithme ZCL [7] pour chaque valeur de L sont donnés dans le tableau 4.4.

	Temps d'exécution (μ s)	Consommation énergétique (μ J)
L = 8 (réf)	630	14.49
L = 7	522	12.00
L = 6	458	10.53
L = 5	413	9.49
L = 4	371	8.53
L = 3	341	7.85
L = 2	261	6.00

Tableau 4.4: Consommation énergétique et temps d'exécution pour le calcul de l'algorithme ZCL sur un bloc 8×8 avec le microcontrôleur ATmega128L.

Lorsque les 64 coefficients de la DCT sont calculés (L = 8), il faut : 99402, 250, 238 et 630 μ s au microcontrôleur pour transformer un bloc de 8×8 respectivement pour les algorithmes de : Loeffler, BDCT, CBDCT et ZCL. La dépense énergétique est ainsi : 2286, 5.77, 5.47, et 14,49 μ J respectivement pour ces mêmes algorithmes. Autrement dit, les algorithmes BDCT et CBDCT réduisent la consommation énergétique par environ 60 % et 62% respectivement par rapport à l'algorithme CL [101].

Comme prévu, en raison de la réduction considérable du nombre d'opérations arithmétiques (voir chapitre 3), les algorithmes proposés dans cette thèse ont des performances bien supérieures en termes de temps d'exécution et de consommation énergétique par rapport à Z-

CL [7]. En moyenne, environ 60% et 64% (Voir figure 4.1) des économies en temps de calcul et en énergie ont été obtenus en utilisant l’algorithme BDCT zonale et CB-DCT zonale respectivement. Cette réduction a été obtenue au prix d’une baisse acceptable de la qualité des images reconstruites comme nous allons voir dans le reste du chapitre.

En ce qui concerne l’algorithme intDCT zonale, il est plus performant par rapport à ZCL. Des économies en termes d’énergie d’environ 44 % ont été obtenues respectivement pour $L = 4, 3$ et 2 . Etant donné que cette dernière opère sur des tailles de 4×4 , l’effet de bloc devient significatif et par conséquent une baisse d’environ 3 dB dans les bitrates supérieurs à 0,5 bpp a été notée [161-162]. Dans les bitrates inférieurs à 0,5 la baisse en termes de PSNR devient plus significative. Dans le reste du chapitre nous n’allons pas prendre en considération l’algorithme intDCT.

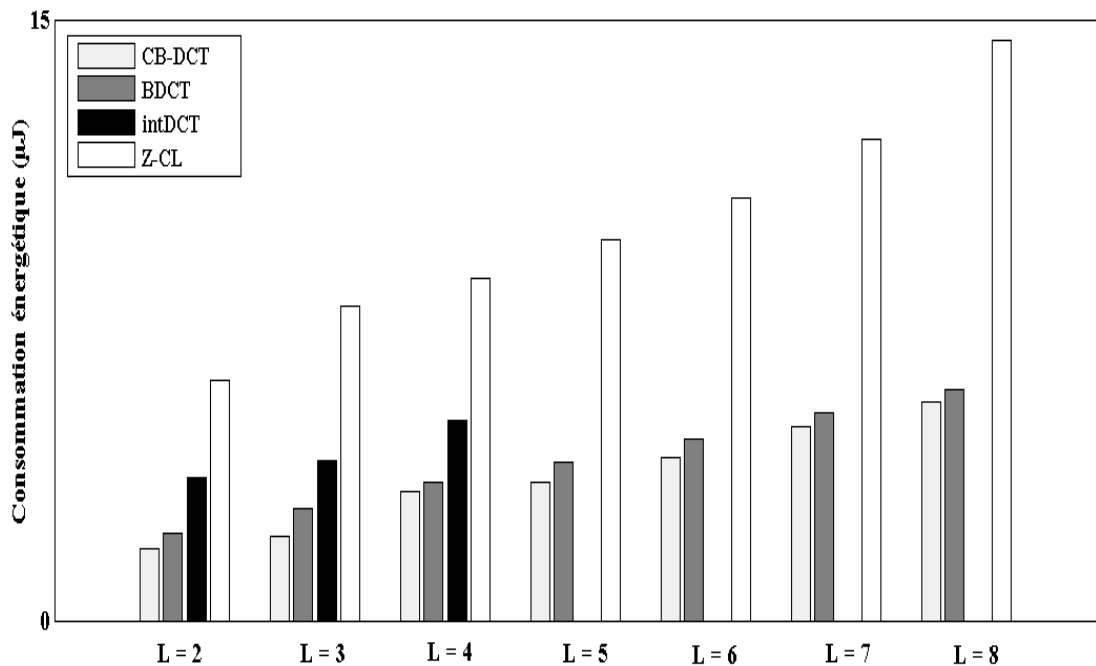


Figure 4.1: Consommation énergétique des différents algorithmes appliqués sur un bloc 8×8 obtenu par le simulateur Atmel Studio avec le microcontrôleur ATmega128L.

4.4 Qualité des images reconstruites

4.4.1 Métriques d’évaluation

Dans la littérature, il existe deux grandes catégories de mesure de qualité. Celles qui sont à référence totale (utilisation complète de l’image originale), et celles qui n’utilisent aucune référence. Les méthodes ”sans référence” sont basées sur la connaissance a priori de la nature

des dégradations ou des images. Les mesures sans référence sont les plus attrayantes car elles permettent d'estimer la qualité uniquement à partir de l'image dégradée. Cependant, le type de dégradation contenu dans l'image est généralement supposé connu, ce qui limite fortement leur utilisation dans des situations pratiques. Les méthodes "sans référence" ne feront pas l'objet de cette thèse. Dans la seconde catégorie, ces types de mesures permettent de déterminer la différence entre une image originale et sa version dégradée. Généralement, aucune information liée au type de dégradation n'est exploitée. Dans ce travail nous allons nous intéresser plus particulièrement à cette seconde catégorie.

Il existe une grande variété de métriques de qualité d'images numériques avec référence, plus de 111 métriques [167].

Les méthodes les plus directes pour mesurer la distorsion sont les méthodes purement mathématiques basées sur une différence mathématique $D(m, n)$ pixel à pixel entre l'image originale $I_0(m, n)$ et l'image dont la qualité est à évaluer $I_d(m, n)$. L'avantage de ces mesures mathématiques réside dans la simplicité d'implantation, ainsi que dans la rapidité de calcul.

$$D(m, n) = |I_0(m, n) - I_d(m, n)|^p \quad (4.3)$$

Lorsque $p = 2$, cette égalité permet de calculer l'erreur quadratique moyenne (MSE, Mean square error), qui est définie par l'équation :

$$MSE = \frac{1}{M \cdot N} \sum_{m=1}^M \sum_{n=1}^N |I_0(m, n) - I_d(m, n)|^2 \quad (4.4)$$

La métrique de type signal la plus connue et la plus utilisée en image est le Peak Signal to Noise Ratio (PSNR) qui est corrélé à la MSE. Le PSNR est défini par l'équation :

$$PSNR = 10 \cdot \text{Log}_{10} \left(\frac{d^2}{MSE} \right) \quad (4.5)$$

Où d est l'amplitude maximale du signal. En général, dans le cas d'une image où les composantes d'un pixel sont codées sur 8 bits, $d = 255$.

Taux de compression et débit binaire

Le *taux de compression* (TC) est défini comme le rapport de nombre des bits nécessaires pour représenter l'image originale au nombre de bits dans l'image compressée.

Considérons une image en niveaux de gris 512×512 pixels, si chaque pixel est représenté sur 8 bits, elle nécessite 262144 octets pour le stockage. Si après la compression, elle nécessite seulement 16384 octets, alors le taux de compression est de 16 :1.

$$TC = \frac{\text{nombre de bits dans l'image originale}}{\text{nombre de bits retenus}} \quad (4.6)$$

Une autre variation du rapport de compression qui est très utilisée dans la littérature est le débit (ou bien Bitrate). Il est défini comme le nombre moyen des bits pour représenter pixel de l'image. Alors, pour l'exemple précédent, nous pouvons dire qu'un TC = 16 :1 correspond à un débit de 0.5 bits per pixel (*bpp*). Nous pouvons le mesurer par la formule suivante :

$$\text{Débit (bpp)} = \frac{\text{nombre de bits d'un pixel de l'image originale}}{TC} \quad (4.7)$$

4.4.2 Résultats de compression par les DCT de faibles complexités

Dans cette partie, nous présentons les résultats expérimentaux obtenus par l'implémentation des algorithmes : Loeffler, BDCT et CBDCT. Nous avons appliqué la chaîne de compression < DCT, Quantification scalaire, Codage de Huffman > sur un ensemble d'images de tests (voir figure 4.2). Les images de tests ont une taille de 512×512 pixels sauf pour l'image 'Cameraman' elle est de 256×256 pixels. Ces images ont été sélectionnées parmi celles communément utilisées par les concepteurs de méthodes de compression d'images. Ils sont disponibles sur [168]

En adoptant le PSNR comme métrique, en variant le paramètre L et en utilisant plusieurs taux de compression, allant de 0,1 à 0,5 bpp, nous avons obtenu les résultats présentés sur les tableaux : 4.5, 4.6 et 4.7, respectivement pour les algorithmes de Loeffler, BDCT et CBDCT.



Figure 4.2: Image originales de tests (8 bpp)

	PSNR (dB)						
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	18,89	22,13	18,77	20,14	18,34	20,92	19,87
0,2	23,41	30,28	20,86	25,96	23,76	28,25	25,42
0,3	24,17	31,96	21,77	27,27	24,48	29,69	26,56
0,4	24,39	32,42	22,15	27,7	24,63	30,03	26,89
0,5	24,47	32,6	22,32	27,84	24,68	30,17	27,01

(a)

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	18,89	22,57	18,8	22,15	18,37	21,22	20,33
0,2	23,46	30,44	20,9	26,27	24,44	28,63	25,71
0,3	24,51	33,01	22,03	28,16	25,76	30,83	27,38
0,4	24,99	34,14	22,73	29,1	26,29	31,61	28,14
0,5	25,28	34,78	23,22	29,63	26,52	31,97	28,57

(b)

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	19,45	22,81	19,01	20,75	18,44	21,49	20,33
0,2	23,69	30,49	20,98	26,38	24,57	28,89	25,83
0,3	24,97	33,2	22,19	28,45	26,42	31,36	27,77
0,4	25,76	34,77	23,03	29,75	27,36	32,48	28,86
0,5	26,39	35,7	23,67	30,59	27,94	33,09	29,56

(c)

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	19,45	23,05	19,01	20,75	18,44	21,49	20,37
0,2	23,73	30,51	20,98	26,38	24,59	28,93	25,85
0,3	25,17	33,23	22,21	28,52	26,54	31,65	27,89
0,4	26,41	34,9	23,07	29,95	27,78	32,91	29,17
0,5	27,39	35,97	23,79	30,9	28,62	33,71	30,06

(d)

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	19,53	23,29	19,01	20,75	18,44	21,49	20,42
0,2	23,74	30,53	20,98	26,38	24,57	28,98	25,86
0,3	25,17	33,23	22,21	28,52	26,62	31,65	27,9
0,4	26,64	34,93	23,07	29,93	27,99	33,07	29,27
0,5	28	36,01	23,81	31,02	28,92	33,96	30,29

(e)

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	20,27	24,11	19,01	20,15	19,81	21,49	20,81
0,2	23,74	30,81	20,98	26,48	24,65	28,98	25,97
0,3	25,44	33,36	22,21	28,52	26,44	31,65	27,94
0,4	26,79	34,84	23,12	29,89	28,05	33,07	29,29
0,5	28,10	35,89	23,81	30,92	28,98	33,96	30,30

(f)

Tableau 4.5: Résultats de décompression des différentes images de tests par la transformée de Loeffler zonale : (a),(b),(c),(d),(e) et (f) respectivement pour $L = 3,4,5,6,7$ et 8

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	18,5	19,4	17,37	19,2	17,2	18,6	18,38
0,2	22,94	28,55	20,73	25,23	22,9	26,51	24,48
0,3	23,57	29,49	21,41	26	23,32	27,19	25,16
0,4	23,7	29,74	21,73	26,26	23,41	27,36	25,37
0,5	23,76	29,81	21,84	26,37	23,44	27,42	25,44

(a)

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	18,6	20,41	17,9	19,44	16,73	18,56	18,61
0,2	23,23	28,95	20,77	25,45	23,57	27,18	24,84
0,3	24,21	30,99	21,67	26,97	24,64	28,64	26,19
0,4	24,75	31,76	22,21	27,64	25,09	29,17	26,77
0,5	25,02	32,12	22,61	28,02	25,27	29,44	27,08

(b)

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	18,4	20,36	18	19,44	16,9	18,6	18,62
0,2	23,3	28,98	20,75	25,54	23,71	27,16	24,91
0,3	24,55	31,12	21,75	27,13	25,24	28,92	26,45
0,4	25,4	32,03	22,48	28,07	25,91	29,65	27,26
0,5	25,86	32,57	23	28,6	26,32	30,01	27,73

(c)

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	18,4	20,4	17,9	19,5	16,8	18,6	18,60
0,2	23,4	28,98	20,67	25,55	23,71	27,27	24,93
0,3	24,73	31,12	21,75	27,23	25,4	29,2	26,57
0,4	25,7	32,21	22,58	28,26	26,43	30,11	27,55
0,5	26,4	32,92	23,15	29,01	27,05	30,61	28,27

(d)

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	18,4	20,36	17,90	19,5	16,73	18,56	18,58
0,2	23,4	28,95	20,67	25,55	23,71	27,27	24,93
0,3	24,82	31,09	21,7	27,2	25,45	29,3	26,59
0,4	25,78	32,34	22,61	28,43	26,59	30,52	27,71
0,5	26,7	33,15	23,25	29,29	27,44	31,21	28,51

(e)

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	18,4	20,37	17,90	19,6	18,73	18,57	18,93
0,2	23,3	28,98	20,67	25,6	23,8	27,2	24,93
0,3	24,82	31,09	21,71	27,21	25,39	29,49	26,62
0,4	25,79	32,38	22,62	28,45	26,6	30,83	27,78
0,5	26,71	33,25	23,24	29,35	27,65	31,82	28,67

(f)

Tableau 4.6: Résultats de décompression des différentes images de tests par la transformée BDCT zonale : (a),(b),(c),(d),(e) et (f) respectivement pour $L = 3,4,5,6,7$ et 8

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	18,71	20,62	18,21	19,42	18,18	19,44	19,10
0,2	23,82	29,27	20,86	25,6	23,14	27,37	25,01
0,3	23,79	30,6	21,56	26,53	23,65	28,45	25,76
0,4	23,98	30,9	21,88	26,86	23,76	28,67	26,01
0,5	24,04	31,02	22,02	26,97	23,79	28,75	26,10

(a)

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	18,71	20,62	18,22	19,42	18,19	19,44	19,10
0,2	23,27	29,41	20,81	25,85	23,72	27,72	25,15
0,3	24,14	31,29	21,83	27,23	24,7	29,27	26,41
0,4	24,55	31,98	22,43	27,89	25,06	29,80	26,95
0,5	24,8	32,32	22,80	28,24	25,23	30,03	27,24

(b)

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	18,71	20,62	18,22	19,42	18,19	19,44	19,10
0,2	23,48	29,44	20,89	25,83	23,87	27,83	25,22
0,3	24,55	31,43	21,95	27,48	25,16	29,73	26,72
0,4	25,25	32,30	22,70	28,37	25,82	30,36	27,47
0,5	25,75	32,82	23,22	28,90	26,20	30,70	27,93

(c)

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	18,75	20,73	18,22	19,46	18,27	19,54	19,16
0,2	23,42	29,45	20,93	25,83	23,91	28,02	25,28
0,3	24,79	31,51	22,01	27,53	25,35	30,13	26,89
0,4	25,71	32,55	22,79	28,48	26,17	31,07	27,80
0,5	26,48	33,17	23,39	29,19	26,66	31,60	28,42

(d)

PSNR (dB), L=7							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	18,75	20,73	18,22	19,46	18,27	19,54	19,16
0,2	23,52	29,45	20,93	25,84	23,91	28,03	25,28
0,3	24,89	31,56	22,02	27,61	25,59	30,25	26,99
0,4	25,94	32,6	22,81	28,69	26,71	31,27	28,00
0,5	26,85	33,5	23,44	29,55	27,51	31,97	28,80

(e)

PSNR (dB), L=8							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	18,75	20,73	18,22	19,46	18,27	19,54	19,16
0,2	23,52	29,45	20,93	25,84	23,91	28,03	25,28
0,3	24,89	31,56	22,02	27,62	25,61	30,27	27,00
0,4	25,95	32,67	22,81	28,73	26,79	31,42	28,06
0,5	26,9	33,69	23,44	29,61	27,85	32,27	28,96

(f)

Tableau 4.7: Résultats de décompression des différentes images de tests par la transformée CBDCT zonale : (a),(b),(c),(d),(e) et (f) respectivement pour $L = 3,4,5,6,7$ et 8

4.4.3. Discussion

D'après les résultats, présentés dans les tableaux ci-dessus, nous pouvons affirmer que:

Pour des grandes valeurs de L (entre 6 et 8), et pour le même débit, nous constaterons visuellement (voir figure 4.3, 4.4 et 4.5) et dans les valeurs du PSNR, que les qualités des images reconstruites pour les trois DCT zonale proposées (Loeffler, BDCT et CB-DCT) est très proches de celles obtenue avec la DCT de référence (c.-à-d. pour $L = 8$).

Ainsi, pour un débit de 0,5 bpp, la dégradation moyenne ne dépasse pas les 0,25 dB pour l'algorithme Loeffler zonale (Tableau 4.5):

- Pour $L = 6$, la dégradation moyenne en termes de PSNR est de 0,24 dB par rapport à la DCT référence
- Pour $L = 7$, la dégradation en termes de PSNR est de 0,02 dB par rapport à la DCT de référence

Pour un débit de 0,5 bpp, la dégradation moyenne ne dépasse pas les 0,5 dB pour l'algorithme BDCT zonale (Tableau 4.6), ainsi:

- Pour $L = 6$, la dégradation moyenne en termes de PSNR est de 0,40 dB par rapport à la DCT référence
- Pour $L = 7$, la dégradation en termes de PSNR est de 0,16 dB par rapport à la DCT de référence

Pour un débit de 0,5 bpp, la dégradation moyenne est au voisinage de 0,5 dB pour l'algorithme CB-DCT zonale (Tableau 4.7), ainsi:

- Pour $L = 6$, la dégradation moyenne en termes de PSNR est de 0,55 dB par rapport à la DCT référence
- Pour $L = 7$, la dégradation en termes de PSNR est de 0,16 dB par rapport à la DCT de référence

Il y a donc matière à faire des économies d'énergie en utilisant une telle DCT avec un très faible impact sur la distorsion d'image.

Pour des petites valeurs de L (5 et en dessous), et pour le même débit, la distorsion s'accroît rapidement lorsque L diminue. Ainsi, pour un débit de 0,5 bpp, les dégradations moyennes par rapport à l'algorithme de référence pour l'algorithme de Loeffler zonale sont les suivantes (Tableau 4.5):

- Pour $L = 5$, la dégradation moyenne en termes de PSNR est de 0,74 dB
- Pour $L = 4$, la dégradation moyenne en termes de PSNR est de 1,74 dB

Pour l'algorithme BDCT zonale (Tableau 4.6):

- Pour $L = 5$, la dégradation moyenne en termes de PSNR est de 0,94 dB
- Pour $L = 4$, la dégradation moyenne en termes de PSNR est de 1,59 dB

Et pour l'algorithme CB-DCT zonale (Tableau 4.7):

- Pour $L = 5$, la dégradation moyenne en termes de PSNR est de 1,03 dB
- Pour $L = 4$, la dégradation moyenne en termes de PSNR est de 1,72 dB

En fait, la qualité d'image reste visuellement acceptable pour beaucoup d'applications jusqu'à $L = 4$ (voir par exemple les figure 4.3, 4.4 et 4.5). En dessous de ces valeurs (c.-à-d. pour $L = 3$ et 2), les effets de blocs classiques de la compression par DCT à bas débit deviennent trop marqués.



L = 8
PSNR = 28,10 dB Bitrate = 0,5 bpp



L = 6
PSNR = 27,39 dB Bitrate = 0,5 bpp



L = 4
PSNR = 25,28 dB Bitrate = 0,5 bpp



L = 2
PSNR = 23,42 dB Bitrate = 0,5 bpp

Figure 4.3: image Barbara codée à 0.5 bpp par la transformée de Loeffler zonale, pour différentes valeurs de L



L = 8
PSNR = 26,71 dB Bitrate = 0,5 bpp



L = 6
PSNR = 26,42 dB Bitrate = 0,5 bpp



L = 4
PSNR = 25,02 dB Bitrate = 0,5 bpp



L = 2
PSNR = 22,91 dB Bitrate = 0,5 bpp

Figure 4.4: image Barbara codée à 0.5 bpp par la transformée BDCT zonale, pour différentes valeurs de L



Figure 4.5: image Barbara codée à 0.5 bpp par la transformée CB-DCT zonale, pour différentes valeurs de L

Nous pouvons aussi constater à partir des résultats des tableaux 4.5, 4.6 et 4.7 que, à très bas débit (bitrate < 0.2 *bpp*) quelle que soit la valeur de L , la distorsion par rapport aux algorithmes de références diminue. Les valeurs du PSNR ainsi que la qualité visuelle des images reconstruites (voir Figures 4.6, 4.7 et 4.8) deviennent de plus en plus proches de celles obtenues avec les DCT de références (c.-à-d. pour $L = 8$).

Les dégradations moyennes en termes de PSNR pour un débit de 0,2 *bpp* sont :

- Pour l'algorithme de Loeffler zonale : 0.11, 0.12, 0.14, 0.27 et 0.55 dB
- Pour l'algorithme de BDCT zonale : 0.00, 0.01, 0.02, 0.09 et 0.45 dB
- Pour l'algorithme de CB-DCT zonale : 0, 0, 0.06, 0.14 et 0.27 dB.

respectivement pour $L = 7, 6, 5, 4$ et 3 .



$L = 8$
PSNR = 23,74 dB Bitrate = 0,2 bpp



$L = 6$
PSNR = 23,57 dB Bitrate = 0,2 bpp



$L = 4$
PSNR = 23,46 dB Bitrate = 0,2 bpp



$L = 2$
PSNR = 23,21 dB Bitrate = 0,2 bpp

Figure 4.6: image Barbara codée à 0.2 bpp par la transformée de Loeffler zonale, pour différentes valeurs de L



L = 8
PSNR = 23,30 dB Bitrate = 0,2 bpp



L = 6
PSNR = 23,30 dB Bitrate = 0,2 bpp



L = 4
PSNR = 23,23 dB Bitrate = 0,2 bpp



L = 2
PSNR = 22,73 dB Bitrate = 0,2 bpp

Figure 4.7: image Barbara codée à 0.2 bpp par la transformée BDCT zonale, pour différentes valeurs de L

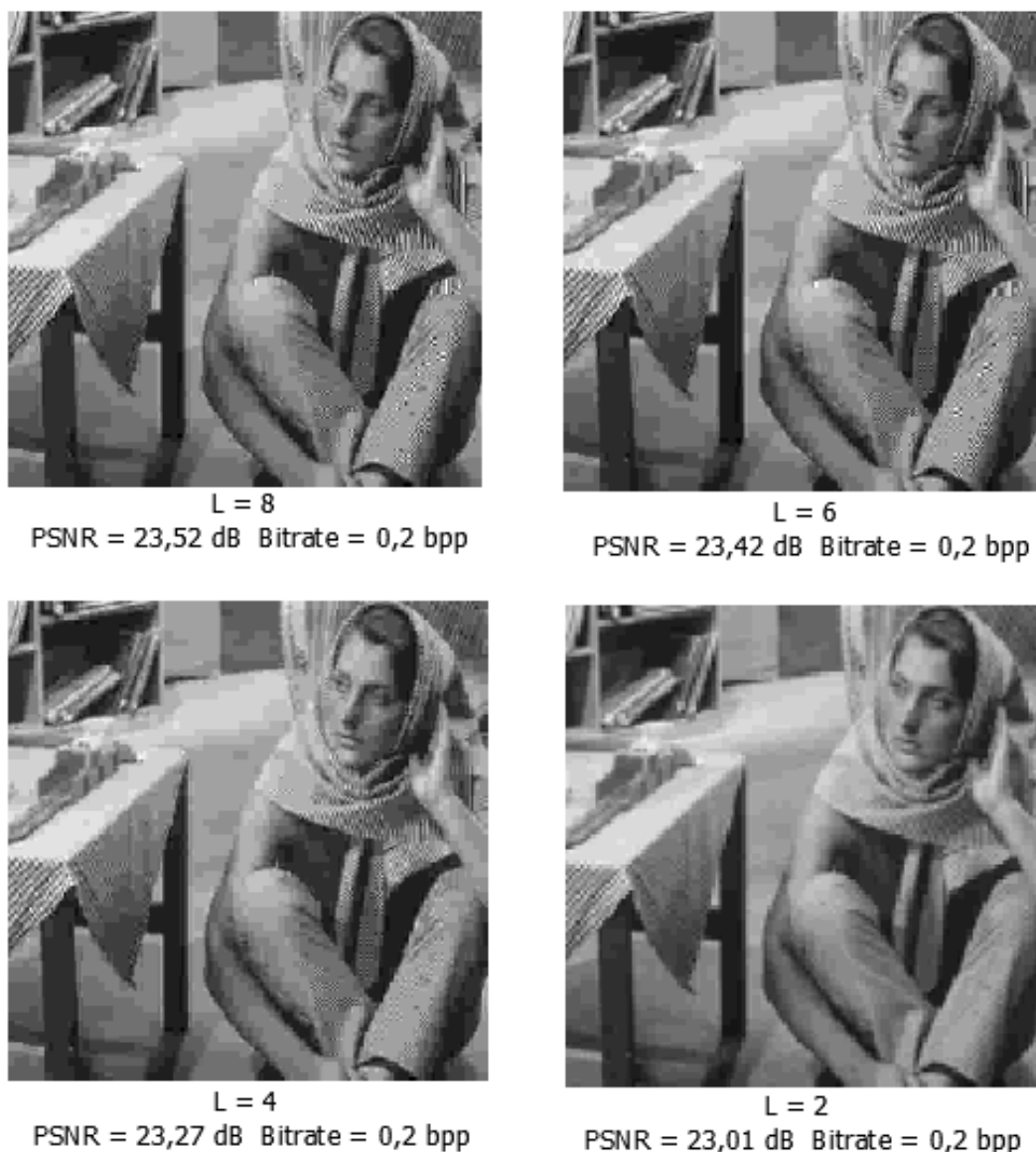


Figure 4.8: image Barbara codée à 0.2 bpp par la transformée CB-DCT zonale, pour différentes valeurs de L

Les résultats montrés dans les Tableaux 7.5, 7.6 et 7.7 sont représentés sous forme de courbes 'débit –distorsion' sur les figures 4.9, 4.10 et 4.11 respectivement pour les algorithmes de Loeffler zonale, BDCT zonale et CB-DCT zonale. Ces courbes donnent la variation de la distorsion obtenue (PSNR) après reconstruction de l'image en fonction du débit binaire (bitrate) et aussi pour différentes dimensions de la zone (L). Les courbes ont été tracées en prenant la moyenne du PSNR de toutes les images des tests.

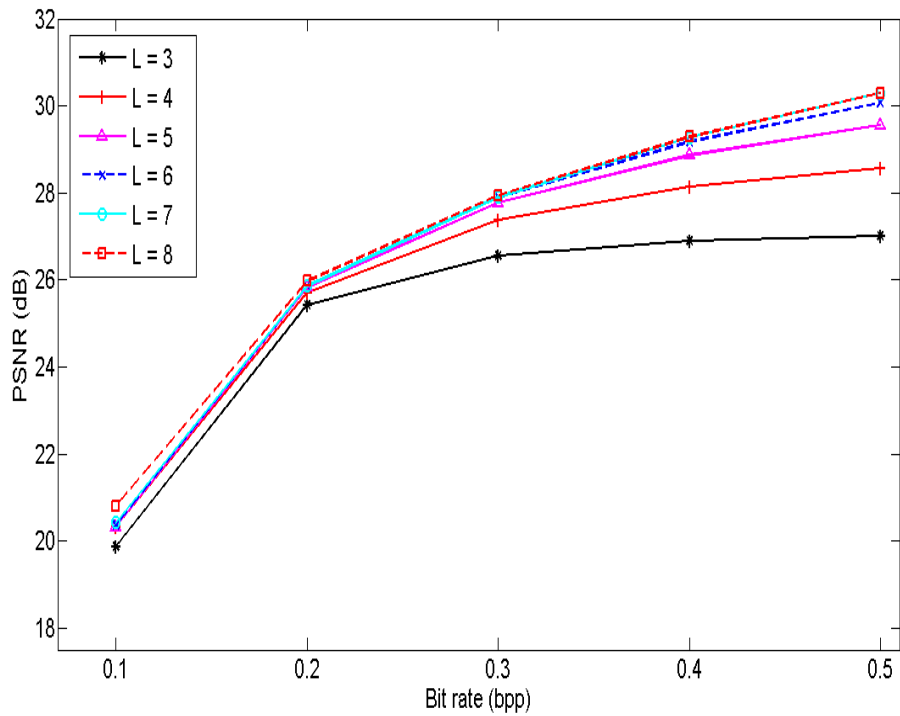


Figure 4.9: Courbes débit – distorsion moyenne de l’algorithme Loeffler zonale

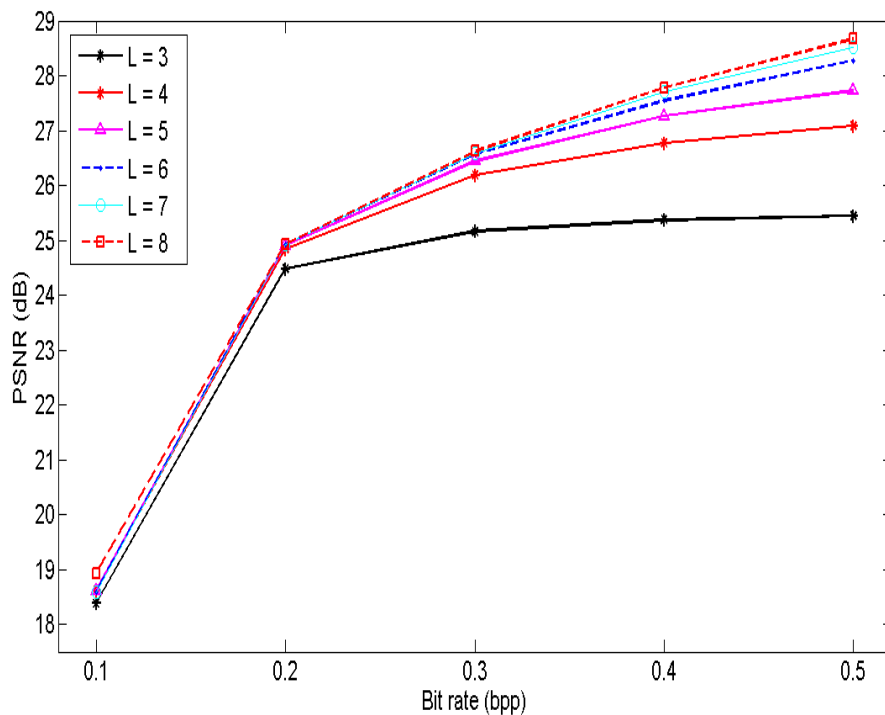


Figure 4.10: Courbes débit – distorsion moyenne de l’algorithme BDCT zonale

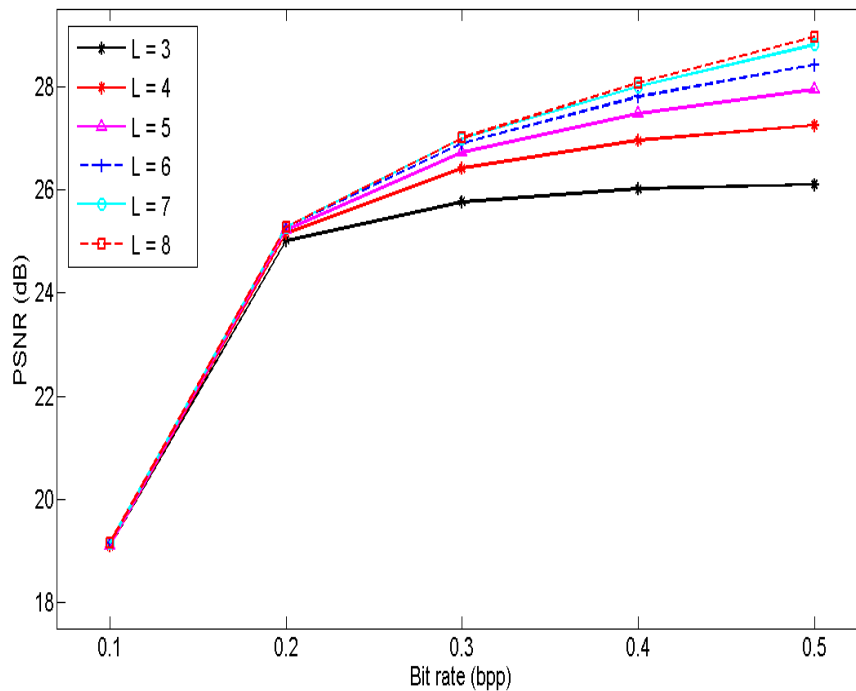


Figure 4.11: Courbes débit – distorsion moyenne de l’algorithme CB-DCT zonale

Les courbes représentées ci-dessus montrent une appréciation globale des PSNR moyennes vis-à-vis des taux de compression. À partir de ces courbes nous pouvons affirmer que :

Pour les débits supérieurs à 0,2 bpp, la distorsion par rapport aux algorithmes de références augmente (et bien évidemment le PSNR diminue) quand L diminue. En fait, la dégradation est très acceptable pour des valeurs de L qui sont supérieures à 5 (L = 6 et 7) et les économies énergétiques par exemple pour L est égale à 6 sont déjà significatives : 20% , 21,5 % et 26% respectivement pour l’algorithmes de Loeffler, BDCT et CBDCT.

Pour les valeurs de L qui sont supérieures à 3 (L = 4 et 5) la dégradation augmente un peu, mais elle reste acceptable. Les applications qui tolèrent une telle distorsion peuvent utiliser ces valeurs de L. les économies d’énergie s’avèrent alors importantes : pour L=4, elles représentent 34%, 40% et 41% respectivement pour l’algorithme de Loeffler, BDCT et CBDCT.

Enfin, pour les valeurs de L inférieures à 4 (L = 2 et 3) la dégradation devient plus importante. Mais aussi les économies d’énergie deviennent très intéressantes : par exemple pour L = 2,

53%, 62% et 67% respectivement pour l'algorithme de Loeffler, BDCT et CBDCT. De telles économies d'énergie vont permettre aux nœuds capteurs de prolonger considérablement la durée de vie.

A très bas débits (c.-à-d. inférieurs à 0.2 bpp), la dégradation n'est pas notable. De même la qualité visuelle reste pratiquement la même. Ce constat est valide quelle que soit la valeur de L. Il sera intéressant donc pour les applications qui travaillent à bas débits, d'utiliser les valeurs de L qui donnent les meilleures performances énergétiques. Autrement dit, opter pour $L = 3$ ou bien $L = 2$.

En résumé, nous recommandons de choisir :

- $L = 3$ pour un débit entre en dessous de 0,2 bpp
- $L = 4$ pour un débit entre 0.2 et 0.3 bpp
- $L = 5$ pour un débit entre 0.3 et 0.4 bpp
- $L = 6$ pour les débits supérieurs à 0.4 bpp

4.5 Comparaison avec la méthode proposée par Lecuire et al [7]

En vue de l'évaluation des résultats obtenus par les transformations proposées (BDCT zonale et CBDCT zonale), nous procédons à une comparaison avec les résultats obtenus par la transformation zonale basée sur l'algorithme de Cordic-Loeffler [7] (ZCL). Notons que cette dernière transformation (ZCL), a prouvé son efficacité aussi bien en matière de qualité d'images que pour la consommation énergétique. En fait, elle est considérée comme un travail de référence dans la littérature des réseaux de capteurs d'images.

La comparaison est faite en termes des caractéristiques débits – distorsion et en termes d'efficacité énergétique.

4.5.1 Comparaison en fonction du paramètre 'L'

En fixant le débit à 0.5 bpp, nous avons compressé les images de test en utilisant les transformations BDCT, CB-DCT. Les tableaux 4.8-(a) et (b) représentent les résultats obtenus par ces deux transformations respectivement. Les résultats de ZCL sont montrés dans le tableau 4.8-(c)

PSNR (db) – BDCT zonale					
L	Lena	Baboon	Peppers	Goldhill2	Moyenne
L = 2	28,11	20,84	26,01	26,6	25,39
L = 3	29,81	21,84	27,42	28,1	26,79
L = 4	32,12	22,61	29,44	29,5	28,42
L = 5	32,57	23	30,01	29,84	28,86
L = 6	32,92	23,15	30,61	30,11	29,20
L = 7	33,15	23,22	31,21	30,23	29,45
L = 8	33,25	23,24	31,82	30,25	29,64

(a)

PSNR (db) – CB DCT zonale					
L	Lena	Baboon	Peppers	Goldhill2	Moyenne
L = 2	28,84	20,98	26,76	27,02	25,90
L = 3	31,02	22,02	28,75	28,78	27,64
L = 4	32,32	22,8	30,03	29,64	28,70
L = 5	32,82	23,22	30,7	30,01	28,91
L = 6	33,17	23,39	30,61	30,21	29,35
L = 7	33,5	23,44	31,97	30,32	29,81
L = 8	33,69	23,44	32,27	30,38	29,95

(b)

PSNR (db) - ZCL					
L	Lena	Baboon	Peppers	Goldhill2	Moyenne
L = 2	28,19	21,06	27,47	27,3	26,01
L = 3	30,48	22,34	30	29,52	28,09
L = 4	33,51	23,26	31,93	30,3	29,75
L = 5	34	23,64	32,1	30,4	30,04
L = 6	34,25	23,65	33	30,6	30,38
L = 7	34,26	23,67	33,42	31,01	30,59
L = 8	34,26	23,68	33,42	31,02	30,60

(c)

Tableau 4.8: Résultats de décompression des différentes images de tests par la transformée :(a)-BDCT zonale, (b)-CBDCT zonale et (c)-Z-CL DCT. Les résultats sont donnés pour L = 2,3,4,5,6,7 et 8 pour un bitrate de 0.5 bpp.

D’après les résultats, présentés dans les tableaux ci-dessus:

Quelle que soit la valeur de L, et pour le même débit, nous constatons visuellement (voir figure 4.12 et 4.13) et dans les valeurs du PSNR que les performances des deux méthodes proposées sont très proches de ZCL. La dégradation moyenne par rapport à ZCL en termes de PSNR est au voisinage de 1 dB et 0.7 dB respectivement pour BDCT zonale et CBDCT

zonale. Alors que dans certains cas les dégradations sont relativement faibles comme par exemple :

- L'image Goldhille2 où la dégradation en termes de PSNR est 0.49 dB ($L = 6$) et 0.28 dB ($L=2$) respectivement pour BDCT zonale et CBDCT zonale.
- L'image Baboon où la dégradation moyenne en termes de PSNR est 0.48 dB et 0.22 dB respectivement pour BDCT zonale et CBDCT zonale.



(a)
PSNR = 30,21 dB



(b)
PSNR = 30,60 dB



(c)
PSNR = 30,09 dB

Figure 4.12: image Goldhill codée à 0.5 bpp et pour $L = 6$, par les transformées : (a)-CBDCT zonale, (b)-ZCL et (c)-BDCT zonale.



(a)
PSNR = 27,02 dB



(b)
PSNR = 27,3 dB



(c)
PSNR = 26,6 dB

Figure 4.13: image Goldhill codée à 0.5 bpp et pour $L = 2$, par les transformées : (a)-CB-DCT zonale, (b)-ZCL et (c)-BDCT zonale.

Les résultats montrés dans les Tableaux 4.7-(a), (b) et (c) sont représentés sous forme de courbes ‘débit –distorsion’ sur la Figure 4.14. Ces courbes donnent la variation de la distorsion obtenue (PSNR) après reconstruction de l’image en fonction de la taille de la zone

(paramètre L). Les courbes ont été tracées en prenant la moyenne du PSNR des images de tests et en fixant le bitrate à 0.5 bpp.

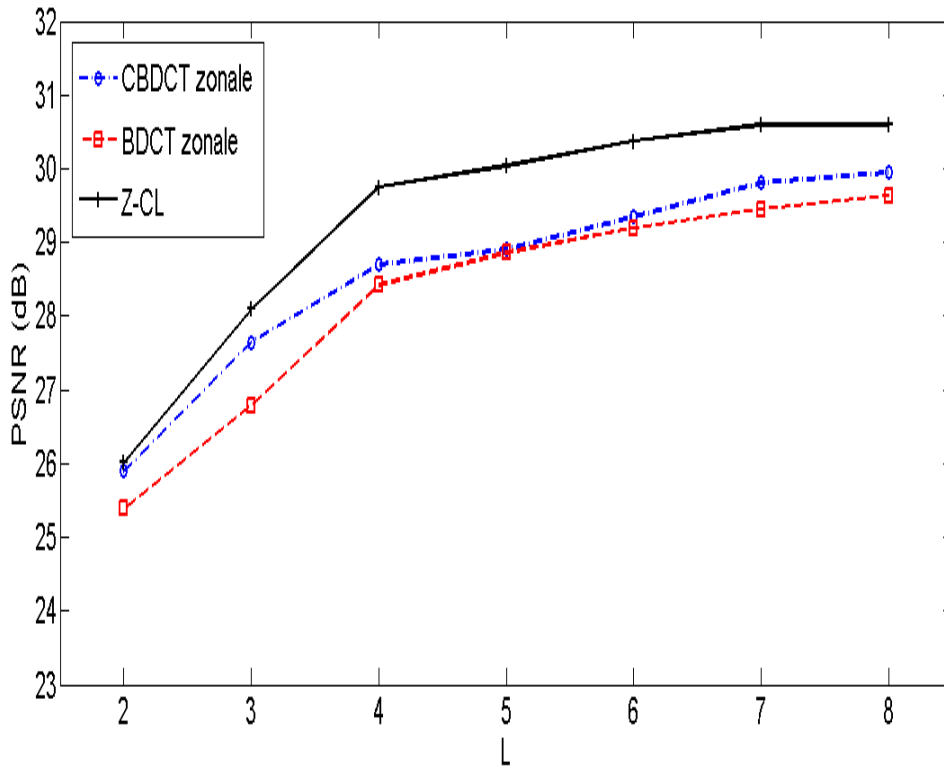


Figure 4.14: Courbes PSNR moyenne des algorithmes CB-DCT zonale, BDCT zonale et ZCL en fonction du paramètre 'L'. Les résultats sont donnés pour un bitrate de 0.5 bpp

Les courbes représentées ci-dessus montrent une appréciation globale des PSNR moyennes vis-à-vis le paramètre L. À partir de ces courbes nous pouvons affirmer que les deux méthodes proposées possèdent des performances en termes de PSNR proches de la méthode ZCL. Les dégradations, en moyenne, sont acceptables visuellement, et ne sont pas notables. Notons que, en dépit de ces dégradations, les économies énergétiques des méthodes proposées sont significatives (voir tableau 4.9):

- Pour L = 8 : 60.18 et 62.25 % respectivement pour BDCT zonale et CBDCT zonale
- Pour L = 7 : 56.83 et 59.58 % respectivement pour BDCT zonale et CBDCT zonale
- Pour L = 6 : 57.08 et 61.54 % respectivement pour BDCT zonale et CBDCT zonale
- Pour L = 5 : 58.17 et 63.65 % respectivement pour BDCT zonale et CBDCT zonale
- Pour L = 4 : 59.32 et 62.25 % respectivement pour BDCT zonale et CBDCT zonale

- Pour $L = 3$: 64.56 et 73.38 % respectivement pour BDCT zonale et CBDCT zonale
- Pour $L = 2$: 63.67 et 69.83 % respectivement pour BDCT zonale et CBDCT zonale

En résumé, environ 60 % et 64 % d'économies en énergie et temps d'exécution ont été obtenues en utilisant les deux méthodes proposées (BDCT zonale et CBDCT zonale respectivement), en dépit d'une baisse moyenne acceptable inférieure à 1 dB. Notons que la méthode CBDCT zonale est plus performante que la méthode BDCT zonale, en termes de PSNR et aussi bien en termes d'énergie et temps d'exécution.

	Consommation énergétique (μJ)		
	ZCL	BDCT zonale	CBDCT zonale
$L = 8$ (réf)	14.49	5.77	5.47
$L = 7$	12.00	5.18	4.85
$L = 6$	10.53	4.52	4.05
$L = 5$	9.49	3.97	3.45
$L = 4$	8.53	3.47	3.22
$L = 3$	7.85	2.78	2.09
$L = 2$	6.00	2.18	1.81

Tableau 4.9 Consommation énergétique des différents algorithmes appliqués sur un bloc 8×8 obtenu par le simulateur Atmel Studio avec le microcontrôleur ATmega128L.

Les courbes représentées dans la figure 4.14 peuvent être analysées à PSNR égal. Il est clair que le L nécessaire pour atteindre un PSNR donné en utilisant les deux méthodes proposées est inférieur au L nécessaire pour atteindre le même PSNR en utilisant la méthode ZCL. Ceci va permettre aux deux méthodes proposées de consommer une énergie plus faible que celle consommée par ZCL, et aussi de prendre moins de temps pour s'exécuter. Une illustration de cette analyse est montrée dans le tableau 4.10.

Il est évident à partir du tableau ci-dessus que les deux méthodes proposées sont plus performantes que ZCL. Pour une qualité d'image similaire, la consommation énergétique et le temps d'exécution sont réduits par une moyenne de 52% et 61 % respectivement pour BDCT zonale et CBDCT zonale. Cette réduction va permettre d'accroître considérablement la durée de vie du réseau de capteurs d'images.

Algorithme	Temps d'exécution (μ s)	Consommation énergétique (μ J)	PSNR (db)
Z-CL	261	6,00	21.06
	341	7.85	22.34
	371	8.53	23.26
BDCT zonale	120	2.78	21.84
	150	3.47	22.61
	225	5.18	23.22
CBDCT zonale	91	2.09	22.02
	140	3.22	22.80
	150	3.45	23.22

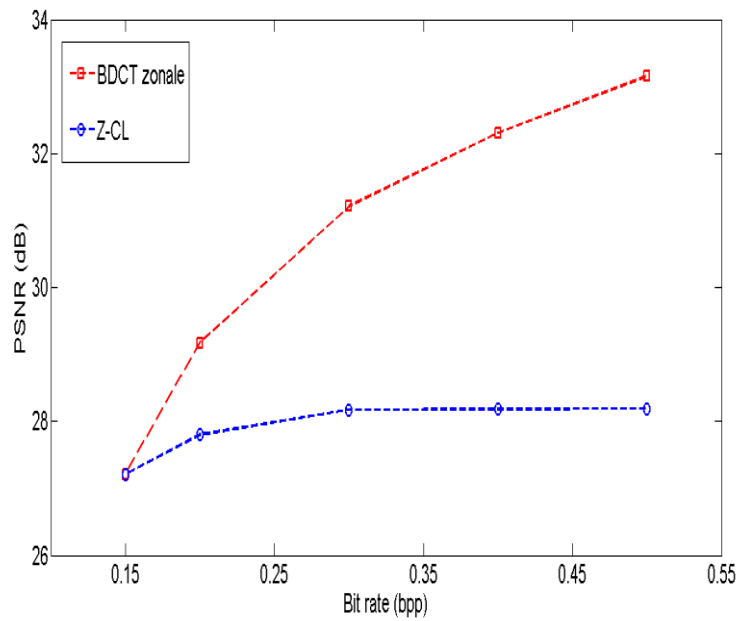
Tableau 4.10: Consommation énergétique et temps d'exécution pour le calcul des algorithmes : Z-CL, CBDCT zonale et BDCT zonale. Les résultats sont donnés pour l'image 'Baboon' pour un bitrate de 0.5 bpp

4.5.2 Comparaison à consommation énergétique égale

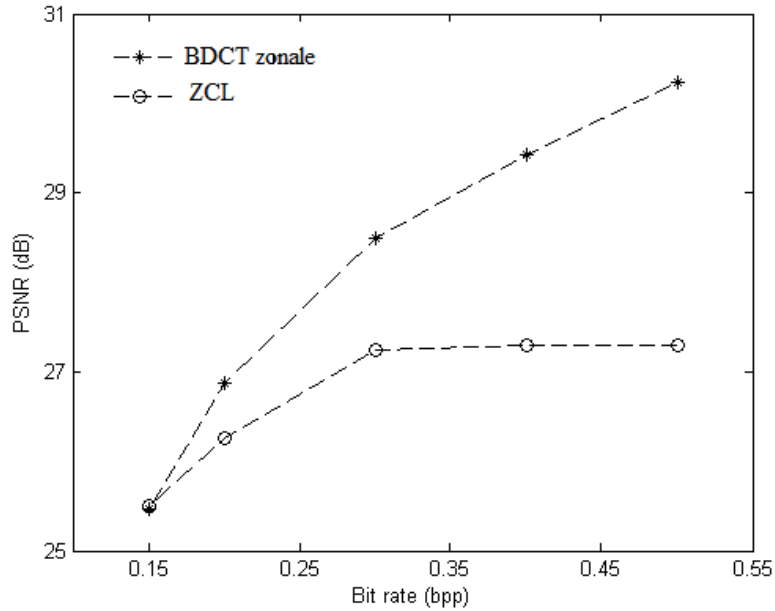
Les courbes ci-dessous montrent les caractéristiques 'débit - distorsion' pour la même consommation énergétique, respectivement pour l'image Lena, et Goldhill2. Ces courbes, montrent clairement que la méthode proposée 'BDCT zonale' est beaucoup plus performante. En fait, des gains importants en termes de PSNR peuvent être obtenus. Dans certains cas le gain dépasse les 3 dB, comme par exemple :

- L'image Lena où le PSNR est de 5 dB à 0.5 bpp, 4 dB à 0.4 bpp
- L'image Goldhill2 où le PSNR est de 3.5 dB à 0.5 bpp, 3 dB à 0.4 bpp

Notons que ces résultats sont valables pour un grand nombre d'images de tests et ne dépendent pas du type d'image. En ce qui concerne la 2ème méthode proposée, 'CBDCT zonale', non prise en compte elle est plus performante que la méthode BDCT zonale.



(a)



(b)

Figure 4.15: Courbes débit – distorsion des algorithmes BDCT zonale et ZCL pour les images : (a)-Lena, (b)- Goldhill2. Les résultats sont donnés pour une consommation énergétique = 6 μ J pour ZCL et 5,77 μ J pour BDCT zonale.

4.6 Table de quantification dyadique

En vue de réduire encore la complexité de la chaîne de compression JPEG, nous avons testé la table de quantification dyadique montrée ci-après.

$$Q = \begin{pmatrix} 16 & 16 & 16 & 16 & 32 & 64 & 64 & 64 \\ 16 & 16 & 16 & 16 & 32 & 64 & 64 & 64 \\ 16 & 16 & 16 & 32 & 32 & 64 & 64 & 64 \\ 16 & 16 & 32 & 32 & 32 & 64 & 64 & 64 \\ 32 & 32 & 32 & 64 & 128 & 128 & 128 & 128 \\ 64 & 64 & 64 & 64 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \\ 128 & 128 & 128 & 128 & 128 & 128 & 128 & 128 \end{pmatrix}$$

Cette table ne contient que des multiples de deux. Ceci va permettre d'implémenter cette dernière en n'utilisant que des opérations de décalages et d'additions. Notons que la table de quantification standard de JPEG nécessite une complexité assez importante lors de son implémentation.

Nous avons testé la table dyadique dans une chaîne de compression JPEG en utilisant l'algorithme de Loeffler pour l'étape de la transformation et le codage de Huffman. Le tableau 4.11 montre les résultats obtenus pour différents débits.

PSNR (dB)							
bpp	Barbara	Lena	Baboon	Boat	Cameraman	Peppers	Moyenne
0,1	19,97	23,4	18,81	19,83	19,37	21,03	20,40
0,2	23,86	30,66	20,97	26,33	24,53	28,93	25,88
0,3	25,53	33,31	22,17	28,48	26,61	31,6	27,95
0,4	27,02	34,96	23,01	29,87	27,86	32,93	29,28
0,5	28,27	36,02	23,69	30,97	28,98	33,77	30,28

Tableau 4.11: Résultats de décompression des différentes images de tests par l'algorithme de Loeffler et la table de quantification dyadique

A partir des résultats du tableau 4.11, nous pouvons constater que les valeurs des PSNR sont très proches de celles obtenues avec la table standard de JPEG (voir tableau 4.5-(f)). Ainsi :

- Les dégradations sont négligeables pour les débits allant de 0.5 à 0.2 bpp. Elles ne dépassent pas les 0.1 dB.
- Pour les débits inférieurs à 0.2 dB, ces dégradations sont au voisinage de 0.4 dB

Les résultats montrés dans les Tableaux 4.11 et 4.5-(f) sont représentés sous forme de courbes ‘débit –distorsion’ sur la Figure 4.17. Les courbes ont été tracées en prenant la moyenne du PSNR des images des tests.

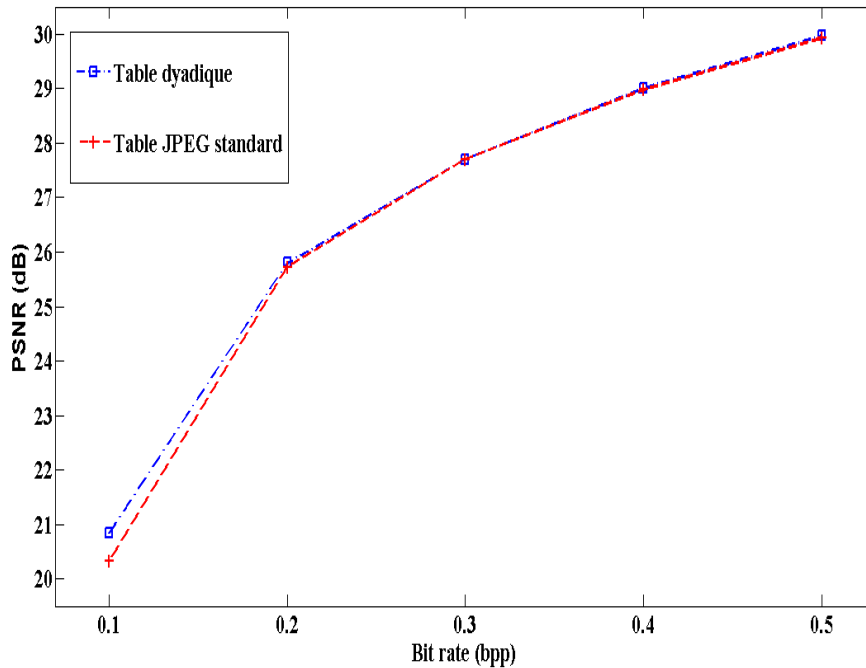


Figure 4.17: Courbe débit – distorsion moyenne : comparaison entre la table de quantification dyadique et la table de quantification standard

À partir de ces courbes nous pouvons affirmer que les deux chaînes testées possèdent des performances similaires en termes de PSNR. Les deux courbes sont pratiquement superposées pour les bitrates supérieures à 0.2 bpp. Pour les bitrates inférieures à 0.2 bpp, une dégradation légère de la méthode qui utilise la table dyadique est notée.

4.7 Conclusion

Dans ce chapitre nous avons évalué les performances des méthodes proposées. Dans un premier temps, nous avons étudié l’impact de la diminution du paramètre L sur la qualité visuelle de l’image. Nous nous sommes basés sur le fait que la qualité visuelle de l’image est fonction du taux de compression. Autrement dit, plus l’image est compressée, plus les composantes dans un bloc deviennent non significatifs (égaux à zéro), et par conséquent nous pouvons réduire préalablement le nombre de coefficients à calculer. En fait, nous avons proposé l’utilisation d’un L optimal pour chaque débit. Nous avons testé les transformations proposées dans le chapitre 3 en adoptant la chaîne de compression JPEG.

Nous avons constaté que le paramètre L peut être diminué selon le débit, sans perdre significativement de la qualité de l'image par rapport aux algorithmes de références ($L = 8$). Pour démontrer encore l'efficacité des méthodes proposées, nous les avons comparées avec les travaux récemment publiés dans [7]. Grâce à l'utilisation des méthodes proposées, le gain en coût d'énergie obtenu est approximativement 60% et 64 % respectivement pour BDCT zonale et CBDCT zonale. Les méthodes proposées fournissent le meilleur compromis entre qualité des images reconstruites et consommation énergétique.

Enfin, nous avons testé une table de quantification dyadique dans une chaîne de compression JPEG. Les résultats en termes de PSNR sont presque similaires avec la table standard de JPEG. En contrepartie, cette table est réputée être de faible complexité et ne demande que des opérations de décalages et des additions lors de son implémentation, ce qui réduit le cout énergétique de son exécution.

Conclusions et perspectives

Conclusions et perspectives

Durant cette dernière décennie, les réseaux de capteurs sans fils ont attiré l'intérêt d'entrepreneurs, universitaires et industriels surtout par rapport à la grande diversité d'applications que cette technologie met à notre disposition. Toutefois, les réseaux de capteurs sans fils doivent aussi résister à des défis importants dont, principalement, leurs capacités de calcul et de stockage réduites et surtout leur énergie embarquée limitée. L'énergie étant une ressource critique, elle constitue souvent un obstacle majeur au déploiement de ce type de réseaux. La gestion de la consommation énergétique constitue donc l'objectif le plus recherché dans ce domaine.

Cette thèse a porté sur l'efficacité énergétique dans les réseaux de capteurs sans fils, avec une insistance particulière sur la compression d'images locale, c'est-à-dire la compression au niveau des nœuds capteurs. Pour cela, nous avons proposé deux nouveaux algorithmes de compression d'images efficaces en énergie. Ces algorithmes, basés sur le standard JPEG, fournissent des caractéristiques 'débit - distorsion' très proches des algorithmes de références tout en réduisant la consommation d'énergie. Les critères de performance prennent en compte la qualité des images reconstruites, le temps d'exécution ainsi que la consommation énergétique.

L'étape de transformation étant la plus coûteuse en matière d'énergie, nos algorithmes de compression proposés s'intéressent essentiellement à réduire la complexité de cette étape. Pour cela ils combinent des transformations qui ont été proposées récemment (Binary DCT[153] et CB-DCT[155]) et une approche zonale [134-135] pour la sélection des coefficients. Les deux transformations BDCT et CB-DCT visent à réduire le nombre d'opérations nécessaires pour calculer les coefficients de la DCT, tandis que l'approche zonale permet de réduire préalablement le nombre de coefficients à traiter. Ces réductions ont entraîné une économie considérable d'énergie et de temps d'exécution.

Les algorithmes de compression d'image proposés dans le cadre de cette thèse, comparé à un algorithme de référence dans le domaine, à savoir ZCL [7], ont montré des performances bien supérieures. Ils ont prouvé une efficacité en termes d'énergie et aussi en termes de qualité des images reconstruites.

Les travaux présents tout au long de cette thèse sont probants puisqu'on arrive à des économies très importantes en termes d'énergie et de temps d'exécution. Plusieurs

perspectives peuvent donc être envisagées sur la base de ces travaux. Nous les structurons comme suit :

- Pour montrer l'efficacité de nos propositions, l'implantation matérielle de la chaîne de compression peut être adoptée sur circuit programmable. Dans la littérature, plusieurs méthodes de compression ont été synthétisées sur des circuits basés sur la technologie ASIC [169-171]
- Continuer de chercher à réduire la complexité des algorithmes de compression et/ou tester et proposer des nouvelles transformations. Une longue liste d'approximations DCT est disponible dans littérature (par exemple [157-159]). L'évaluation de ces dernières doit toujours être faite sur la base des caractéristiques '**énergie –distorsion**' plutôt que des caractéristiques '**débit-distorsion**'
- Etudier le codage et la transmission des régions d'intérêt (ROI ou Region Of Interest) des images fixes afin de donner plus d'importance à ces régions lors de la compression. Ces régions seront compressées avec le moins de dégradations possibles. le reste, notamment l'arrière-plan, peut subir une compression plus élevée et avec une dégradation importante. Pour cela il faut penser à extraire ces régions d'une manière automatique ou semi-automatique. Plusieurs méthodes de détection et d'extraction existent [172-173], mais elles possèdent une complexité calculatoire élevée. Il faut donc travailler à réduire ces complexités et les adapter aux nœuds capteurs [174-175].
- Procéder à l'étude de la transmission d'images sur des réseaux de capteurs sans fils sous la contrainte de l'énergie, tout en tenant compte des pertes de paquets [176]. Pour lutter contre ces pertes, plusieurs techniques d'entrelacement existent dans la littérature [4].
- Du point de vue du réseau, faire les simulations en prenant en compte des réseaux à grande échelle.

Références

Références

- [1] Raghunathan, V., Schurgers, C., Park, S., & Srivastava, M. B. (2002). Energy-aware wireless microsensor networks. *Signal Processing Magazine, IEEE*, 19(2), 40-50.
- [2] Shih, E., Cho, S. H., Ickes, N., Min, R., Sinha, A., Wang, A., & Chandrakasan, A. (2001, July). Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking* (pp. 272-287). ACM.
- [3] Ferrigno, L., Marano, S., Paciello, V., & Pietrosanto, A. (2005, July). Balancing computational and transmission power consumption in wireless image sensor networks. In *Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2005. VECIMS 2005. Proceedings of the 2005 IEEE International Conference on* (pp. 6-pp). IEEE.
- [4] Duran-Faundez, C., Lecuire, V., & Lepage, F. (2011). Tiny block-size coding for energy-efficient image compression and communication in wireless camera sensor networks. *Signal Processing: Image Communication*, 26(8), 466-481.
- [5] Taylor, C. N., Panigrahi, D., & Dey, S. (2002, January). Design of an adaptive architecture for energy efficient wireless image communication. In *Embedded processor design challenges* (pp. 260-273). Springer Berlin Heidelberg.
- [6] Kouadria, N., Doghmane, N., Messadeg, D., & Harize, S. (2013). Low complexity DCT for image compression in wireless visual sensor networks. *Electronics Letters*, 49(24), 1531-1532.
- [7] Lecuire, V., Makkaoui, L., & Moureaux, J. M. (2012). Fast zonal DCT for energy conservation in wireless image sensor networks. *Electronics letters*, 48(2), 125-127.
- [8] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. *Communications magazine, IEEE*, 40(8), 102-114.
- [9] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer networks*, 38(4), 393-422.
- [10] Cerpa, A., & Estrin, D. (2004). ASCENT: Adaptive self-configuring sensor networks topologies. *mobile computing, IEEE transactions on*, 3(3), 272-285.
- [11] Krishnamachari, B. (2005). *Networking wireless sensors*. Cambridge University Press.
- [12] Karl, H., & Willig, A. (2007). *Protocols and architectures for wireless sensor networks*. John Wiley & Sons.
- [13] Boukerche, A., Turgut, B., Aydin, N., Ahmad, M. Z., Bölöni, L., & Turgut, D. (2011). Routing protocols in ad hoc networks: A survey. *Computer Networks*, 55(13), 3032-3080.

- [14] Arampatzis, T., Lygeros, J., & Manesis, S. (2005, June). A survey of applications of wireless sensors and wireless sensor networks. In *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation* (pp. 719-724). IEEE.
- [15] Akyildiz, I. F., Melodia, T., & Chowdhury, K. R. (2007). A survey on wireless multimedia sensor networks. *Computer networks*, 51(4), 921-960.
- [16] Raghunathan, V., Schurgers, C., Park, S., & Srivastava, M. B. (2002). Energy-aware wireless microsensor networks. *Signal Processing Magazine, IEEE*, 19(2), 40-50.
- [17]. Xu, Y., Heidemann, J., & Estrin, D. (2001, July). Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th annual international conference on Mobile computing and networking* (pp. 70-84). ACM.
- [18] Hurni, P., & Braun, T. (2009). Calibrating wireless sensor network simulation models with real-world experiments. In *NETWORKING 2009* (pp. 1-13). Springer Berlin Heidelberg.
- [19] Pottie, G. J., & Kaiser, W. J. (2000). Wireless integrated network sensors. *Communications of the ACM*, 43(5), 51-58.
- [20] Alippi, C., Anastasi, G., Galperti, C., Mancini, F., & Roveri, M. (2007, October). Adaptive sampling for energy conservation in wireless sensor networks for snow monitoring applications. In *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE International Conference on* (pp. 1-6). IEEE.
- [21] Ye, W., Heidemann, J., & Estrin, D. (2004). Medium access control with coordinated adaptive sleeping for wireless sensor networks. *Networking, IEEE/ACM Transactions on*, 12(3), 493-506.
- [22] Tilak, S., Abu-Ghazaleh, N. B., & Heinzelman, W. (2002). A taxonomy of wireless micro-sensor network models. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(2), 28-36.
- [23] Chen, Y., & Zhao, Q. (2005). On the lifetime of wireless sensor networks. *Communications Letters, IEEE*, 9(11), 976-978.
- [24] M. Khan and J. Misic. On the lifetime of wireless sensor networks. In Y. Zhang, J. Zheng, and H. Hu, editors, *Wireless Networks and Mobile Communications*, volume 6, BocaRaton, FL, 2008. Auerbach Publications, CRC Press.
- [25] Tian, D., & Georganas, N. D. (2002, September). A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications* (pp. 32-41). ACM.

- [26] Soro, S., & Heinzelman, W. B. (2005, April). Prolonging the lifetime of wireless sensor networks via unequal clustering. In *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International* (pp. 8-pp). IEEE.
- [27] Blough, D. M., & Santi, P. (2002, September). Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks. In *Proceedings of the 8th annual international conference on Mobile computing and networking* (pp. 183-192). ACM.
- [28] Kumar, S., Arora, A., & Lai, T. H. (2005, November). On the lifetime analysis of always-on wireless sensor network applications. In *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on* (pp. 3-pp). IEEE.
- [29] Zhang, H., & Hou, J. C. (2005). Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc & Sensor Wireless Networks*, 1(1-2), 89-124.
- [30] Zhang, H., & Hou, J. (2004, May). On deriving the upper bound of α -lifetime for large sensor networks. In *Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing* (pp. 121-132). ACM.
- [31] Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000, January). Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on* (pp. 10-pp). IEEE.
- [32] Anastasi, G., Conti, M., Di Francesco, M., & Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3), 537-568.
- [33] T. Armstrong, Wake-up based power management in multi-hop wireless networks. <http://www.eecg.toronto.edu/~trevor/Wakeup/index.html>.
- [34] Zheng, R., Hou, J. C., & Sha, L. (2003, June). Asynchronous wakeup for ad hoc networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing* (pp. 35-45). ACM.
- [35] Ditzel, M., & Elferink, F. H. (2006, September). Low-power radar for wireless sensor networks. In *Radar Conference, 2006. EuRAD 2006. 3rd European* (pp. 139-141). IEEE.
- [36] Wu, H., & Pan, Y. (Eds.). (2008). *Medium access control in wireless networks*(Vol. 8). Nova Publishers.
- [37] Ye, W., & Heidemann, J. (2004). Medium access control in wireless sensor networks. *Wireless sensor networks*, 73-91.
- [38] Becker, M., Beylot, A. L., Dhaou, R., Gupta, A., Kacimi, R., & Marot, M. (2009). Experimental study: Link quality and deployment issues in wireless sensor networks. In *NETWORKING 2009* (pp. 14-25). Springer Berlin Heidelberg.

- [39] Li, J., & Mohapatra, P. (2007). Analytical modeling and mitigation techniques for the energy hole problem in sensor networks. *Pervasive and Mobile Computing*, 3(3), 233-254.
- [40] Anastasi, G., Conti, M., Passarella, A., & Pelusi, L. (2008). Mobile-relay forwarding in opportunistic networks. *Adaptation and Cross Layer Design in Wireless Networks*.
- [41] Chakrabarti, A., Sabharwal, A., & Aazhang, B. (2003, January). Using predictable observer mobility for power efficient design of sensor networks. In *Information Processing in Sensor Networks* (pp. 129-145). Springer Berlin Heidelberg.
- [42] Vuran, M. C., Akan, Ö. B., & Akyildiz, I. F. (2004). Spatio-temporal correlation: theory and applications for wireless sensor networks. *Computer Networks*, 45(3), 245-259.
- [43] Fasolo, E., Rossi, M., Widmer, J., & Zorzi, M. (2007). In-network aggregation techniques for wireless sensor networks: a survey. *Wireless Communications, IEEE*, 14(2), 70-87
- [44] Srisooksai, T., Keamarungsi, K., Lamsrichan, P., & Araki, K. (2012). Practical data compression in wireless sensor networks: A survey. *Journal of network and computer applications*, 35(1), 37-59.
- [45] Vijay Raghunathan, Saurabh Ganeriwal, and Mani Srivastava. Emerging techniques for long lived wireless sensor networks. *IEEE Communications Magazine*, 44(4) :108–114, April 2006
- [46] Liu, G., Tan, R., Zhou, R., Xing, G., Song, W. Z., & Lees, J. M. (2013, April). Volcanic earthquake timing using wireless sensor networks. In *Proceedings of the 12th international conference on Information processing in sensor networks* (pp. 91-102). ACM.
- [47] Ditzel, M., & Elferink, F. H. (2006, September). Low-power radar for wireless sensor networks. In *Radar Conference, 2006. EuRAD 2006. 3rd European* (pp. 139-141). IEEE
- [48] Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., & Anderson, J. (2002, September). Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications* (pp. 88-97). ACM.
- [49] DARPA IPTO, (1999.). SensIT : Sensor Information Technology program.
- [50] Xu, N., Rangwala, S., Chintalapudi, K. K., Ganesan, D., Broad, A., Govindan, R., & Estrin, D. (2004, November). A wireless sensor network for structural monitoring. In *Proceedings of the 2nd international conference on Embedded networked sensor systems* (pp. 13-24). ACM.
- [51] Allgood, G., Manges, W. W., & Smith, S. F. (1999). It's time for sensors to go wireless. *Sensors Magazine*, May.

- [52] Ieee standard for information technology part 15.4 (2003): Wireless LAN medium access control (MAC) and physical layer (PHY) specification for low rate wireless personal area networks (LR-WPANs).
- [53] Alliance, Z., & Specifications, Z. (2009). ZigBee Document 053474r17. *ZigBee Specification*.(January 2008) <http://www.zigbee.org>.
- [54] Misra, S., Reisslein, M., & Xue, G. (2008). A survey of multimedia streaming in wireless sensor networks. *Communications Surveys & Tutorials, IEEE*, 10(4), 18-39.
- [55] Schettini, R, C. Fernandez-Maloigne et S. Susstrunck (2003). Color image processing. *Pattern Recognition Letter*.
- [56] Duran-Faundez, C. (2009). *Transmission d'images sur les réseaux de capteurs sans fil sous la contrainte de l'énergie* (Doctoral dissertation, Université Henri Poincaré-Nancy I).
- [57] Kulkarni, P., Ganesan, D., & Shenoy, P. (2005, June). The case for multi-tier camera sensor networks. In *Proceedings of the international workshop on Network and operating systems support for digital audio and video* (pp. 141-146). ACM.
- [58] Kulkarni, P., Ganesan, D., Shenoy, P., & Lu, Q. (2005, November). SensEye: a multi-tier camera sensor network. In *Proceedings of the 13th annual ACM international conference on Multimedia* (pp. 229-238). ACM.
- [59] Arora, A., Dutta, P., Bapat, S., Kulathumani, V., Zhang, H., Naik, V., ... & Miyashita, M. (2004). A line in the sand: a wireless sensor network for target detection, classification, and tracking. *Computer Networks*, 46(5), 605-634.
- [60] Tezcan, N., & Wang, W. (2008). Self-orienting wireless multimedia sensor networks for occlusion-free viewpoints. *Computer networks*, 52(13), 2558-2567.
- [61] Srivastava, M, John, V, Deborah, E & Mohammad, R. (n.d.). Cyclops. http://research.cens.ucla.edu/projects/2007/Multiscaled_Actuated_Sensing/Cyclops/. CENS, Research Project.
- [62] Wawerla, J., Marshall, S., Mori, G., Rothley, K., & Sabzmeydani, P. (2009). Bearcam: Automated wildlife monitoring at the arctic circle. *Machine Vision and Applications*, 20(5), 303-317.
- [63] Horn, B. (1986). *Robot vision*. MIT press.
- [64] Kahn, J. M., Katz, R. H., & Pister, K. S. (1999, August). Next century challenges: mobile networking for "Smart Dust". In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking* (pp. 271-278). ACM.
- [65] Aboelaze, M., & Aloul, F. (2005, March). Current and future trends in sensor networks: a survey. In *Wireless and Optical Communications Networks, 2005. WOCN 2005. Second IFIP International Conference on* (pp. 551-555). IEEE.

- [66] Stankovic, J. A. (2004). Research challenges for wireless sensor networks. *ACM SIGBED Review*, 1(2), 9-12.
- [67] Wu, H., & Abouzeid, A. A. (2004, July). Power aware image transmission in energy constrained wireless networks. In *Computers and communications, 2004. Proceedings. ISCC 2004. Ninth international symposium on* (Vol. 1, pp. 202-207). IEEE.
- [68] Chen, P., Oh, S., Manzo, M., Sinopoli, B., Sharp, C., Whitehouse, K., ... & Sastry, S. (2006, May). Instrumenting wireless sensor networks for real-time surveillance. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on* (pp. 3128-3133). IEEE.
- [69] He, T., Vicaire, P., Yan, T., Luo, L., Gu, L., Zhou, G., ... & Abdelzaher, T. (2006, April). Achieving real-time target tracking using wireless sensor networks. In *Real-Time and Embedded Technology and Applications Symposium, 2006. Proceedings of the 12th IEEE* (pp. 37-48). IEEE.
- [70] Shnayder, V., Hempstead, M., Chen, B. R., Allen, G. W., & Welsh, M. (2004, November). Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the 2nd international conference on Embedded networked sensor systems* (pp. 188-200). ACM.
- [71] Chow, K. Y., Lui, K. S., & Lam, E. Y. (2006, January). Balancing image quality and energy consumption in visual sensor networks. In *Wireless Pervasive Computing, 2006 1st International Symposium on* (pp. 5-pp). IEEE.
- [72] Downes, I., Rad, L. B., & Aghajan, H. (2006). Development of a mote for wireless image sensor networks. *Proc. of COGNITIVE systems with Interactive Sensors (COGIS), Paris, France*.
- [73] Bhaskaran, V., & Konstantinides, K. (1997). *Image and video compression standards: algorithms and architectures*. Springer.
- [74] Ang, L. M., Seng, K. P., Chew, L. W., Yeong, L. S., & Chia, W. C. (2013). Multiple-View Information Reduction Techniques for WMSN Using Image Stitching. In *Wireless Multimedia Sensor Networks on Reconfigurable Hardware*(pp. 207-248). Springer Berlin Heidelberg.
- [75] Bovik, A. C. (2010). *Handbook of image and video processing*. Academic Press.
- [76] Salomon, D. (2004). *Data compression: the complete reference*. Springer.
- [77] Fischer, T. R. (1986). A pyramid vector quantizer. *Information Theory, IEEE Transactions on*, 32(4), 568-583.

- [78] Guillemot, L., Gaudeau, Y., Moussaoui, S., & Moureaux, J. (2008). Entropy-coded lattice vector quantization dedicated to the block mixture densities. *Image Processing, IEEE Transactions on*, 17(9), 1574-1586.
- [79] Gaudeau, Y., Guillemot, L., & Moureaux, J. M. (2008). Fast dead zone lattice vector quantisation. *Electronics Letters*, 44(3), 191-192.
- [80] Kimura, N., & Latifi, S. (2005, April). A survey on data compression in wireless sensor networks. In *Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on* (Vol. 2, pp. 8-13). IEEE.
- [81] Kusuma, J., Doherty, L., & Ramchandran, K. (2001, October). Distributed compression for sensor networks. In *ICIP (1)* (pp. 82-85).
- [82] Pradhan, S. S., Kusuma, J., & Ramchandran, K. (2002). Distributed compression in a dense microsensor network. *Signal Processing Magazine, IEEE*, 19(2), 51-60.
- [83] Arici, T., Gedik, B., Altunbasak, Y., & Liu, L. (2003, October). PINCO: a pipelined in-network compression scheme for data collection in wireless sensor networks. In *Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on* (pp. 539-544). IEEE.
- [84] Sadler, C. M., & Martonosi, M. (2006, October). Data compression algorithms for energy-constrained devices in delay tolerant networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems* (pp. 265-278). ACM.
- [85] Maniezzo, D., Yao, K., & Mazzini, G. (2002). Energetic trade-off between computing and communication resource in multimedia surveillance sensor network. In *Mobile and Wireless Communications Network, 2002. 4th International Workshop on* (pp. 373-376). IEEE.
- [86] Wallace, G. K. (1991). The JPEG still picture compression standard. *Communications of the ACM*, 34(4), 30-44.
- [87] Romberg, J. (2008). Imaging via compressive sampling [introduction to compressive sampling and recovery via convex programming]. *IEEE Signal Processing Magazine*, 25(2), 14-20.
- [88] Candes, E. J., Romberg, J. K., & Tao, T. (2006). Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8), 1207-1223.
- [89] Thyagarajan, K. S. (2011). *Still Image and video compression with MATLAB*. John Wiley & Sons.

- [90] Gersho, A. (1982). On the structure of vector quantizers. *Information Theory, IEEE Transactions on*, 28(2), 157-166.
- [91] Gersho, A., & Gray, R. M. (1992). *Vector quantization and signal compression*. Springer.
- [92] Welstead, S. T. (1999). *Fractal and wavelet image compression techniques* (p. 232). (Bellingham, WA: SPIE Optical Engineering Press.
- [93] Sayood, K. (2012). *Introduction to data compression*. Newnes.
- [94] Makkaoui, L. (2012). *Compression d'images dans les réseaux de capteurs sans fil* (Doctoral dissertation, Université de Lorraine).
- [95] Huffman, D. A. (1952). A method for the construction of minimum redundancy codes. *Proceedings of the IRE*, 40(9), 1098-1101.
- [96] Witten, I. H., Neal, R. M., & Cleary, J. G. (1987). Arithmetic coding for data compression. *Communications of the ACM*, 30(6), 520-540.
- [97] Golomb, SW. (1966). Run-length encodings. *IEEE Transactions on Information Theory*, 12, 399-401
- [98] Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing*. Prentice Hall.
- [99] Roma, N., & Sousa, L. (2011). A tutorial overview on the properties of the discrete cosine transform for encoded image and video processing. *Signal Processing*, 91(11), 2443-2464.
- [100] Loeffler, C., Ligtenberg, A., & Moschytz, G. S. (1989, May). Practical fast 1-D DCT algorithms with 11 multiplications. In *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on* (pp. 988-991). IEEE.
- [101] Heyne, B., & Götze, J. (2007). A low-power and high-quality implementation of the discrete cosine transformation. *Advances in Radio Science*, 5.
- [102] Liang, J., & Tran, T. D. (2001). Fast multiplierless approximations of the DCT with the lifting scheme. *Signal Processing, IEEE Transactions on*, 49(12), 3032-3044.
- [103] Britanak, V., Yip, P. C., & Rao, K. R. (2010). *Discrete cosine and sine transforms: general properties, fast algorithms and integer approximations*. Academic Press.
- [104] Ruiz, G. A., Michell, J. A., & Burón, A. (2006). High throughput parallel-pipeline 2-D DCT/IDCT processor chip. *Journal of VLSI signal processing systems for signal, image and video technology*, 45(3), 161-175.

- [105] Chiasserini, C. F., & Magli, E. (2002, September). Energy consumption and image quality in wireless video-surveillance networks. In *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on* (Vol. 5, pp. 2357-2361). IEEE.
- [106] White, S. A. (1989). Applications of distributed arithmetic to digital signal processing: A tutorial review. *ASSP Magazine, IEEE*, 6(3), 4-19.
- [107] Shams, A. M., Chidanandan, A., Pan, W., & Bayoumi, M. A. (2006). NEDA: A low-power high-performance DCT architecture. *Signal Processing, IEEE Transactions on*, 54(3), 955-964.
- [108] August, N. J., & Ha, D. S. (2004). Low power design of DCT and IDCT for low bit rate video codecs. *Multimedia, IEEE Transactions on*, 6(3), 414-422.
- [109] Xanthopoulos, T. (1999). *Low power data-dependent transform video and still image coding* (Doctoral dissertation, Massachusetts Institute of Technology).
- [110] Mammeri, A., Khoumsi, A., Ziou, D., & Hadjou, B. (2008, April). Energy-aware JPEG for visual sensor networks. In *The Maghrebian conference on software engineering and artificial intelligence MCSEAI* (pp. 1-7).
- [111] Yu, W., Sahinoglu, Z., & Vetro, A. (2004, December). Energy efficient JPEG 2000 image transmission over wireless sensor networks. In *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE* (Vol. 5, pp. 2738-2743). IEEE.
- [112] Wu, M., & Chen, C. W. (2003, October). Multiple bitstream image transmission over wireless sensor networks. In *Sensors, 2003. Proceedings of IEEE* (Vol. 2, pp. 727-731). IEEE.
- [113] Wolf, J. K., & Kurkoski, B. M. (2008). Slepian-Wolf coding. *Scholarpedia*, 3(11), 6789.
- [114] Wagner, R., Nowak, R., & Baraniuk, R. (2003, September). Distributed image compression for sensor networks using correspondence analysis and super-resolution. In *Image processing, 2003. ICIP 2003. Proceedings. 2003 international conference on* (Vol. 1, pp. I-597). IEEE.
- [115] Belongie, S., Malik, J., & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4), 509-522.
- [116] Gehrig, N., & Dragotti, P. L. (2004, September). Distributed compression in camera sensor networks. In *Multimedia Signal Processing, 2004 IEEE 6th Workshop on* (pp. 311-314). IEEE.

- [117] Wu, H., & Abouzeid, A. A. (2004, August). Energy efficient distributed JPEG2000 image compression in multihop wireless networks. In *Proc. of IEEE Workshop on Applications and Services in Wireless Networks* (pp. 152-160).
- [118] Lu, Q., Luo, W., Wang, J., & Chen, B. (2008). Low-complexity and energy efficient image compression scheme for wireless sensor networks. *Computer Networks*, 52(13), 2594-2603.
- [119] Claerbout, J. F., & Muir, F. (1973). Robust modeling with erratic data. *Geophysics*, 38(5), 826-844.
- [120] Candes, E., & Romberg, J. (2007). Sparsity and incoherence in compressive sampling. *Inverse problems*, 23(3), 969.
- [121] Qaisar, S., Bilal, R. M., Iqbal, W., Naureen, M., & Lee, S. (2013). Compressive sensing: From theory to applications, a survey. *Communications and Networks, Journal of*, 15(5), 443-456.
- [122] Girod, B., Aaron, A. M., Rane, S., & Rebollo-Monedero, D. (2005). Distributed video coding. *Proceedings of the IEEE*, 93(1), 71-83.
- [123] Wyner, A. D., & Ziv, J. (1976). The rate-distortion function for source coding with side information at the decoder. *Information Theory, IEEE Transactions on*, 22(1), 1-10.
- [124] Aaron, A., Setton, E., & Girod, B. (2003, September). Towards practical Wyner-Ziv coding of video. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on* (Vol. 3, pp. III-869). IEEE.
- [125] Aaron, A., Rane, S., Zhang, R., & Girod, B. (2003, March). Wyner-Ziv coding for video: Applications to compression and error resilience. In *Data Compression Conference, 2003. Proceedings. DCC 2003* (pp. 93-102). IEEE.
- [126] Sheng, T., Hua, G., Guo, H., Zhou, J., & Chen, C. W. (2008, October). Rate allocation for transform domain Wyner-Ziv video coding without feedback. In *Proceedings of the 16th ACM international conference on Multimedia* (pp. 701-704). ACM.
- [127] Pudlewski, S., Prasanna, A., & Melodia, T. (2012). Compressed-sensing-enabled video streaming for wireless multimedia sensor networks. *Mobile Computing, IEEE Transactions on*, 11(6), 1060-1072.
- [128] Soon, I. Y., Koh, S. N., & Yeo, C. K. (1998). Noisy speech enhancement using discrete cosine transform. *Speech communication*, 24(3), 249-257.
- [129] P Vishwakarma, V., Pandey, S., & Gupta, M. N. (2010). An illumination invariant accurate face recognition with down scaling of DCT coefficients. *CIT. Journal of Computing and Information Technology*, 18(1), 53-67.

- [130] Huang, H. Y., Yang, C. H., & Hsu, W. H. (2010). A video watermarking technique based on pseudo-3-D DCT and quantization index modulation. *Information Forensics and Security, IEEE Transactions on*, 5(4), 625-637.
- [131] Koc, U. V., & Liu, K. R. (1998). DCT-based motion estimation. *Image Processing, IEEE Transactions on*, 7(7), 948-965.
- [132] Ahmed, N., Natarajan, T., & Rao, K. R. (1974). Discrete cosine transform. *Computers, IEEE Transactions on*, 100(1), 90-93.
- [133] Blinn, J. F. (1993). What's that deal with the DCT?. *Computer Graphics and Applications, IEEE*, 13(4), 78-83.
- [134] Wang, Z. (1991). Pruning the fast discrete cosine transform. *IEEE Transactions on Communications*, 39(5), 640-643.
- [135] Skodras, A. N. (1994). Fast discrete cosine transform pruning. *Signal Processing, IEEE Transactions on*, 42(7), 1833-1837.
- [136] Hou, H. S. (1986, December). A fast recursive algorithm for computing the discrete cosine transform. In *Applications of Digital Image Processing IX* (pp. 14-25). International Society for Optics and Photonics.
- [137] Kashef, B. G., & Habibi, A. (1984, December). Direct computation of higher-order DCT coefficients from lower-order DCT coefficients. In *28th Annual Technical Symposium* (pp. 425-431). International Society for Optics and Photonics.
- [138] Vetterli, M., & Nussbaumer, H. J. (1984). Simple FFT and DCT algorithms with reduced number of operations. *Signal processing*, 6(4), 267-278.
- [139] Tseng, B. D., & Miller, W. C. (1978). On computing the discrete cosine transform. *Computers, IEEE Transactions on*, 100(10), 966-968.
- [140] Narasimha, M. J., & Peterson, A. M. (1978). On the computation of the discrete cosine transform. *Communications, IEEE Transactions on*, 26(6), 934-936.
- [141] Hein, D., & Ahmed, N. A. S. I. R. (1978). On a real-time Walsh-Hadamard/cosine transform image processor. *Electromagnetic Compatibility, IEEE Transactions on*, (3), 453-457.
- [142] Haralick, M. R., (1976). A storage efficient way to implement the discrete cosine transform. *Computers, IEEE transaction on*, vol C25
- [143] Chen, H. C., Guo, J. I., Chang, T. S., & Jen, C. W. (2005). A memory-efficient realization of cyclic convolution and its application to discrete cosine transform. *Circuits and Systems for Video Technology, IEEE Transactions on*, 15(3), 445-453.

- [144] Meher, P. K. (2006). Systolic designs for DCT using a low-complexity concurrent convolutional formulation. *Circuits and Systems for Video Technology, IEEE Transactions on*, 16(9), 1041-1050.
- [145] W. H. Chen, CH Smith, and S. C. Fralick, (1974). A fast computational algorithm for the discrete cosine transforms. *Communications, IEEE Transactions on* 25(9)
- [146] Lee, B. G. (1984). A new algorithm to compute the discrete cosine transform. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 32(6),
- [147] Suehiro, N., & Hatori, M. (1986). Fast algorithms for the DFT and other sinusoidal transforms. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 34(3), 642-644.
- [148] Wang, Z. (1984). Fast algorithms for the discrete W transform and for the discrete Fourier transform. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 32(4), 803-816.
- [149] Ghanbari, M., & Pearson, D. E. (1982, February). Fast cosine transform implementation for television signals. In *IEE Proceedings F (Communications, Radar and Signal Processing)* (Vol. 129, No. 1, pp. 59-68). IET Digital Library.
- [150] Rabiner, L. R., & Gold, B. (1975). Theory and application of digital signal processing. *Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p., 1.*
- [151] Duhamel, P., & H'Mida, H. (1987, April). New 2 n DCT algorithms suitable for VLSI implementation. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'87.* (Vol. 12, pp. 1805-1808). IEEE.
- [152] Pai, C. Y., Lynch, W. E., & Al-Khalili, A. J. (2003, August). Low-power data-dependent 8× 8 DCT/IDCT for video compression. In *Vision, Image and Signal Processing, IEE Proceedings-* (Vol. 150, No. 4, pp. 245-255). IET.
- [153] Bouguezel, S., Ahmad, M. O., & Swamy, M. N. S. (2013). Binary discrete cosine and Hartley transforms. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 60(4), 989-1002.
- [154] Haweel, T. I. (2001). A new square wave transform based on the DCT. *Signal processing*, 81(11), 2309-2319.
- [155] Cintra, R. J., & Bayer, F. M. (2011). A DCT approximation for image compression. *Signal Processing Letters, IEEE*, 18(10), 579-582.
- [156] Higham, N. J. (1986). Computing the polar decomposition-with applications. *SIAM Journal on Scientific and Statistical Computing*, 7(4), 1160-1174.
- [157] Bouguezel, S., Ahmad, M. O., & Swamy, M. N. S. (2011, May). A low-complexity parametric transform for image compression. In *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on* (pp. 2145-2148). IEEE.

- [158] Bouguezel, S., Ahmad, M. O., & Swamy, M. N. S. (2010, August). A novel transform for image compression. In *IEEE International Midwest Symposium on Circuits and Systems* (pp. 509-512).
- [159] Bouguezel, S., Ahmad, M. O., & Swamy, M. N. S. (2008). Low-complexity 8×8 transform for image compression. *Electronics Letters*, 44(21), 1249-1250.
- [160] Malvar, H. S., Hallapuro, A., Karczewicz, M., & Kerofsky, L. (2003). Low-complexity transform and quantization in H. 264/AVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7), 598-603.
- [161] Kouadria N., Doghmane.N, &Messadeg, D. (2013, november). A low quality and a low complexity of JPEG algorithm based on integer DCT used in the Wireless Image Sensor Networks. *The International Conference on Signal, Image, Vision and their Application (SIVA)*, 126-129.
- [162] Kouadria N., Doghmane.N, &Messadeg, D., (2014, March). Low complexity image compression using integer DCT for wireless sensor networks. *International Conference on Artificial Intelligence and Information Technology (ICA2IT)*
- [163] ISO/IEC 10918-1/ITU-T Recommendation T.81. *Digital compression and coding of continuous-tone still images*. <http://www.jpeg.org/jpeg/>
- [164] Lee, D. U., Kim, H., Rahimi, M., Estrin, D., & Villasenor, J. D. (2009). Energy-efficient image compression for resource-constrained platforms. *Image Processing, IEEE Transactions on*, 18(9), 2100-2113.
- [165] Phamila Y, A. V., & Amutha, R. (2013). Low complexity energy efficient very low bit-rate image compression scheme for wireless sensor network. *Information Processing Letters*, 113(18), 672-676.
- [166] Polastre, J., Szewczyk, R., & Culler, D. (2005, April). Telos: enabling ultra-low power wireless research. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on* (pp. 364-369). IEEE.
- [167] Pedersen, M., & Hardeberg, J. Y. (2009). Survey of full-reference image quality metrics.
- [168]JUSP- SIPI Image database, university of southern California. www.sipi.usc.edu/database/
- [169]Chefi, A., Soudani, A., & Sicard, G. (2013). Contribution to the design of a CMOS image sensor with low-complexity video compression for wireless sensor networks. *Journal of Systems Architecture*, 59(10), 818-825.

- [170] Chefi, A., Soudani, A., & Sicard, G. (2013, June). A CMOS image sensor with low-complexity video compression for wireless sensor networks. In *New Circuits and Systems Conference (NEWCAS), 2013 IEEE 11th International* (pp. 1-4). IEEE.
- [171] Kaddachi, M. L., Soudani, A., Lecuire, V., Torki, K., Makkaoui, L., & Moureaux, J. M. (2012). Low power hardware-based image compression solution for wireless camera sensor networks. *Computer Standards & Interfaces*, 34(1), 14-23.
- [172] Benezeth, Y., Jodoin, P. M., Emile, B., Laurent, H., & Rosenberger, C. (2008, December). Review and evaluation of commonly-implemented background subtraction algorithms. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* (pp. 1-4). IEEE.
- [173] Piccardi, M. (2004, October). Background subtraction techniques: a review. In *Systems, man and cybernetics, 2004 IEEE international conference on* (Vol. 4, pp. 3099-3104). IEEE.
- [174] Pham, D. M., & Aziz, S. M. (2013). Object extraction scheme and protocol for energy efficient image communication over wireless sensor networks. *Computer Networks*, 57(15), 2949-2960.
- [175] Aziz, S., & Pham, D. (2013). Energy efficient image transmission in wireless multimedia sensor networks. *IEEE Communications Letters*
- [176] Boluk, P. S., Baydere, S., & Harmanci, A. E. (2011). Robust image transmission over wireless sensor networks. *Mobile Networks and Applications*, 16(2), 149-170.