

UNIVERSITE BLIDA 1

Faculté de Technologie
Département d'Électronique

THESE DE DOCTORAT

en Electronique

IMPLEMENTATION DES TRANSFORMEES EN ONDELETTES
ET DES RESEAUX DE NEURONES SUR DES ARCHITECTURES
HAUTEMENT PERFORMANTES

Par

FERHAT TALEB née ALIM FATIHA

Devant le jury composé de :

A. GUESSOUM	Professeur, U. Blida 1	Président
Y. SMARA	Professeur, USTHB, Alger	Examineur
A. NAMANE	Professeur, U. Blida 1	Examineur
M. BELHOCINE	Directeur de recherche, CDTA	Examineur
H. BESSALAH	Maitre de recherche, TIC	Rapporteur
A. SALHI	Professeur, U. Blida 1	Co-Rapporteur

Juillet, 2017

REMERCIEMENTS

Je remercie Dieu le tout puissant qui m'a donné le courage et la patience afin de pouvoir finir ce projet.

J'adresse mes sentiments respectueux et reconnaissants à Monsieur Hamid BESSALAH, Ex Ministre des technologies de l'information et de la communication ; de m'avoir lancé dans la voie de la recherche ; je tiens à le remercier pour m'avoir proposé ce sujet de recherche et d'avoir été toujours disponible par ses conseils, suggestions et recommandations, et cela malgré ses occupations.

Mes très sincères remerciements à mon Co-promoteur Hassen. SALHI professeur à l'université de Blida et mon enseignant, pour sa contribution dans le projet son aide et surtout ses encouragements pour finir cette thèse et ses judicieux conseils durant mes études d'ingénieur, de Magister et de Doctorat.

J'adresse également mes remerciements à Monsieur GUESSOUM. Abderrezak, professeur à l'université de Blida pour l'honneur qu'il m'a fait en présidant ce jury, ainsi que professeur NAMANE. Abderrahmane, aussi mes remerciements au professeur et directeur de recherche SMARA.Youcef à l'USTHB et à Monsieur BELHOCINE. Mahmoud directeur de recherche au CDTA ; à tous les membres du jury qui ont bien voulu prendre le temps de s'intéresser au sujet de cette thèse, et ont contribué à améliorer la qualité.

Je remercie vivement Professeur K.Bouyoucef ainsi que Professeur K.Boualga et S.Khoudri du C.A.C de Blida pour le temps et le travail qu'avaient nécessité leurs expertises.

Je tiens à exprimer ma gratitude et toute ma reconnaissance à mes collègues : Belkacem, Farid, Sofiane, Issad, Abdeladhime, qui m'ont été d'un grand support tout au long de mes travaux et pour l'ambiance de travail qu'ils ont su créer et grâce à leur serviabilité ; la difficulté du travail de recherche était surmontable (à notre solide et meilleur équipe A3Sp). A Wahiba, Ratiba, Linda, Khadija pour leurs encouragements et bonne amitié. Mes sincères remerciements à Monsieur A. HANTOUT pour ses précieux conseils, ainsi que pour la lecture critique de cette thèse.

J'adresse mes très sincères remerciements au Chef du Département d'Electronique Monsieur B. KAZED ; et au responsable de la post-graduation Docteur Habib pour son aide et sa modestie.

Je tiens également à assurer de mes remerciements tous les membres dirigeants de L'université Saad Dahlab de Blida ; Messieurs ; Le Recteur, le Vice-Recteur chargé de la PGRS, le Doyen de la Faculté des Sciences de l'Ingénieur, et mes anciens enseignants et personnel qui ont contribués à ma formation trouvent ici l'expression de ma profonde gratitude.

A tous les esprits scientifiques qui croient que le savoir est universel ; en nous guidant et en nous simplifiant la recherche scientifique.

إهداء

إلى مصدر نجاحي أمي و أبي

إلى قرة عيني زوجي و ابني عبد العزيز

إلى كل عائلتي

المجد و الخلود لشهدائنا الأبرار

RESUME

L'objectif général de cette recherche est l'implémentation d'une méthode hybride entre les transformées en ondelettes et les réseaux de neurones (WT-SOM) pour la compression des images médicales avec pertes. Dans ce travail ; nous sommes passés d'un niveau de représentation abstrait mathématique puis algorithmique de la transformée en ondelette et des réseaux de neurones à un niveau plus concret implémentation sur circuit Virtex II du FPGA.

Les transformées en ondelettes et les réseaux de neurones demeurent des outils largement utilisés dans les domaines de l'analyse spectrale, du filtrage numérique, de la reconnaissance des formes et de la compression. De ce fait, l'amélioration des performances et de leur exécution constitue une préoccupation majeure dans le contexte d'un traitement massif de l'information. A travers ces deux transformées, nous avons tenté de démontrer la nécessité de recourir à des méthodes de compression avec pertes en imagerie médicale qui, encore vus avec une certaine méfiance de la part des médecins, permettent aujourd'hui d'obtenir de bonnes performances tout en fournissant des images reconstruites d'une qualité appréciable.

Mots clés : Arithmétique en ligne, transformée en ondelette, réseaux de neurones, compression, images médicales, FPGA.

ABSTRACT

The overall objective of this research is the implementation of a hybrid between wavelet transforms and neural networks (WT-SOM) for medical image compression with losses. so we increased from a level of an abstract mathematical representation then algorithmic of the wavelet transforms and neural networks, to a more concrete circuit implementation on virtex ii FPGA.

Wavelet transforms and neural networks are still widely used tools in the fields of spectral analysis, digital filtering, the pattern recognition and compression. there by improving performance and their implementation is a major concern in the context of a mass processing of information. Through these two transforms, we have attempted to demonstrate the need for methods of lossy compression in medical imaging, still distrusted by some doctors, can now achieve good performance while providing the reconstructed images of a quality appreciated.

Keywords: arithmetic-on line wavelet transforms, neural networks, compression, medical images, FPGA.

ملخص

الهدف العام من هذا البحث هو تنفيذ اختزال بين تحويل الموجات والشبكات العصبية (WT-SOM) لضغط الصور الطبية. ثم انتقلنا من مستوى التمثيل الرياضي المجرد خوارزميات تحويل الموجات وكذا الشبكات العصبية لتنفيذ أكثر واقعية. في الدارة المنطقية المبرمجة FPGA (Field Programmable Gate Array) يعتبر تحويل الموجات والشبكات العصبية وسائل مستخدمة على نطاق واسع في مجالات التحليل الطيفي، والترشيح الرقمي، والتعرف على الأنماط والضغط، وبالتالي فان تحسين أداءها هو مصدر قلق كبير في مجال معالجة الكم الهائل من المعلومات. ومن خلال هذين التحويلين، حاولنا أن تظهر أنه من اللازم استخدام أساليب الضغط المحتوي على خسائر في ميدان التصوير الطبي، ولكن مع ذلك، لا يزال الأطباء ينظرون اليه بعين الحذر وعدم الثقة. مع أنه أصبح اليوم يحقق أداء جيد وفي الوقت نفسه يوفر صوراً ذات نوعية فائقة.

الكلمات الأساسية:

تحويل الموجات، حساب متتابع، الشبكات العصبية، ضغط، الصور الطبية، FPGA.

TABLE DES MATIERES

RESUME	VI
ABSTRACT	VI
ملخص	V
TABLE DES MATIERES.....	VI
LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX.....	X
LISTE DES ACRONYME	VI
INTRODUCTION.....	1
CHAPITRE 1 : ETAT DE L'ART DE LA COMPRESSION.....	5
1.1. Introduction générale	5
1.2. Natures des informations : la redondance	5
1.2.1. Niveau pixel.....	6
1.2.2. Niveau bloc.....	6
1.2.3. Niveau contexte local des blocs.....	7
1.2.4. Niveau scène.....	7
1.3. Les méthodes de compression.....	7
1.3.1. Les méthodes sans pertes « sans distorsion ou réversible.....	7
1.3.1.1 Codage de Huffman.....	8
1.3.1.2. Codage arithmétique.....	9
1.3.1.3. Le Codage par plage (<i>Run length Encoding : RLE</i>).....	9
1.3.2. Les méthodes avec pertes « avec distorsion ou non réversible.....	10
1.3.2.1 Codage par quantification.....	11
1.3.2.1.1. La quantification scalaire.....	11
1.3.2.1.2. La quantification vectorielle	12
1.3.2.2. Principe de la quantification	12
1.3.2.3. Le dictionnaire (Codebook)	12
1.3.2.4. Construction du dictionnaire.....	13
1.4. Algorithmes de construction du dictionnaire.....	14
1.4.1. Les algorithmes non-neuronaux	15
1.4.1.1. Algorithme de Lloyd Généralisé (<i>Generalized Lloyd Algorithm GLA</i>).....	15
1.4.1.2. Algorithme de Linde-Buzo-Gray (LBG).....	15
1.4.1.3. Agrégation constructive progressive	15

1.4.1.4. Quantification Vectorielle adaptative par ré-apprentissage	16
1.4.2. Les algorithmes neuronaux	16
1.4.2.1. Compétition simple.....	17
1.4.2.2. Compétition sensible à la fréquence	17
1.4.2.3. Carte auto-organisatrice (Carte de Kohonen).....	17
1.5. Codage par transformation.....	17
1.5.1. Transformation de Karhunen-loeve (KLT).....	18
1.5.2. Transformations spectrales ou sinusoïdales.....	18
1.5.3 Transformation en ondelettes.....	19
1.6. les critères d'évaluation de la qualité de compression.....	20
1.6.1 les critères d'évaluations objectives des méthodes sans pertes.....	20
1.6.2. les critères d'évaluation des méthodes avec pertes.....	21
1.6.2.1. Le taux de compression	22
1.6.2.2. L'évaluation de la distorsion	22
1.6.2.2.1. Erreur Quadratique Moyenne MSE (Mean Square Error).....	23
1.6.2.2.2. Rapport signal sur bruit crête PSNR (Peak Signal to Noise Ratio).....	23
1.6.2.2.3. Similarité structurelle SSIM (Structural SIMilarity)	23
1.6.3. Les critères d'évaluations subjectives.....	24
1.7. Conclusion.....	24
CHAPITRE 2 : PARTICULARITES DES IMAGES MEDICALES ET	
QUANTIFICATION VECTORIELLE.....	25
2.1. Introduction	25
2.2. Spécificités des images médicales.....	25
2.2.1. La radiographie.....	26
2.2.2. Le scanner (tomodensitométrie).....	26
2.2.3. L'échographie.....	26
2.2.4. L'imagerie par résonance magnétique (IRM).....	27
2.2.5. Tomographie par émission de positions(TEP).....	28
2.3. Les propriétés des images médicales.....	28
2.4. Compression réversible des images médicales	29
2.5. Compression irréversible des images médicales.....	29
2.5.1. Compression JPEG2000 avec pertes des images médicales.....	30
2.5.2. Méthodes de quantification vectorielle appliquées aux images médicales	30

2.6. Conclusion	31
CHAPITRE 3 : TRANSFORMÉES EN ONDELETTES ET RESEAUX DE NEURONES.....	32
3.1. Introduction.....	32
3.2. Introduction à la transformée en ondelettes.....	32
3.2.1. La transformée en ondelettes	34
3.3. Analyse multi résolution de Stéphane Mallat. (S. Mallat)	35
3.4. Algorithme de décomposition –reconstruction de S. Mallat.....	36
3.4.1. Décomposition/reconstruction	37
3.4.1.1. Décimation.....	37
3.4.1.2. Reconstruction.....	38
3.4.1.3. Interpolation par des zéros	38
3.5. Calcul de complexité.....	38
3.6. Carte auto-organisatrice (Carte de Kohonen).....	39
3.6.1. Algorithme d'apprentissage.....	39
3.6.2. Propriétés de la carte auto-organisatrice	41
3.7. Evaluation de l'algorithme de s. Mallat par l'arithmétique en ligne	43
3.7.1. Introduction.....	43
3.7.2. mode parallèle.....	43
3.7.3. Mode série.....	44
3.7.3.1 Calcule avec LSB en tête.....	44
3.7.3.2 Calcule avec MSB en tête.....	44
3.7.4. Arithmétique en ligne.....	45
3.7.4.1 Représentation des nombres dans les systèmes redondants.....	45
3.7.4.1.1. Formulation mathématique.....	46
3.7.4.1.2. Facteur de Redondance K	47
3.7.5. Application à une fonction $Z=f(X, Y)$	47
3.7.6. Modélisation de l'algorithme de S.Mallat par l'arithmétique en ligne.....	48
3.7.6.1. Description Mathématique.....	48
3.7.6.2. Calcul de l'intervalle de convergence.....	49
3.7.6.3. Calcul en ligne.....	49
3.7.6.4. Le résidu partiel.....	50

3.7.6.5. Le résidu complet.....	50
3.7.7. Représentation en complément à deux.....	51
3.8. Conclusion	52
CHAPITRE 4 : IMPLEMENTATION SOFTWARE ET HARDWARE DES ONDELETTES ET LES RESEAUX DE NEURONES	54
4.1. Introduction	54
4.2. Méthode proposée : modélisation et conception.....	55
4.2.1. Bloc 1 : partie prétraitement par ondelette	55
4.2.1.1. Le choix des coefficients de filtre.....	56
4.2.2. Bloc 2: La quantification.....	57
4.2.2.1. Détermination des différents paramètres du réseau de Kohonen.....	59
4.2.2.1.1. Test pour 16 neurones et 50 itérations	59
4.2.2.1.2. Test pour 25 neurones et 50 itérations	59
4.2.3. Génération des dictionnaires.....	60
4.2.3.1. Calcul du facteur d'apprentissage $\beta_j(t)$	61
4.2.3.2. Génération du dictionnaire pour la quantification scalaire.....	61
4.2.3.3. Dictionnaire pour la quantification vectorielle.....	62
4.2.4. La décompression ou restitution des images	62
4.2.5. Calcul des performances.....	63
4.3. PARTIE HARDWARE : Architecture globale.....	64
4.3.1. Bloc1 de décomposition par ondelette	64
4.3.1.1. Bloc de normalisation.....	66
4.3.1.1.2. Taille des mémoires.....	66
4.3.1.2. Bloc de sélection.....	67
4.3.1.2.1. Additionneur à propagation de retenue.....	67
4.3.1.2.2. Circuit de sélection.....	68
4.3.2. Bloc 2 : Implémentation du quantificateur vectorielle et du quantificateur scalaire par l'algorithme de Kohonen.....	69
4.3.2.1. Implémentation de la quantificateur scalaire QS.....	69
4.3.2.1.1. Bloc du multiplexage.....	70
4.3.2.1.2. Bloc de calcul de la distance euclidienne	71
4.3.2.1.3. Bloc comparateur.....	72
4.3.2.1.3.1 Module A.....	73

4.3.2.1.3.2. Module B.....	73
4.3.2.1.3.3. Module C.....	74
4.3.2.1.3.4. Module D.....	74
4.3.2.1.4. Bloc de calcul du facteur d'apprentissage β	74
4.3.2.1.4.1. Module B1	76
4.3.2.1.4.2. Sous module (L-M)	77
4.3.2.1.4.3. Sous module MAX.....	78
4.3.2.1.4.4. Module de calcul de Gv.....	79
4.3.2.1.5. Bloc mise à jour	79
4.3.2.2. Implémentation de la quantification vectorielle (QV)	81
4.3.2.2.1. Bloc de calcul de distance II.....	82
4.3.2.2.2 . Bloc comparateur et bloc de calcul du β _facteur d'apprentissage ..	83
4.3.2.2.3. Bloc mémoire (1)	83
4.3.2.2.4. Bloc mise à jour (2)	84
4.3.2.2.5. Bloc mémoire (2).....	85
4.4. Résultats d'implémentation et de simulation sur circuit Virtex II	85
4.4.1. Résultats des différentes images testées sur circuit Virtex II	86
4.5. Résultats d'implémentation et de simulation sur la carte Zedboard	96
4.5.1. Résultats des différentes images testées sur la carte Zedboard	97
4.6. Conclusion	98
Conclusion generale	99
Production scientifique	101
LES APPENDICES	102
A. Les standards	102
B. L'interface graphique de compression Dicom.....	107
C. Les FPGA's.....	111
LES REFERENCES.....	122

LISTE DES ILLUSTRATIONS GRAPHIQUES ET TABLEAUX

Figure 1.1 : niveaux du modèle d'une image	6
Figure 1.2 : Arbre binaire de Huffman.....	9
Figure 1.3 : Exemples de Codage par plage RLE.....	10
Figure 1.4 : Schéma général d'une méthode de compression.....	10
Figure 1.5 : Principe générale de la quantification vectorielle	12
Figure 1.6 : Principe de la quantification vectorielle pour la constitution d'un dictionnaire	13
Figure 1.7 : Exemple de codage et décodage de QV	14
Figure 1.8 : Compromis dans le mode de codage sans perte	21
Figure 1.9 : Compromis dans le mode de codage avec perte.....	21
Figure 2.1 : Vues du cerveau pour différentes modalités d'imagerie.....	26
Figure 3.1 : Schéma de la transformée en ondelettes	33
Figure 3.2 : Principe de l'analyse multi résolution	33
Figure 3.3 : Représentation de la fonction $\psi(t)$ en fonction du facteur d'échelle a	34
Figure 3.4 : Procédé de décomposition pour l'analyse multi résolution	35
Figure 3.5 : Décompositions successives de la transformée en ondelettes.....	36
Figure 3.6 : Les quatre sous-bandes de l'image décomposée.....	36
Figure 3.7 : Schéma de L'algorithmme de décomposition de S.Mallat	37
Figure 3.8 : Schéma de la décimation	38
Figure 3.9 : Processus de reconstruction du signal.....	38
Figure 3.10 : Processus d'interpolation du signal	38
Figure 3.11 : Exemple d'une carte de Kohonen.....	39
Figure 3.12 : Algorithme d'apprentissage de Kohonen.....	41
Figure 3.13 : Deux exemples de dictionnaires.....	41
Figure 3.14 : Divers modes de transmission d'un nombre entier de huit chiffres	44
Figure 3.15 : Calcul en ligne de l'expression $\sqrt{A + B + C \times D}$	45
Figure 4.1 : Schéma bloc de l'architecture proposée.....	55
Figure 4.2 : Représentation des images originales pour le test.....	55
Figure 4.3 : décomposition d'une image en quatre bandes.....	57
Figure 4.4 : Résultat de décomposition des images testes en quatre bandes.....	57
Figure 4.5 : Procédé de quantification de la bande approximation et détails	58

Figure 4.6 : Reconstruction des images pour 16 neurones et 50 itérations.....	59
Figure 4.7 : Reconstruction d'images pour 25 neurones et 50 itérations.....	60
Figure 4.8 : Représentation du dictionnaire pour la quantification scalaire	62
Figure 4.9 : Représentation du dictionnaire pour la quantification vectorielle.....	62
Figure 4.10: Reconstruction des images	62
Figure 4.11: Structure générale du Bloc1	65
Figure 4.12 : Architecture de la cellule de base du BLOC 1.....	65
Figure 4.13 : Utilisation d'une ROM lors du calcul $Lp[j]$	66
Figure 4.14 : Schéma logique du circuit de sélection.....	68
Figure 4.15 : L'architecture globale de la quantification scalaire.....	69
Figure 4.16 : Bloc du multiplexage.....	71
Figure 4.17 : Cellule de base du bloc de calcul de la distance Euclidienne.....	71
Figure 4.18 : Schéma synoptique du bloc de calcul de la distance Euclidienne.....	72
Figure 4.19 : Schéma synoptique du bloc comparateur.....	73
Figure 4.20 : Cellule de base pour les modules du comparateur.....	74
Figure 4.21 : Schéma synoptique du bloc de calcul du facteur d'apprentissage β	75
Figure 4.22 : Schéma du module B1.....	77
Figure 4.23 : Cellule de base du sous module (L-M).....	77
Figure 4.24 : Schéma du sous module (L-M).....	78
Figure 4.25 : Cellule de base du sous module MAX.....	79
Figure 4.26 : Schéma du sous module MAX.....	79
Figure 4.27 : Cellule de base du bloc mise à jour.....	80
Figure 4.28 : Schéma du bloc mise à jour.....	81
Figure.4.29 : L'architecture globale de la quantification vectorielle.....	81
Figure 4.30 : Cellule de base du bloc de calcul de distance (II).....	83
Figure 4.31 : Mémoire élémentaire de 1 ^{er} module.....	83
Figure 4.32 : Schéma synoptique du Bloc mémoire (1).....	84
Figure 4.33 : Cellule de base du bloc mise à jour.....	85
Figure 4.34 : Schéma synoptique du Bloc mémoire (2).....	85
Figure 4.35 : représentation des images test reconstruites en software (par Matlab) et en hardware (par Modelsim).....	96
Figure 4.36 : représentation des images test reconstruites en hardware sur la carte Zedboard	97

LISTE DES TABLEAUX

Tableau 2.1: Propriétés des images médicales pour différentes modalités et différents organes	28
Tableau 2.2 : Application de la Compression JPEG2000 sur différentes images Médicales	30
Tableau 2.3 : Quelques approches de quantification vectorielle appliquées sur des images IRM	30
Tableau 3.1 : Nombre d'opérations nécessaires aux différents types de transformées sur n échantillons	39
Tableau 3.2: type de facteur de redondance	47
Tableau 3.3: Notations utilisées.....	47
Tableau 3.4 : le codage en complément à deux	52
Tableau 4.1: Coefficients du filtre de LE Gall.....	56
Tableau 4.2: Coefficients du filtre après simplification	56
Tableau 4.3 : Test pour 16 neurones et 50 itérations.....	59
Tableau 4.4: Test pour 25 neurones et 50 itérations.....	59
Tableau 4.5 : Calcul de performance des images DICOM.....	63
Tableau 4.6: Taille des mémoires pour chaque filtre.....	67
Tableau 4.7: taille des résidus partiels $L_p[j]$	67
Tableau 4.8 : Les Bits de sélection.....	68
Tableau 4.9 : Fonction de sélection.....	68
Tableau 4.10 : Table de vérité du multiplexeur.....	70
Tableau 4.11: Les différentes constantes des poids synoptiques initiaux.....	70
Tableau 4.12 : Les différentes valeurs pré-calculées	75
Tableau 4.13 : Les différentes valeurs pré-calculées de L_j et M_j	76
Tableau 4.14: Les performances en taux d'occupation sur FPGA.Circuit: 2v8000ff1152.5	86
Tableau 4.15 : Les performances fréquentiels sur FPGA. Circuit: 2v8000ff1152-5	86

LISTE DES ACRONYME

BD	Bit Depth
CAR	L' Association Canadienne des Radiologistes
CLB	Configurables Logique Blocs
COMP	comparateur
DCM	Digital Clock Manager
DCT	Discret Cosine Transform
DICOM	Digital Imaging and Communications in Medecine
DWT	Discret Wavlete Transform
EQM	Erreur Quadratique moyenne
FFT	Fast Fourier Ttransform
FPGA	Field Programmable Gate Array
GLA	Generalized Lloyd Algorithm
HL7	Health Level Seven
HVS	le système visuel humain
IOB	Input Output Blocs
IRM	Imagerie par Résonance Magnétique
ISO	International Standards Organisation
JPEG	Joint Photographic Experts Group
JPEG LS	Joint Photographic Experts Group Lossless
KLT	Transformation de Karhunen-loeve
LBG	Linde-Buzo-Gray
LUT	Look Up Table
MIPS	Millions of Instruction Per Second
MSB	Most Significant Bit
MSE	Mean Square Error
MUX	multiplexeurs
ORL	Oto-rhino-laryngologie
PACS	Picture Archiving and Communication System
PSNR	Peak Signal to Noise Ratio
PTSVM	pruned tree-structured VQ
QS	Quantification scalaire
QV	Quantification Vectorielle
RC	Le rapport de compression
RLE	Run length Encoding
ROI	Region of interest
SNR	Signal-to-Noise Ratio
SOM	Self Organising Map
SQ	Scalar Quantization
SSIM	Structural SIMilarity
SVQ	scalar-vector quantizer
TDM	Tomodensitométrie
TEP	Tomographie par émission de positions
VHDL	Very high Speed integrated circuits Hardware Description Langage
VLTSVM	variable-length tree-structured VQ

INTRODUCTION GENERALE

Le domaine du traitement de l'image connaît une progression importante liée à l'évolution des technologies de l'information et de la communication. De nos jours, le traitement numérique de l'information s'impose dans toutes les disciplines comme un élément incontournable, offrant des services aussi nombreux que variés ; facilitant grandement notre vie quotidienne. En parallèle à l'évolution de la puissance des systèmes d'information, nous assistons à une course à l'augmentation des bandes passantes pour accéder aux grandes masses de données à travers des services et usages innovants. Les problématiques se posent en termes de contenus multimédias, de supports et protocoles de transmission des services et usages, de partage d'information et de sécurité. Dans ce cadre, la compression de données constitue un passage obligé à la fois pour la transmission et l'archivage.

Le monde de la médecine a parallèlement évolué en s'adaptant aux changements apportés par le numérique et s'est vu doté d'une multitude de services qui, d'un côté, ont facilité le travail du personnel médical (Gestion de bases de données, Systèmes d'information...) et d'un autre côté, ont offert de nouveaux outils de diagnostic aux médecins ; grâce à l'imagerie médicale. L'imagerie médicale est un domaine en plein essor, du fait du développement des technologies numériques. Elle permet une investigation plus fine des organes humains grâce à la mise à disposition de systèmes de radiologie de plus en plus performants. La contrepartie réside dans une quantité considérable de données générée qui peut rapidement saturer les systèmes conventionnels de transmission et de stockage. A titre d'exemple, un fichier ne comprenant que les données relatives aux images d'une séquence spatiale IRM de douze coupes est de l'ordre de trois Moctets par exemple [1]. Les études ont prouvé que le service de radiologie d'un grand hôpital peut produire plus de 20 Térabits de données d'images par année [2]. Ceci pose de sérieux problèmes pour les services d'imagerie des hôpitaux. Par ailleurs, ces données doivent être conservées un certain nombre d'années pour répondre aux contraintes réglementaires actuelles. L'augmentation croissante et continue des capacités de stockage apporte une réponse partielle à ce problème mais demeure la plupart du temps insuffisante.

Actuellement, la compression de données dans un service de radiologie est toujours effectuée. Elle est réalisée par des standards comme JPEG LS ou JPEG 2000 ; dont la syntaxe est prévue dans le standard de format d'images médicales DICOM (Digital Imaging and COmmunications in Medecine). Ce type de compression avec une reconstruction exacte de l'image de départ, garantissant l'intégrité des données, demeure la plus prônée des praticiens pour des raisons évidentes de diagnostic [3], [4]. Cependant, elle offre de faibles performances. L'origine de la préférence des médecins pour la compression sans perte par rapport à la compression avec pertes est d'éviter les erreurs médicales liées à une mauvaise reconstruction de l'image. En effet, le principal problème de la compression avec pertes pour les images médicales est dû au fait que des détails importants au niveau diagnostique pourraient disparaître, d'autres pourraient éventuellement apparaître. Ces détails sont généralement des structures difficiles à discerner, car elles entraînent de faibles changements de contraste. Ainsi par exemple, des images peuvent révéler des lésions à travers des détails potentiellement sensibles à la compression avec perte puisqu'ils sont petits et ont des bords faiblement définis. Notons que les erreurs liées à l'utilisation éventuelle de la compression avec pertes pourraient entraîner pour le médecin des problèmes réglementaires et juridiques importants. Il faudrait en effet prouver qu'une erreur de diagnostic ne vient pas d'un problème de la qualité de l'image reconstruite [5].

Cependant, les spécialistes s'accordent à dire qu'une véritable augmentation des performances ne peut être atteinte qu'en abordant une autre alternative : abandonner la généralité, exploiter la spécificité. On ne s'intéresse plus à créer un système qui est capable de compresser n'importe quel type d'image, mais on développe plutôt des algorithmes et des systèmes qui sont très puissants dans une certaine application. Outre la volonté d'augmenter le taux de compression, l'autre but des recherches est de définir un système "intelligent" de compression, en ajoutant de nouvelles fonctionnalités.

Souvent, les images brutes ne peuvent être étudiées telles quelles se présentent et doivent subir une étape de transformation nécessaire à l'augmentation de l'efficacité du traitement. Cette étape de transformation consistera notamment, à extraire l'information nécessaire à représenter l'image de manière plus compacte pour la compression.

Le domaine du traitement de l'image et du signal connaît une progression importante. Il fait appel à un ensemble de théories et de méthodes qui constituent des composantes fondamentales du traitement du signal ; de nombreuses méthodes et transformées pour l'analyse existent. Une technique qui a suscité une attention particulière et a acquis une reconnaissance grandissante dans la communauté scientifique, la transformée

en ondelette qui est considérée comme une évolution de la transformée de Fourier. De par ses qualités associées à l'analyse multi résolution, elle permet l'observation et l'étude des objets présents dans l'image à différentes échelles. Citons par exemple l'algorithme de Stéphane Mallat qui présente un caractère séquentiel, qui réduit considérablement les performances globales. De ce fait, l'amélioration des performances de son exécution constitue une préoccupation majeure dans le contexte d'un traitement massif de l'information.

Ainsi, le problème consiste à accélérer le processus de calcul de ces opérateurs sériels et à améliorer leurs performances en utilisant des architectures et des arithmétiques adéquates. Le développement d'algorithmes fondés sur l'arithmétique en ligne s'avère très efficace pour le calcul d'une chaîne d'opérations arithmétiques sérielles et permet de résoudre le problème de rapidité et d'engendrer des architectures pour une implémentation hardware, vu les nombreux avantages qu'elle procure du fait de la circulation des opérandes de manière sérielle le bit le plus fort en tête-MSB- (Most Significant Bit). Ce qui favorise un traitement massif de l'information en utilisant le pipeline au niveau du bit et une meilleure précision, en particulier pour les opérations non parallélisables.

Ce travail se consacre à l'implémentation d'un système hybride ondelettes et réseaux de neurones (WT-SOM) pour la compression des images médicales sur circuit FPGA VirtexII. Les images médicales présentent des caractéristiques spécifiques qui demandent à être exploitées par un algorithme de compression spécifique et efficace.

Nous montrerons que pour les images médicales et la modélisation de l'algorithme de S. Mallat par l'arithmétique en ligne détaillée en [6] et [7] et la réalisation de la quantification vectorielle QV par carte SOM (Self Organising Map) est efficace. La QV permet de créer un dictionnaire, au niveau bloc, par une approche neuronale, outil très utilisé pour la compression avec perte [8] et les données de grande dimension.

La Quantification Vectorielle constitue une étape cruciale dans la compression des images. La complexité du codage en QV est probablement sa plus grande faiblesse ; pour cette raison, on lui préfère souvent la quantification scalaire, en particulier dans le cadre de normes telles que JPEG, JPEG 2000. En réduisant cette complexité, sans dégrader la qualité, on pourrait redonner à la QV une place plus grande dans la panoplie des méthodes utilisées pour la compression des images afin d'améliorer les performances de son implémentation sur circuit FPGA. C'est actuellement un domaine de recherche très actif et la mise en œuvre d'un réseau de neurones sur FPGA avec un grand nombre de neurones reste une tâche difficile et coûteuse [9] et [10].

Par ailleurs, les algorithmes de traitement de signal sont basés sur des calculs répétitifs et utilisent un nombre réduit d'opérations simples d'un microprocesseur complet, donc ne nécessitant pas toutes les capacités disponibles sur une machine universelle. Ainsi, un circuit qui effectue les tâches simples et répétitives d'un algorithme de traitement de signal sera moins complexe et plus rapide qu'un microprocesseur. Ce sont toutes ces considérations qui ont guidé ce travail de recherche, afin de proposer une méthode de compression pour des applications spécifiques.

Cette thèse s'articule autour de cinq chapitres : Dans le premier chapitre, nous donnerons une introduction sommaire sur l'état de l'art des méthodes de compression. Le deuxième chapitre fera l'objet d'une étude sur la particularité des images médicales et leurs compressions. Nous aborderons dans le troisième chapitre les transformées utilisées dans ce travail, telle que la transformée en ondelettes qui constitue une des techniques la plus représentée dans le domaine de l'analyse et la synthèse des signaux, ainsi que les réseaux de neurones en particulier l'algorithme de Kohonen.

La modélisation mathématique de l'algorithme de Stéphane Mallat par l'arithmétique en ligne et ces particularités sera définie dans le quatrième chapitre.

La conception et l'implémentation de notre architecture sur FPGA de Xilinx ainsi que les résultats des simulations obtenus feront l'objet du cinquième chapitre. Dans le dernier chapitre, nous donnerons une conclusion et quelques perspectives sur l'amélioration des travaux que nous avons abordés tout au long de cette thèse.

CHAPITRE 1

ETAT DE L'ART DE LA COMPRESSION

1.1. Introduction

La compression est une technique qui modifie la représentation initiale des données pour qu'elles occupent moins de place sur une unité de stockage et moins de temps lors d'une transmission. Cela est possible grâce à une redondance de l'information généralement présente dans une image. Cette redondance peut être d'origine statistique, spatiale ou encore fréquentielle. Une caractéristique très importante de la compression d'images est qu'elle peut être avec pertes. Du fait qu'une image, est visualisée par des humains donc on peut se permettre de négliger quelques informations imperceptibles par le système visuel humain (HVS).

Nous allons dans ce chapitre exposer quelques notions de base sur la compression des images, et aussi les critères de performance et de mesure de qualité de l'image compressée. Nous présentons par la suite, les méthodes les plus utilisées pour la compression avec et sans pertes.

1.2. Natures des informations la redondance

Une image numérique présente la particularité de posséder des corrélations importantes entre les pixels voisins. Cette corrélation est vue comme une redondance des informations pertinentes. La redondance peut être de deux natures : la redondance spatiale qui apparaît directement entre les pixels voisins de l'image originale et la redondance spectrale qui est liée aux fréquences et qui est acquise avec les transformations de domaines. Il y a différentes manières d'approcher les images pour les compresser

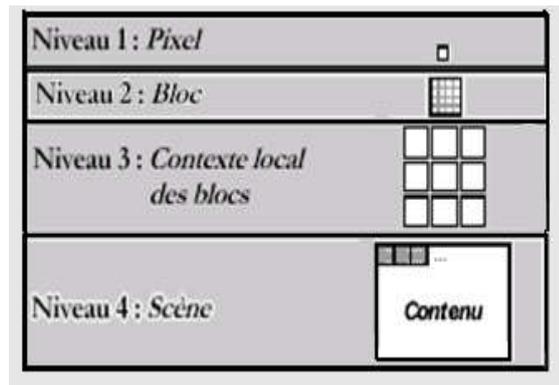


Figure 1.1 : niveaux du modèle d'une image

Quatre niveaux ou couches se différencient figure 1.1, le niveau le plus bas est celui du niveau scène ou le contenu de l'image, ensuite le niveau bloc, puis le contexte local des blocs. Le niveau le plus haut est du pixel.

1.2.1. Niveau pixel

Quelques approches considèrent l'image en tant qu'un ensemble de pixels. Les pixels sont pris un par un par le codeur, la valeur de chaque pixel qui est codée et transmise au décodeur. Cependant, la performance est très limitée en traitant les pixels indépendamment. Pour exploiter leurs dépendances, les pixels sont souvent considérés dans un certain voisinage, par rapport aux pixels adjacents.

1.2.2. Niveau bloc

Le niveau bloc est appliqué principalement par les algorithmes irréversibles, qui consiste à regrouper les pixels en blocs, en général, les pixels ne se recouvrent pas, et l'ensemble de ces blocs couvre l'image. L'unité du codage devient ainsi le bloc : ce ne sont pas les valeurs de pixel qui sont codées, mais toutes les opérations s'effectuent au niveau des blocs. Le passage à ce niveau supérieur est très important, puisqu'il y a des possibilités plus larges pour interpréter un bloc que des pixels individuels. En exploitant les corrélations intra-bloc, en se basant sur la similitude entre les imagerie, les blocs peuvent être décrits de façon très compacte, un bloc permet également d'extraire et d'exploiter des informations de plus haut niveau.

1.2.3. Niveau contexte local des blocs

Comme les pixels, les blocs peuvent aussi être traités soit individuellement, ou bien par rapport aux blocs adjacents. L'exploitation des dépendances inter-blocs nécessite des techniques plus complexes.

1.2.4. Niveau scène

L'ensemble des pixels constitue l'image complète, les motifs des blocs forment le contenu de la scène d'image. Elles permettent ainsi de trouver, de développer les meilleures techniques de compression et de les adapter à ce type de données. La distinction de ces quatre niveaux ne signifie évidemment pas qu'une méthode se place entièrement à un seul niveau. Le premier niveau est toujours concerné, puisque l'étape finale de la décompression est la restitution des valeurs de pixels. Cependant, le niveau pixel est rapidement dépassé dans les approches de codage en blocs. Le troisième niveau est une extension du niveau bloc, qui tient compte des régularités entre les blocs voisins. Il y a donc certaines relations, interactions entre les couches du modèle. Le niveau de la scène peut aussi donner des indications au deuxième niveau, et vice versa : l'analyse d'un bloc indépendant peut fournir certaines informations concernant le contenu de l'image.

1.3. Les méthodes de compression

Toutes les techniques de compression d'images essayent en général de tirer partie de cette redondance. On différencie généralement les méthodes de compression selon la distorsion ou la perte d'informations qui est engendrée, comme suit :

- Les méthodes sans pertes « sans distorsion ou réversible ».
- Les méthodes avec pertes « avec distorsion ou non réversible ».

1.3.1 Les méthodes sans pertes « sans distorsion ou réversible »

La compression sans pertes permet de retrouver, après reconstruction, la valeur exacte du signal compressé ; la même information est réécrite d'une manière plus concise, sans modifier le contenu. Le processus de codage sans pertes crée des "mots-codes" à partir d'un dictionnaire. Ces processus s'appuient sur des informations statistiques de l'image. Les codes statistiques les plus répandus sont le codage de Huffman et le codage arithmétique. Les codeurs statistiques permettent de s'approcher au mieux de l'entropie.

Ils ont pour principe d'associer aux valeurs les plus probables les mots binaires les plus courts. Un codeur entropique permet de compresser une séquence de symboles en se basant sur leur probabilité d'apparition. Chaque symbole se voit assigner une nouvelle représentation dépendant de sa probabilité d'apparition de manière est ce que le codage de la source s'approche au plus de son entropie. L'entropie est une grandeur qui caractérise la quantité d'information que contient une image. Par exemple, une image dont tous les pixels ont la même valeur contient très peu d'informations car elle est extrêmement redondante, donc son entropie est faible. En revanche une image dont tous les pixels ont une valeur aléatoire contient beaucoup d'informations, son entropie est forte. L'entropie (en bits) est calculée par la formule suivante :

$$H = - \sum_{i=1}^N P_i \log_2 (P_i) \quad 1.1$$

Où P_i est la probabilité de présence du niveau de gris i , et N étant le nombre du niveau de quantification ($N = 256$ pour une image à niveau de gris codée sur 8 bits).

Dans ce qui suit, nous allons présenter les trois méthodes de compression sans pertes les plus utilisées. Plus de méthodes sont détaillées dans [11]. [12].

1.3.1.1 Codage de Huffman

Le codage Huffman [13] ; [14] est une méthode statistique basée sur l'attribution d'un mot de code binaire pour chaque symbole de la chaîne à compresser. La longueur des mots du code des symboles est variable. Les symboles ayant la probabilité d'apparition forte sont codés avec des chaînes de bits plus courtes, tandis que les symboles dont la probabilité d'apparition est faible sont codés par des chaînes plus longues. Le codeur de Huffman est un arbre binaire ordonné par tous les symboles et par leurs fréquences d'apparition. Pour représenter le codage de Huffman, on utilise l'arbre binaire. Soit un message à coder "ABBBBAAC". La fréquence d'apparition ainsi que le code Huffman correspondant est donné dans le tableau suivant et représentés par la figure 1.2.

Symbole	Fréquence d'apparition	Code Huffman
A	3	01
B	4	1
C	1	00

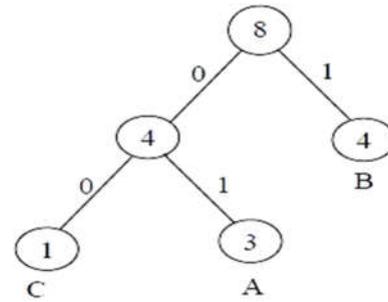


Figure 1.2 : Arbre binaire de Huffman

1.3.1.2. Codage arithmétique

Le codage arithmétique [15] se singularise par sa capacité à coder chaque symbole sur un nombre non entier de bits. En réalité, il n'assigne pas un mot de code à chaque symbole mais il associe un point de l'intervalle $[0,1]$ à un ensemble de symboles. Le principe repose sur le découpage de l'intervalle $[0,1]$. Chaque symbole se voit attribuer une partition de l'intervalle dont la taille est égale à sa probabilité d'occurrence. L'ordre de rangement est mémorisé pour être utilisé lors du décodage.

Le codage arithmétique est généralement plus performant que le codage de Huffman. Cependant il est gourmand en ressources et nécessite de connaître à priori l'intégralité du signal avant de pouvoir procéder au codage.

1.3.1.3. Le Codage par plage (Run length Encoding : RLE)

Le codage RLE permet simplement de remplacer des éléments significatifs, successifs et identiques par un seul d'entre eux, précédé du nombre de répétitions. Le nombre de répétitions doit être supérieur à trois. Si le nombre de répétitions est inférieur à trois, la séquence est considérée comme étant sans répétitions et est codée comme tel. Lors du codage, une plage sans répétitions est laissée intacte et est précédée d'un zéro suivi de sa longueur. Si la longueur est impaire, un zéro est rajouté à la fin de la séquence ; exemple figure 1.3.

10 10 10 10 10	est codé 05 10
0A 0A 0A 0A 0A 0A 0A 0A 0A	est codé 0A 0A
23 65 55 34 22	est codé 00 05 23 65 34 22 00
10 10 89 23	est codé 00 04 10 10 89 23

Figure 1.3 : Exemples de Codage par plage RLE

Ce procédé apporterait une compression dérisoire si on cherche à l'appliquer à un texte puisque les répétitions triples sont très rare. En revanche, si on l'applique à une image, il est aisé d'apercevoir que les plages de couleurs (ou niveaux de gris) homogènes sont souvent importantes, même si leur nombre est parfois faible, donnant un bon taux de compression.

1.3.2. Les méthodes avec pertes « avec distorsion ou non réversible ».

Les méthodes irréversibles sont des méthodes qui tirent parti d'une redondance présente dans l'image, en même temps elles exploitent le fait que l'œil n'est pas sensible à certaines dégradations. La quantification est un des mécanismes utilisé dans les algorithmes de compression, qui produit des pertes d'information [16]. Afin de mieux compresser l'information, la source est transformée en groupe de coefficients.

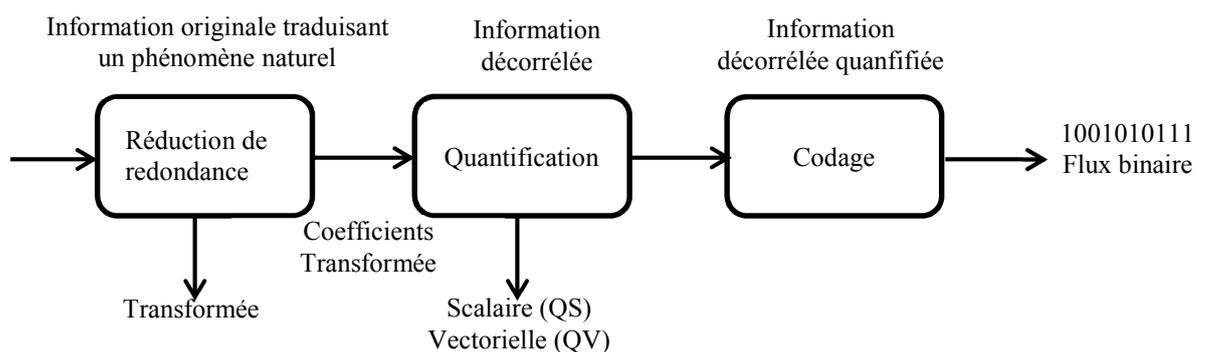


Figure.1.4 : Schéma général d'une méthode de compression

La chaîne de compression avec perte la plus répandue est représentée à la figure 1.4. Elle comporte trois étapes essentielles : une étape de transformation des données de façon à les

représenter de manière plus compacte suivie d'une étape de quantification (où se produisent les pertes) et d'une étape de codage pour éliminer les dernières redondances. On distingue trois approches selon le contenu de la première étape :

- l'approche fréquentielle, telle qu'elle est utilisée dans JPEG à travers la transformée en cosinus discrète (DCT) [17].
- l'approche spatio-fréquentielle, retenue dans JPEG2000, basée en particulier sur la transformée en ondelettes (DWT) [18], [19] dont l'efficacité a été prouvée pour coder des signaux non stationnaires tels que les images [20], [21] ;
- l'approche par fractales [22] - [24] ou plus récemment par ondelettes géométriques [25] - [27], visant à améliorer la prise en compte de l'information géométrique des contours dans l'image.

1.3.2.1 Codage par quantification

La quantification fait partie de plusieurs méthodes de compression d'images [28] [29]. L'objectif est de réduire la taille des coefficients de façon que cette réduction n'apporte pas de dégradations visuelles à l'image. L'information décorrélée peut prendre des valeurs entières, réelles, complexes ou vectorielles.

On distingue principalement deux types de quantification : quantification scalaire (QS) et quantification vectorielle (QV). La première est appliquée sur des scalaires comme l'intensité d'un pixel ou la valeur d'un coefficient tandis que la seconde est appliquée sur des blocs de pixels ou coefficients voisins. C'est dans cette étape de quantification que la perte d'information est introduite donc une dégradation de la qualité de restitution est introduite.

1.3.2.1.1. La quantification scalaire

La quantification scalaire (Scalar Quantization : SQ) est réalisée indépendamment pour chaque élément. D'une manière générale, on la définit comme étant l'association de chaque valeur réelle x , à une autre valeur q qui appartient à un ensemble fini de valeurs, [30], [31].

1.3.2.1.2. La quantification vectorielle

La quantification vectorielle - (Vector Quantization : VQ) a été développée par Gersho et Gray [31] [32] ; et elle fait aujourd'hui l'objet de nombreuses publications dans le domaine de la compression numérique [33] - [38]. Le principe de la quantification vectorielle est issu des travaux de Shannon qui montre qu'il était toujours possible d'améliorer la compression de données en codant non pas des scalaires, mais des vecteurs. Un quantificateur vectoriel Q associe à chaque vecteur d'entrée $X_i = (x_j, j = 1 \dots k)$ un vecteur $Y_i = (y_j, j = 1 \dots k) = Q(X_i)$, ce vecteur Y_i étant choisi parmi un dictionnaire (codebook) de taille finie (figure 1.5) . La VQ produit de meilleurs résultats que la SQ, néanmoins ; la VQ nécessite un codage complexe et de grandes capacités de mémoire.

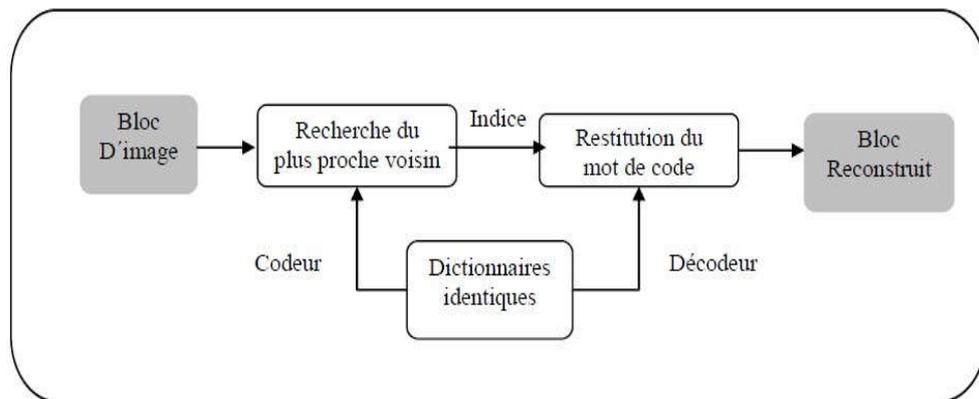


Figure 1.5 : Principe générale de la quantification vectorielle

1.3.2.2. Principe de la quantification

Le principe de la quantification vectorielle figure 1.5 est de définir un dictionnaire (codebook) de taille finie, cela a donné lieu à de nombreux algorithmes de quantification.

1.3.2.3. Le dictionnaire (Codebook)

Le succès d'un quantificateur dépend principalement de son dictionnaire. Si ce dernier est bien représentatif de l'espace à quantifier, la reconstruction sera de bonne qualité.

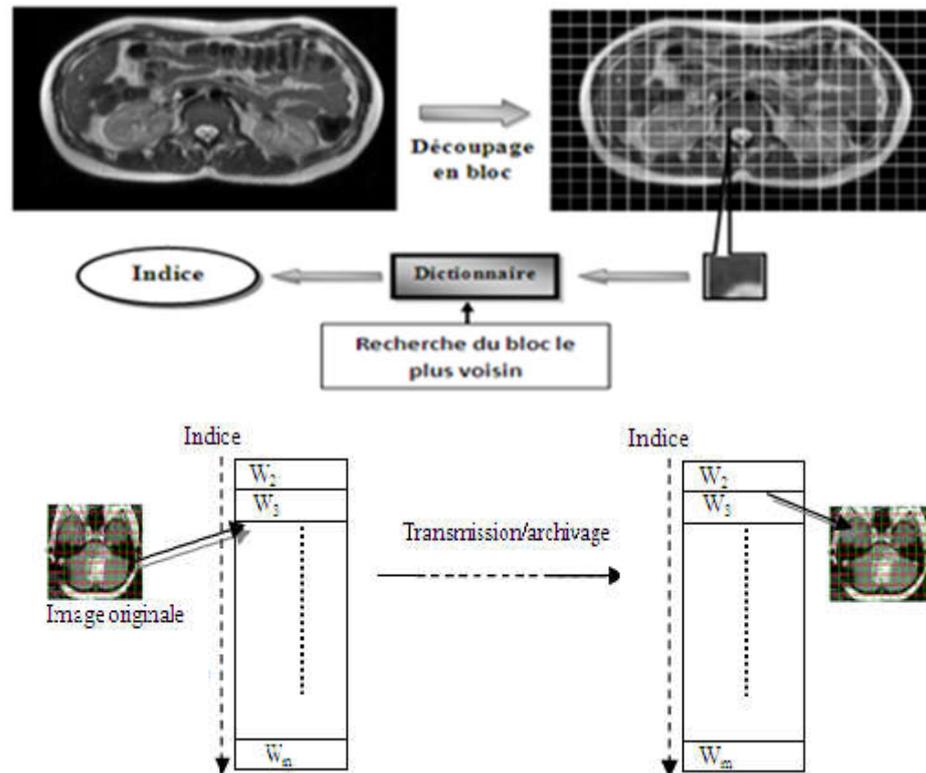


Figure 1.6. Principe de la quantification vectorielle pour la constitution d'un dictionnaire

Le principe de la quantification est de réduire la taille des coefficients de façon que cette réduction n'apporte pas de dégradation visuelles à l'image est de définir un dictionnaire de taille fini

1.3.2.4. Construction du dictionnaire

On ne dispose en général pas de la loi de probabilité du signal, c'est pourquoi les algorithmes travaillent à partir d'une base d'apprentissage, ensemble de vecteurs typiquement issus de plusieurs images représentatives des images qui devront être codées par la suite.

Un quantificateur vectoriel Q de dimension k et de taille N est une application de \mathbb{R}^k , dans un ensemble fini C contenant N points de reproduction, appelés mots-codes ou prototypes [39]. Les mots-codes appartiennent à \mathbb{R}^k . L'ensemble C des prototypes est appelé dictionnaire. Ainsi, $Q : \mathbb{R}^k \rightarrow C$, $C = \{y_1, y_2, y_3 \dots y_N\} \subset \mathbb{R}^k$.

Généralement, Q est défini par la règle du plus proche voisin $Q(x) = y_{i^*} / \forall i d(x, y_{i^*}) \leq d(x, y_i)$. C'est l'indice i^* qui est stocké ou transmis prenant ainsi moins de place que le vecteur x . L'opération de décompression est appelée reconstruction. Elle consiste en une simple table de scrutation qui associe à l'indice reçu i^* le prototype correspondant y_{i^*} .

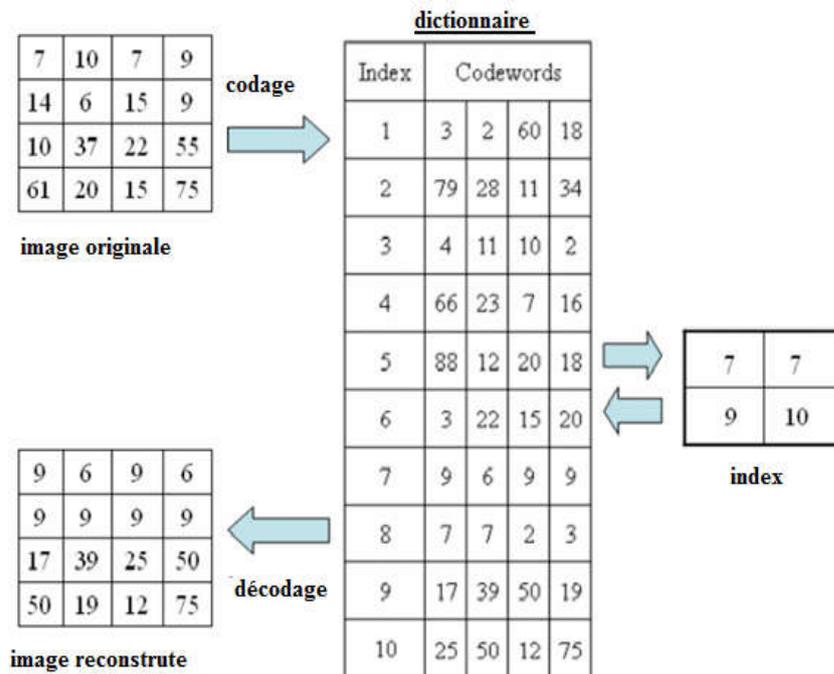


Figure 1.7. Exemple de codage et décodage de QV [40].

1.4. Algorithmes de construction du dictionnaire

Il existe deux types d'algorithmes de construction du dictionnaire [41] :

- Les algorithmes non neuronaux.
- Les algorithmes neuronaux.

De plus, chaque type contient deux classes :

- Hors-ligne (off-line) : Le dictionnaire n'est modifié qu'après passage complet de la base d'apprentissage.
- En-ligne (on-line) : Un ou plusieurs mots codes sont modifiés après passage de chaque vecteur de la base d'apprentissage.

Dans ce qui suit, nous décrirons d'abord brièvement un ensemble d'algorithmes. Nous détaillerons dans le chapitre trois l'algorithme que nous avons utilisé dans notre application : l'algorithme de Kohonen ou carte auto-organisatrice (Self Organizing Map : SOM).

1.4.1. Les algorithmes non-neuronaux

1.4.1.1. Algorithme de Lloyd Généralisé (Generalized Lloyd Algorithm : GLA)

L'algorithme GLA établi par Lloyd en 1957 est un algorithme pour la création des dictionnaires. Il procède par améliorations successives d'un dictionnaire initial suivant deux conditions d'optimalité nécessaires et suffisantes pour minimiser la distorsion moyenne :

- La condition du plus proche voisin : A chaque vecteur d'apprentissage est affecté le mot code le plus proche suivant la distance choisie (Euclidienne, Manhattan...).
- La condition des centroïdes : L'espace d'entrée est partitionné en régions chacune d'elles ayant un mot code comme centroïde. On initialise d'abord un dictionnaire de N vecteurs. On partitionne ensuite la base d'apprentissage en N régions suivant la première condition. On crée ensuite un nouveau dictionnaire dont les éléments sont les centroïdes des régions créées dans l'étape précédente. L'inconvénient de cette méthode réside dans le choix du dictionnaire initial. Si le dictionnaire initial est mal choisi, le dictionnaire final risque de ne pas être représentatif de tout l'espace d'entrée.

1.4.1.2. Algorithme de Linde-Buzo-Gray (LBG)

L'algorithme LBG [42] est introduit en 1980 par Buzo-Gray est similaire au précédent et remédie efficacement au problème d'initialisation du dictionnaire. On commence d'abord par un dictionnaire contenant un seul mot code qui est le centroïde de la base d'apprentissage, on introduit un vecteur aléatoire (perturbateur) de faible amplitude qu'on ajoute et qu'on retranche au mot code initial pour obtenir deux mots codes. On applique ensuite le GLA au dictionnaire de deux vecteurs que nous avons obtenu dans l'étape précédente. On divise chacun des deux mots codes en deux avec le vecteur perturbateur. On applique le GLA aux quatre vecteurs obtenus. On continue ainsi jusqu'à obtenir le nombre de mots codes désiré.

1.4.1.3. Agrégation constructive progressive

Cet algorithme est d'écrit par Akrouf et al a pour objectif d'accroître la rapidité par le parcours de la base d'apprentissage en une seule passe. Pour cela, les mots codes et les régions correspondantes sont définis progressivement en parcourant la base d'apprentissage. A chaque itération, un vecteur d'apprentissage est affecté à un mot code qui est le centroïde d'une région créée autour de lui. Ce mot code est recalculé lorsqu'un nouveau vecteur d'apprentissage lui est affecté.

Au début, le dictionnaire ne contient qu'un seul mot code. Au cours de l'algorithme, un nouveau mot code est créé quand la distorsion entre un vecteur d'apprentissage et son mot code plus proche voisin est supérieure à un seuil.

1.4.1.4. Quantification Vectorielle adaptative par ré-apprentissage

Une manière d'améliorer la qualité de reconstruction consiste à faire évoluer le dictionnaire au cours du temps par un apprentissage sur des blocs d'un certain nombre de vecteurs successifs appelés contextes. Comme le dictionnaire initial ou celui du contexte précédent est déjà assez proche de l'optimum, cet apprentissage est généralement plus court qu'un apprentissage complet à partir de zéro. Le changement de dictionnaire peut porter sur tous les mots codes, ou seulement sur quelques-uns. Par exemple, dans l'algorithme de Gersho ; seuls deux vecteurs codes sont remplacés à chaque changement.

1.4.2. Les algorithmes neuronaux

Les algorithmes neuronaux fonctionnant avec un apprentissage dit compétitif, ils sont particulièrement bien adaptés aux problèmes de construction du dictionnaire. Ils sont tous de type en ligne.

Dans le domaine des réseaux de neurones, on distingue deux catégories de réseaux :

- Réseaux avec apprentissage supervisé : Leurs sorties doivent donner un résultat dont les informations sont données comme entrée du réseau.
- Réseaux avec apprentissage non supervisé : Ils se forment une représentation de l'espace d'entrée en observant seulement les vecteurs de la base d'apprentissage sans avoir, à priori, d'information sur la sortie du réseau.

1.4.2.1. Compétition simple

On initialise d'abord un dictionnaire. On tire un vecteur de la base d'apprentissage et on recherche le mot code correspondant. On le met à jour en ajoutant la distance calculée dans l'étape précédente multipliée par un coefficient d'apprentissage obtenu à partir d'une fonction décroissante au cours du temps et dont les sorties doivent toujours être comprises entre 0 et 1. On s'arrête après un certain nombre d'itérations prédéfini ou bien si la condition d'arrêt est vérifiée.

1.4.2.2. Compétition sensible à la fréquence

Algorithme similaire au précédent, où on ajoute une fonction de mémoire retenant la fréquence de tirage, appelée aussi conscience, des mots codes pour la mise à jour. Cette fonction, qui est croissante en fonction de la fréquence de tirage, est multipliée par la distance entre le vecteur de la base d'apprentissage choisi et le mot code correspondant. Les mots codes situés initialement dans une zone de forte densité de probabilité deviennent moins éligibles au cours du temps alors que les autres, situés dans des zones de plus faible densité de probabilité, seront peu choisis au début de l'algorithme et beaucoup plus éligible à la fin.

1.4.2.3. Carte auto-organisatrice (Carte de Kohonen)

Mis au point par Teuvo Kohonen [43], cet algorithme est basé sur un apprentissage non supervisé. Il donne en sortie une carte organisée adaptée à la topologie de l'espace d'entrée. C'est un réseau de neurones à une seule couche où les neurones sont organisés sur une matrice $N \times M$. Chaque mot code du dictionnaire est représenté par un neurone de la carte : les valeurs des poids synaptiques sont donc les valeurs des mots codes. Après initialisation aléatoire, les neurones se déploient dans l'espace de sortie en respectant la topologie de l'espace d'entrée. Cet algorithme est utilisable en quantification scalaire comme en quantification vectorielle (partie détaillée en chapitre 3).

1.5. Codage par transformation

Les méthodes de compression par transformation n'agissent pas directement sur l'image numérique dans sa représentation originale, mais dans le domaine de la transformée. Cette transformation pouvant être linéaire ou non. Il est bien connu qu'une transformation permet de mettre en évidence certaines propriétés de l'image que la représentation originale ne laisse pas apparaître. En partant d'un ensemble de valeurs numériques corrélées d'une image, le but est d'obtenir un autre ensemble de valeurs le moins corrélées possible dans l'espace transformée. En général, les schémas de codage par transformation subdivisent l'image de taille $N \times N$ en sous images de taille plus petites avant de faire subir à ces sous images une transformation. La transformation consiste en la décomposition de l'image dans une base adéquate de fonctions tels que les coefficients de la transformation soient indépendants et qu'un nombre minimum de ces coefficients contienne une proportion importante de l'énergie de l'image. Ainsi, on pourra mettre à zéro certains d'entre eux sans

nuire de manière significative ni à la quantité d'énergie, ni à l'aspect visuel de l'image reconstruite.

Une transformation adéquate pour la compression d'image devrait permettre la décorrélation des coefficients transformés, la conservation d'énergie ou sa condensation dans un nombre minimum de coefficients. Les transformations linéaires sont les plus utilisées car ayant des expressions analytiques simples et s'implémentant rapidement.

1.5.1. Transformation de Karhunen-loeve (KLT)

Cette transformée a été développée par Karhunen et Loève [44], C'est une série d'analyse spectrale pour la représentation d'un signal aléatoire donné, dont les fonctions de base orthogonales sont obtenues par les vecteurs propres de la matrice d'auto-corrélation correspondante. Cette transformée est optimale au sens où tous les coefficients obtenus sont décorrélés et que la quasi-totalité de l'énergie est conservée par un minimum de coefficients. Mais comme la transformée KLT ne possède pas d'algorithme rapide. On préfère des transformations qui sont indépendantes des images et qui ont des algorithmes rapides, tels que les transformations en ondelettes.

1.5.2. Transformations spectrales ou sinusoïdales

La transformation de Fourier et celles qui s'en déduisent, telles la transformation en sinus, la transformation en cosinus, sont très utilisées en analyse et en filtrage du signal.

Ces transformations possèdent des algorithmes rapides comme la FFT (Fast Fourier Transform) et ses variantes. La variable de l'espace transformé étant la fréquence, une telle décomposition permet de mieux observer la répartition fréquentielle de l'image. Etant donné que ce sont les premiers harmoniques qui contiennent la quasi-totalité de l'énergie, il est donc possible de mettre à zéro une proportion importante des coefficients et de coder l'image rapidement. Malgré la rapidité de la transformation de Fourier, elle décompose l'image en une partie réelle et une partie imaginaire pouvant se convertir en module et argument ce qui n'est pas facile à manipuler ou à interpréter. Les traitements de ces données peuvent s'avérer lourds, d'où la préférence accordée à la transformation en cosinus qui bénéficie de toutes les caractéristiques de la FFT. La transformée en cosinus discrète DCT (Discret Cosine Transform) [45] a été choisie comme standard par JPEG (Joint Photographic Experts Group) pour le codage d'images fixes et a fait l'objet de beaucoup d'études et d'applications de la compression dans tous les domaines de l'imagerie, y compris le

médical. Contrairement à la transformation KLT, la matrice de transformation DCT est complètement indépendante de l'image.

D'autre part, la norme JPEG présente un certain nombre d'inconvénients :

- L'efficacité de codage est limitée.
- Le codage par blocs de 8×8 pixels génère un effet de mosaïque à bas débit gênant visuellement.
- La transmission d'images codées est très peu robuste en environnement bruité.
- Les applications liées à l'image sont de plus en plus spécifiques et nécessitent de nouvelles fonctionnalités non résolues par JPEG.

1.5.3 Transformation en ondelettes

L'introduction de la transformée en ondelettes DWT avait pour but de faire face aux problèmes de non stationnarité des signaux. Lorsqu'on applique la DWT à une image, la transformation est appliquée sur l'ensemble de l'image et non pas par partitionnement, en conséquence absence d'effet de blocs lors de la reconstruction des données. Ceci permet une répartition uniforme de l'erreur de compression sur toute l'image. Donc, au lieu de représenter un signal comme la somme pondérée de sinusoides comme c'est le cas de la transformée de Fourier, nous avons la somme pondérée d'ondes translatées et dilatées dans le temps. La transformation en ondelettes permet d'obtenir une représentation temps-fréquence [46], [47] ou temps échelle. Elle a des propriétés d'adaptation ou de flexibilité très attrayantes notamment le choix des fonctions de bases des ondelettes et des paramètres de dilatation et translation [48]. Les transformées en ondelettes [49], [50] conservent l'énergie du signal et possèdent notamment des algorithmes rapides [51], elles sont donc bien adaptées à la compression d'image.

L'intérêt des ondelettes par rapport aux sinus et aux cosinus se situe surtout à deux niveaux :

- Contrairement aux sinus et cosinus qui ne sont bien localisés qu'en fréquence, les ondelettes le sont également en temps. Par conséquent tout changement de fréquence dans la transformée en ondelettes ne produira de changements que sur une certaine partie du domaine temporel.

-Les ondelettes permettent de représenter de manière compacte un grand nombre de fonction : ainsi les fonctions formées de pics très prononcés nécessitent beaucoup moins d'ondelettes que de sinus ou cosinus pour être représentées.

Les ondelettes sont utilisées dans les deux catégories de techniques de compression que sont la compression sans perte et avec perte [52]

1.6. Les critères d'évaluations de la qualité de compression

Dans la pratique, plusieurs techniques subjectives et objectives sont utilisées [53]. Ils donnent une mesure de performance de la méthode de compression utilisée. Les principaux critères d'évaluation de toute méthode de compression sont :

- Le taux de compression.
- La qualité de reconstruction de l'image.
- La rapidité du codeur et du décodeur.
- La robustesse aux erreurs de transmission.

1.6.1 les critères d'évaluations objectives des méthodes sans pertes

Les compromis liés à ce mode de compression sont selon trois axes [53] illustrés figure 1.7.

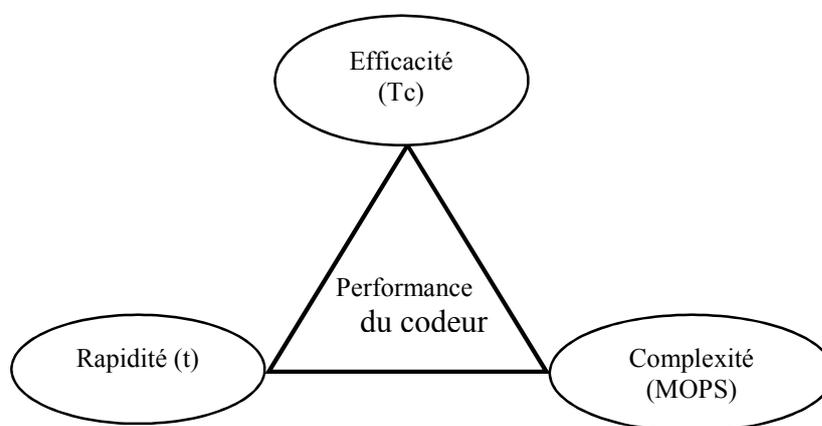


Figure 1.7 : Compromis dans le mode de codage sans perte

- Efficacité du codage : Exprimée par le taux de compression elle est limitée par l'entropie de la source. Plus l'entropie de la source est grande plus il est difficile de la compresser (par exemple un bruit aléatoire).

- Temps de codage : Il est lié à la complexité du processus de codage ou de décodage. Il peut être réduit si on augmente la capacité de calcul du composant de traitement. Pour certaines applications ce temps est une contrainte qui impose le choix de la technique de codage.

- Complexité du codeur : Elle peut être mesurée à l'aide de la taille mémoire utilisée et du nombre d'opérations arithmétiques. Le nombre d'opérations est donné par MOPS (Million of Operations Per Second). Le MIPS (Millions of Instruction Per Second) est parfois utilisé. Si on ajoute l'aspect mobile de l'application, la consommation d'énergie peut être considérée comme caractéristique de la complexité de codage.

1.6.2. Les critères d'évaluations des méthodes avec pertes

Pour ce mode de compression, en plus des trois axes de compromis pour la compression sans pertes, on ajoute un quatrième axe qui est la qualité du signal figure 1.8 : il est utilisé pour caractériser le signal à la sortie du décodeur. Plusieurs mesures sont proposées pour la qualité du signal parmi lesquelles le SNR. (Signal-to-Noise Ratio), le PSNR (Peak-SNR), et le MSE (Mean Square Error).

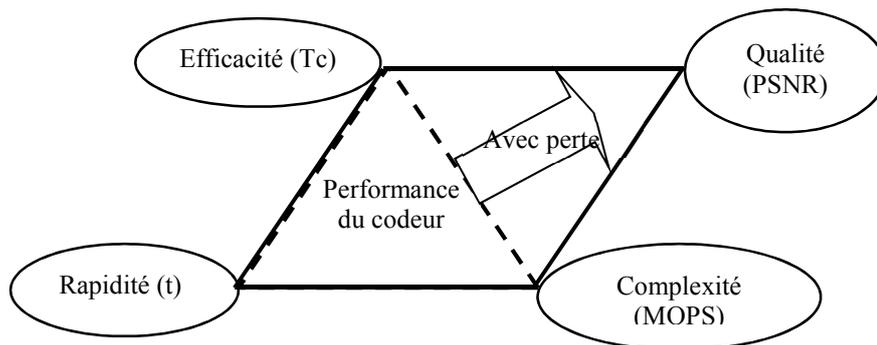


Figure 1.8: Compromis dans le mode de codage avec perte.

1.6.2.1. Le taux de compression

Le rapport de compression est l'une des caractéristiques les plus importantes de toute méthode de compression, il est défini comme :

$$R_c = \frac{\text{Espace mémoire occupé par les données non compressées}}{\text{Espace mémoire occupé par les données compressées}} \quad (1.1)$$

On définit alors le taux compression (pourcentage) par :

$$T_c = 100 \times \left(1 - \frac{1}{R_c} \right) \quad (1.2)$$

Dans la pratique, on utilise plutôt le débit pour mesurer le pouvoir de compactage d'une méthode. Le débit est exprimé en bits par pixel :

$$RC(\text{bpp}) = \frac{\text{nombre de bits codés}}{\text{taille de l'image originale (nombre de pixels)}}$$

1.6.2.2. L'évaluation de la distorsion.

La distorsion est l'information perdue entre le signal original et le signal décodé en fin de chaîne. On utilise deux méthodes, une subjective l'autre objective :

- La méthode subjective est basée sur des tests psycho-visuels de l'œil humain. En imagerie médicale, l'avis d'un médecin spécialiste est indispensable pour confirmer la validité de l'image compressée vis-à-vis du diagnostic.
- Les méthodes objectives sont :

1.6.2.2.1. Erreur Quadratique Moyenne MSE (Mean Square Error).

Ce critère se calcule comme la moyenne des carrés des écarts entre les pixels de l'image reconstruite et les pixels correspondants de l'image originale. La grande valeur de MSE signifie que l'image est de mauvaise qualité. Le MSE est défini comme suit :

$$\text{MSE} = \frac{1}{R} \sum_{i=1}^R (X_i - x_i)^2 \quad (1.3)$$

R : représente la résolution de l'image (nombre total de pixels)

1.6.2.2.2. Rapport signal sur bruit crête PSNR (Peak Signal to Noise Ratio)

La faible valeur du PSNR signifie que l'image est de mauvaise qualité. Le PSNR est défini comme suit :

$$\text{PSNR} = 10 \times \log_{10} \left(\frac{D^2}{\text{MSE}} \right) \text{dB} \quad (1.4)$$

D : est la valeur maximum que pourrait prendre un pixel ; $D = 2^{\text{BD}} - 1$

BD (Bit Depth) : nombre de bits sur lequel les pixels sont codés.

1.6.2.2.3. Similarité structurelle SSIM (Structural SIMilarity).

Basé sur la distorsion structurelle [54] - [56], le SSIM est utilisée comme indicateur de la qualité de l'image décompressée, il fournit un moyen pour quantifier la similarité perceptuelle entre deux images.

Les méthodes traditionnelles sont basées sur le calcul d'erreur entre les valeurs de chaque pixel d'une image distordue et une autre référence ; par contre, le SSIM mesure la similarité de structure de ces deux images. Le SSIM de deux images x et y est défini à l'aide de la moyenne (μ) et la variance (σ) de chaque image, ainsi que de leur covariance (cov), comme suit :

$$SSIM(x,y) = \frac{(2 \mu_x \mu_y + C_1)(2 cov_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (1.5)$$

C_1 et C_2 sont des constantes :

$$C_1 = (K_1L)^2 \text{ et } C_2 = (K_2L)^2$$

L est la dynamique des valeurs des pixels, soit 255 pour une image codée sur 8 bits.

En utilisant l'expression SSIM précédente sans les constantes C_1 et C_2 , les valeurs calculées deviennent instables. Ce problème a été résolu en ajoutant deux constantes, calculées, avec $K_1 = 0.01$ et $K_2 = 0.03$.

Très peu d'études ont été réalisées pour proposer d'autres critères objectifs d'évaluation de la qualité spécifiques à l'imagerie médicale.

1.6.3. Les critères d'évaluations subjectives.

Les mesures de la qualité subjective se fait par des tests psychophysiques ou par des questionnaires avec notations réalisés par des professionnels de la santé. Dans ce cas, on demande à un ensemble de radiologues de noter la qualité des images compressées (généralement sur une échelle de 1 à 5) selon que celles-ci leur permettent ou non d'effectuer un diagnostic (ou une mesure particulière d'organe).

1.7. Conclusion

Généralement les méthodes qui atteignent des taux de compression très élevés sont les méthodes irréversibles. Par contre, les méthodes non réversibles engendrent des taux de compression très faibles. La redondance dans le domaine spatial n'est pas facilement identifiable et généralement ne fournit pas toujours un bon taux de compression. Il est donc nécessaire de faire une transformation pour obtenir une décorrélation de l'information spatiale et un groupement d'énergie fréquentielle. On distingue trois approches ; l'approche fréquentielle, utilisée dans JPEG par la transformée en cosinus discrète ; l'approche spatio-fréquentielle, retenue dans JPEG2000, basée en particulier sur la transformée en ondelettes et l'approche par fractales ou plus récemment par ondelettes géométriques visant à améliorer la prise en compte de l'information géométrique des contours dans l'image.

Le principe de la quantification scalaire ou vectorielle est de réduire la taille des coefficients de façon que cette réduction n'apporte pas de dégradations visuelles à l'image et de définir un dictionnaire de taille finie, cela a donné lieu à de nombreux algorithmes : les algorithmes neuronaux et les algorithmes non neuronaux.

Dans cette thèse, nous nous intéressons à la compression avec perte des images médicales ; présenté au chapitre II.

CHAPITRE 2

PARTICULARITES DES IMAGES MEDICALES

2.1. Introduction

L'imagerie médicale est devenue un outil de diagnostic incontournable dans le milieu médical. Quelle que soit la méthode d'acquisition, radiographie, angiographie, scanner, résonnance magnétique, échographie, endoscopie etc. C'est le procédé par lequel un médecin peut examiner l'intérieur du corps d'un patient sans l'opérer. L'imagerie médicale peut être utilisée à des fins cliniques pour l'établissement d'un diagnostic ou pour le traitement de pathologies mais également dans le cadre de travaux de recherche scientifique des êtres vivants.

L'utilisation de l'image médicale a donné lieu à la création de standards de codage, de stockage et d'échange tels que DICOM (Digital Imaging and Communications in Medicine) [57], et PACS (Picture Archiving and Communication System).

Ce chapitre est un état de l'art des images médicales ; et des méthodes de compression appliquées aux images médicales. Les images médicales n'ont pas échappé aux techniques et aux standards classiques de compression mais de nombreuses méthodes spécifiques ont été développées afin d'optimiser le taux de compression et la qualité de l'image reconstruite.

2.2. Spécificités des images médicales

Suivant les techniques utilisées, les examens d'imagerie médicale permettent d'obtenir des informations sur l'anatomie des organes (leur taille, leur volume, leur localisation, la forme d'une éventuelle lésion, etc.) ou sur leur fonctionnement (leur physiologie, leur métabolisme, etc.). Dans le premier cas on parle d'imagerie structurelle et dans le second d'imagerie fonctionnelle.

Ces différents types de techniques sont souvent employés de façon complémentaire parfois même au sein d'un même système d'imagerie qui permet alors des acquisitions multimodales, simultanées ou non.

Chacune des modalités d'imagerie médicale a ses propres spécificités liées aux phénomènes physiques exploités et physiologiques exploités comme l'illustre la figure 2.1 l'intensité d'un

pixel dépend des caractéristiques chimiques ou physiques d'un tissu, elles-mêmes souvent liées à un phénomène physiologique. Cette intensité reflète :

- Un coefficient d'atténuation des rayons X pour la radiologie,
- Une concentration locale de produit de contraste en radiologie ou en IRM.
- Un changement de concentration en oxyhémoglobine en IRM fonctionnelle.
- Un coefficient d'atténuation ou de réflexion en échographie.

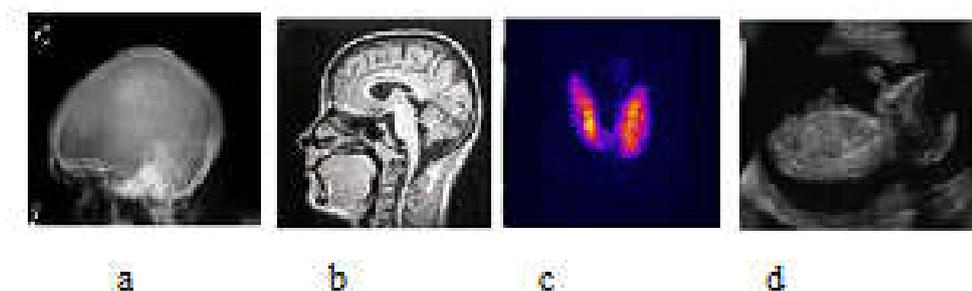


Figure 2.1. Vues du cerveau pour différentes modalités d'imagerie

- a) Tomodensitométrie(TDM). b) Imagerie par résonance magnétique (IRM)
c) Tomographie par émission de positions(TEP) d) Echographie.

2.2.1. La radiographie

La radiographie standard permet principalement d'obtenir des clichés en deux dimensions des structures osseuses et articulaires : elle est notamment utilisée en orthopédie, en rhumatologie et en orthodontie où elle permet d'étudier les traumatismes osseux (fractures, etc.), les déformations du squelette ou les implantations dentaires. La pneumologie y a aussi recours (radio des poumons). Il est aussi possible de visualiser certains organes ou parties creuses, habituellement invisibles aux rayons X, en les « remplissant » d'un produit de contraste, opaque aux rayons X : c'est la radiographie de contraste [58].

2.2.2. Le scanner (tomodensitométrie TDM)

Le scanner explore un volume du corps : c'est en quelque sorte une « endoscopie virtuelle » qui permet d'établir des images tridimensionnelles des organes ou des tissus (os, muscles ou vaisseaux) constitutifs des zones scannées. À son apparition, il a révolutionné la neurologie en offrant la possibilité de « voir » le cerveau. Dans cette dernière indication, il tend de plus en plus à être remplacé, lorsque cela est possible, par l'IRM. Car si le scanner

permet de visualiser une modification de volume ou une anomalie de structure (tumeur, embolie, anévrisme...), il ne permet pas d'en préciser la nature (inflammation, cancer, etc.). En cancérologie, il permet de contrôler la réponse à la chimiothérapie. Il peut aussi être utilisé en chirurgie pour renseigner plus précisément sur les zones où l'intervention est envisagée ou bien encore pour guider les drainages et les biopsies. Pour l'étude de certains organes, il peut être nécessaire d'injecter (par voie intra-veineuse) ou d'ingérer un produit de contraste à base d'iode, opaque aux rayons X [59].

2.2.3. L'échographie

Les images ultrasonores utilisent les propriétés acoustiques de la matière. Une sonde émet des ondes de pression ultrasons qui sont ensuite plus au moins absorbées et réfléchies selon la densité des tissus biologiques rencontrés [60]. L'échographie permet d'explorer le cœur, les organes digestifs, urinaires et génitaux (prostate et testicules, ovaires et utérus). Le tube digestif et les poumons, de même que les os et les gaz, ne lui sont en revanche pas accessibles.

L'échographie doppler, qui utilise une sonde plus fine, permet d'explorer le système vasculaire et a des indications majeures pour les maladies des veines et des artères. Les échographies de la grossesse permettent d'apprécier la vitalité et la morphologie du fœtus ainsi que son environnement (liquide amniotique, placenta, cordon, ...).

2.2.4. L'imagerie par résonance magnétique (IRM)

L'imagerie par résonance magnétique (IRM) [61] permet de visualiser la structure anatomique de tout volume du corps, en particulier des « tissus mous » tels que le cerveau, la moelle épinière, les viscères, les muscles ou les tendons. L'IRM anatomique, qui repose sur l'aimantation des atomes d'hydrogène qui est basé sur la détection des propriétés magnétiques des protons contenus dans les molécules d'eau du corps ou la concentration en eau des tissus varie d'un organe à l'autre et son état physiologique. L'IRM permet aussi de suivre l'activité d'un organe tel le cerveau, à travers l'afflux de sang oxygéné dans certaines de ses zones (IRM fonctionnelle, qui repose sur l'aimantation des noyaux d'hydrogène induite par la présence de l'hémoglobine des globules rouges). Contrairement au scanner X, qu'elle remplace dans de nombreuses indications, l'IRM permet de mieux définir la nature des lésions observées. Elle est utilisée en neurologie. Diagnostic de la sclérose en plaques, de la maladie d'Alzheimer, de l'épilepsie, des accidents vasculaires cérébraux ou des lésions

de la moelle épinière ; ainsi qu'en ophtalmologie, en endocrinologie, en Oto-rhino-laryngologie (ORL), en ostéoarticulaire et en cardiovasculaire.

2.2.5. Tomographie par émission de positions(TEP)

C'est une scintigraphie faite après l'injection d'un traceur faiblement radioactif, il va se fixer au niveau des tissus cancéreux pour émettre, de façon temporaire, des rayonnements pour obtenir des informations sur le métabolisme des cellules qui composent les tissus étudiés au cours de son fonctionnement. Ses applications principales concernent l'étude du métabolisme musculaire et l'étude du métabolisme cérébral [62].

2.3. Les propriétés des images médicales

Les propriétés des images médicales dépendent essentiellement de la modalité d'acquisition, de l'organe étudié et du protocole d'acquisition utilisé. Le tableau 2.1 présente à titre indicatif l'ordre de grandeur de la taille des images et des fichiers images acquis en routine clinique pour différentes modalités et différents organes [63].

Tableau 2.1. Propriétés des images médicales pour différentes modalités et différents organes

Modalité	Organe	Taille de l'image	Nombre de bits par pixel	Nombre de coupes	Taille du fichier
radiologie	thorax	2060*2060	16	-	8 M ₀
TDM	Cerveau	512*512	16	≈300	150 M ₀
	Abdomen	512*512	16	≈500	250 M ₀
	Cœur	512*512	16	126*16phases	1G ₀
IRM	Cerveau	512*512	16	≈20*6 Séries	10 à 60 M ₀
	Abdomen	512*512	16	≈ 30	15 M ₀
	Abdomen3D	512*512	16	104	50 M ₀
	Cœur	256*256	16	20*20phases	50 M ₀
TEP	Corps entier	128*128	16	350	10 M ₀
	Cerveau	256*256	16	47	6 M ₀
	Cœur	128*128	16	47*16phases	24 M ₀

2.4. Compression réversible des images médicales

Une comparaison de méthodes de compression sans perte des images médicales est proposée par Kivijarvi [64]. Les résultats montrent que le taux de compression obtenu dépend fortement du type de l'image. Jpeg-Ls présente les meilleurs taux, tous types d'image confondus, pour des temps de calculs raisonnables.

David Clunie [65] présente également une étude sur la compression sans perte pour plusieurs modalités d'images. Ces travaux montrent que les standards internationaux de compression Jpeg-Ls et Jpeg 2000 sont aussi performants que les meilleures techniques proposées à l'époque dans la littérature, quelle que soit la modalité d'image médicale. Clunie recommande fortement l'adoption de Jpeg-Ls pour sa simplicité, sa facilité d'implémentation et sa rapidité de calcul.

Les études menées par Adamson conclue [66], que les taux de compression sont globalement fonctions du type de l'image médicale pour une compression réversible.

2.5. Compression irréversible des images médicales.

Bien que la compression réversible des images médicales offre le maximum de garantie aux médecins quant à la qualité des images analysées, les taux de compression obtenus avec ces méthodes demeurent faibles par rapport aux taux potentiels des méthodes de compression irréversibles.

De nombreux travaux [67] - [70] ; essaient de quantifier la perte acceptable pour ne pas perturber le diagnostic par la compression avec pertes pour l'imagerie médicale. Les études de Koff et al [71], s'attachent à définir des taux de compression pour lesquels les distorsions sont suffisamment faibles pour être tolérées lors de tests subjectifs. L'Association Canadienne des Radiologistes (CAR) [72] a également publié une norme visant à orienter les radiologues sur les taux de compression maximaux pouvant être utilisés avec les standards JPEG et JPEG2000 pour différentes modalités d'images et régions anatomiques étudiées. La plupart de ces études sont réalisées à l'aide des protocoles d'évaluations strictes, difficiles à mettre en place et coûteuses. Ces expertises portent souvent sur la qualité du diagnostic après compression. Ainsi, pour évaluer correctement un algorithme, il faut pouvoir réunir des images (de patients malades et sains) pour toutes les techniques d'acquisition (et de reconstruction) concernées, et pour toutes les pathologies connues. Ces images doivent ensuite être évaluées par plusieurs médecins spécialistes, pour différents taux de compression. D'autres métriques objectives d'évaluation de la qualité par

apprentissage sur des résultats d'experts a été envisagée [73] ; afin d'optimiser les pertes en compression, pour permettre d'éviter l'introduction d'artefacts, de limiter la suppression de signes pathologiques (informations utiles aux diagnostics). On trouve quelques références à des travaux utilisant des modèles psychovisuels humains [74], en compression d'images naturelles.

2.5.1 Compression JPEG2000 avec pertes des images médicales

Quelques travaux font état de l'application de la compression JPEG 2000 sur diverses images médicales [75]. Les taux de compression acceptables définis après analyse de la fidélité diagnostique sont présentés dans le tableau 2.2.

Tableau 2.2 Application de la Compression JPEG2000 sur différentes images médicales.

Type d'images	Taux de compression acceptable
Radiologie thoracique	20 :1
mammographie	20 :1
Image de poumon	10 :1
IRM de cerveau	25 :1
Echographie	12 :1
Angiographie coronaire	30 :1

2.5.2. Méthodes de quantification vectorielle appliquées aux images médicales

La performance d'un codeur est fortement liée aux caractéristiques des images à traiter. Le tableau 2.3 regroupe un ensemble d'approches appliquées aux images médicales et fournit une vision plus globale et plus large sur les techniques élaborées [76].

Tableau 2.3 : Quelques approches de quantification vectorielle appliquées sur des images IRM.

Méthodes/approches	Type d'image	Références
pruned tree-structured VQ (PTSVQ),	IRM	[78]
scalar-vector quantizer (SVQ)	IRM	[79]
variable-length tree-structured VQ (DI-VLTSVQ)	IRM	[80]

2.6. Conclusion

Les images médicales ne doivent pas être considérées comme des images naturelles en raison des propriétés spécifiques de la source qui les génère, de leur contenu fréquentiel, de la corrélation entre échantillons et de leur dynamique.

La plupart des systèmes de compression sont testés sur des images naturelles monochromes ou couleur (l'image de Lena étant la plus utilisée) dont l'intensité des pixels correspond à un coefficient de réflexion de la lumière naturelle. Les images qui sont acquises en routine clinique traduisent quant à elle des phénomènes physiques et physiologiques complexes, de nature très variée, d'où une grande diversité d'images.

Plusieurs méthodes ont été appliquées aux images médicales, terme qui regroupe les images de toutes les modalités (radiographie, IRM, échographie...). Or, les caractéristiques de ces images sont très différentes d'une modalité à une autre et malgré les récentes avancées technologiques aucune étude n'a encore déterminé un taux de compression référence unique pour un type d'images, une modalité et un organe et une méthode de compression pour toutes les images médicales.

Notre étude consiste à proposer une méthode de compression avec pertes en préservant l'information diagnostique des images médicales IRM par une méthode hybride ; (la transformée d'ondelette et les réseaux de neurones), qui seront détaillés au chapitre 3.

CHAPITRE 3

TRANSFORMÉES EN ONDELETTES ET RESEAUX DE NEURONES

3.1. Introduction

Il existe de nombreuses méthodes utilisées en traitement de l'image. Les critères de sélection de l'une ou l'autre de ces méthodes peut dépendre de l'application envisager ; compression d'image, segmentation, détection de contours, etc.

Les transformations les plus utilisées en compression, que ce soit pour les images fixes ou les séquences d'images, sont la Transformée en Cosinus Discrète (DCT), la Transformée en Ondelette Discrète (DWT). Quel que soit la transformation, la variance (l'énergie) des composantes transformées est concentrée sur les basses fréquences. Mais les transformées discrètes en ondelettes sont les plus recommandées pour la compression d'images [81] - [84],

Dans le domaine du traitement d'images, les réseaux de neurones artificiels, inspirés du comportement du cerveau humain, ou les neurones sont des automates élémentaires ; dont le mode de fonctionnement s'inspire de celui des neurones biologiques ; ils permettent de développer une solution à des problèmes de différents domaines. Plusieurs recherches ont montré que l'utilisation du réseau de Kohonen permet de générer un Codebook assurant une meilleure représentation des données à traiter, ce qui entraîne une diminution du bruit de quantification de la compression des images [85], et en particulier les images médicales [86], exemple, Appendice A.

Dans ce chapitre, nous définirons la transformée en ondelettes dans le contexte de l'analyse multirésolution et nous présenterons les notions d'espaces d'approximation et de détail. Par la suite la quantification par l'algorithme de kohonen sera présentée.

3.2. Introduction à la transformée en ondelette

En 1983, J. MORLET a proposé l'analyse et la synthèse des signaux par les ondelettes « analyse multi résolution ». La méthode permet donc de décomposer un signal donné en une combinaison linéaire temps - fréquence judicieusement choisie.



Figure 3.1 : Schéma de la transformée en ondelettes [87]

Il serait préférable de disposer d'une méthode d'analyse dont la résolution, aussi bien en temps qu'en fréquence, s'adapte au signal en fonction de ses caractéristiques. Pour ce faire, plutôt que de conserver une enveloppe fixe dans laquelle le nombre d'oscillations varie, on conserve un nombre d'oscillations constant dans une enveloppe que l'on peut contracter et dilater à volonté : c'est l'analyse multi résolution, dont la figure 3.2 décrit le principe.

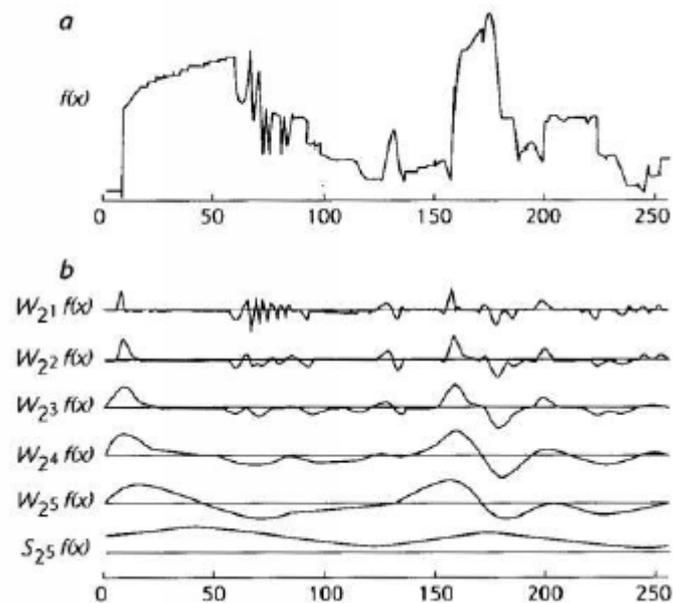


Figure 3.2 : Principe de l'analyse multi résolution [87]

Le signal d'origine (a) est décomposé en un ensemble de sous signaux (b) selon la procédure suivante :

La fonction $f(x)$ est décomposée en un signal de détail $W_1^2 f(x)$ est un signal d'approximation $S_1^2 f(x)$ (qui n'est pas représenté ici). Ce signal $S_1^2 f(x)$ est à son tour décomposé en un signal détail $W_2^2 f(x)$ et une approximation de $S_2^2 f(x)$, et ainsi de suite

jusqu'à obtenir les cinq signaux de détails $W_1^2 f(x)$ à $W_2^5 f(x)$ correspondant aux détails des cinq premières résolutions, et le signal d'approximation à la résolution 5, $S_{25}f(x)$.

Ainsi, l'analyse multi résolution agit comme un zoom mathématique sur le signal à analyser.

On peut, en faisant varier l'échelle d'analyse du signal, extraire les détails présents à telle ou telle résolution.

3.2.1. La transformée en ondelettes continue

La transformée en ondelettes [88], utilise des translations et des dilatations d'une fonction fixe, l'ondelette mère ψ .

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right) \quad ; \text{ Avec } a > 0 \quad (3.1)$$

Avec $a, b \in \mathbb{R}$; $a \neq 0$ ou a sert à dilater (compresser ou étendre) la fonction alors que b sert à la translater (la déplacer selon l'axe des temps).

Quand on analyse un signal $f(x)$ avec les ondelettes, on le transforme en une fonction de deux variables (le temps et l'échelle d'analyse du signal) qu'on peut appeler

$W(a,b)$ noté :

$$W(a,b) = \frac{1}{\sqrt{a}} \int f(x) \psi_{a,b}(x) dx \quad (3.2)$$

a : représente le facteur d'échelle. L'ondelette est plus comprimée lorsque le facteur d'échelle diminue.

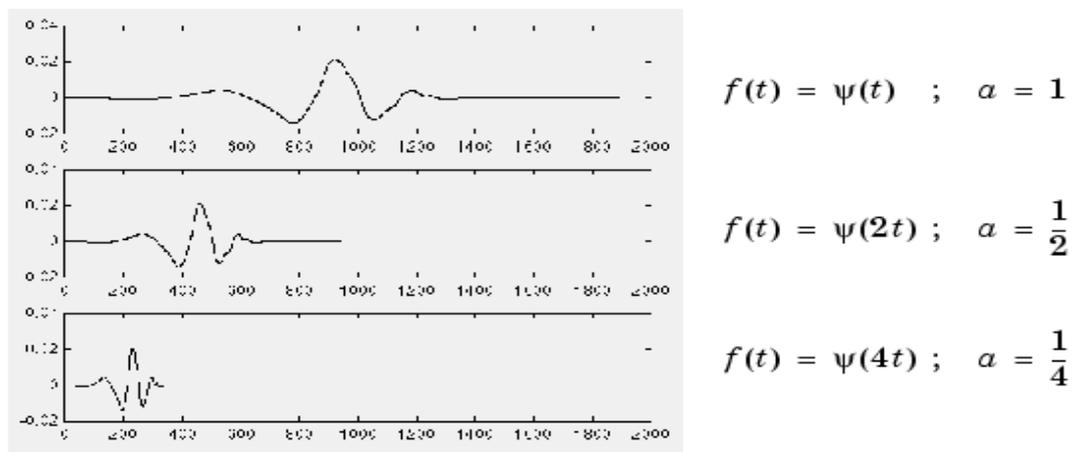


Figure 3.3 : Représentation de la fonction $\psi(t)$ en fonction du facteur d'échelle a

3.3. Analyse multirésolution de Stéphane Mallat. (S. Mallat).

L'algorithme qui donna une avancée aux ondelettes ne fut établi qu'en 1988 par Stéphane Mallat, qui est détaillée de façon théorique dans la référence [89].

Dès ses premiers travaux, avec son algorithme de décomposition et de reconstruction pour l'analyse multi résolution, les ondelettes se sont imposées comme une technique digne d'intérêt pour de nombreux problèmes de traitement de signal. Il a montré que les coefficients d'ondelettes peuvent être calculés à partir d'une transformée pyramidale mise en œuvre à l'aide de filtres numériques, récursifs ou non. Le principe de la transformée pyramidale consiste en la décomposition du signal à analyser à l'aide d'une paire de filtres. L'un de ces filtres fournira les coefficients d'ondelettes (ou détails), le second, les coefficients d'approximation sont à leurs tour décomposés par une seconde paire de filtres, l'ensemble constituant une pyramide de filtres.

Ce signal est décomposé en entités indépendantes qui appartiennent chacune à différentes échelles successives et défini comme un emboîtement des différentes descriptions à différent temps/fréquence. Chaque décomposition donne lieu à des signaux contenant une information discriminante.

D'un point de vue mathématique, l'analyse multirésolution consiste à approcher une fonction f par des approximations successives, représentées à différentes résolutions, (d'où le nom de multi résolution) et contenant de plus en plus d'informations dans des espaces emboîtés, en enregistrant un degré de corrélation avec l'ondelette. L'idée de base est, donc, de mesurer les changements entre les approximations f_j et f_{j+1} de f aux échelles respectives 2^j et $2^{(j+1)}$. (D'où l'appellation Dyadique).

La résolution en sortie de chaque paire (ou banc) de filtres étant deux fois inférieure à la résolution d'entrée, on parle d'analyse multirésolution dyadique.

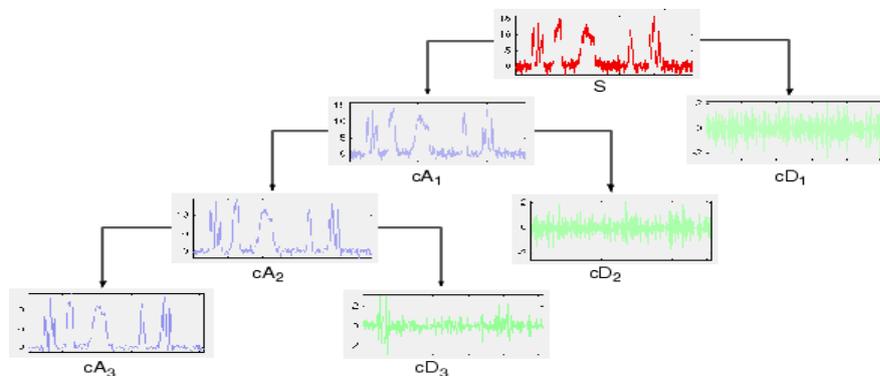


Figure 3.4 : Procédé de décomposition pour l'analyse multirésolution

La représentation multi-résolution d'une image peut se faire avec la transformée en ondelettes, cela produit quatre sous-bandes d'un quart de résolution ($4 \times 1/4 = 1$). La DWT est calculée sur les lignes (ou les colonnes) divisant l'image en deux : L et H, puis sur les colonnes (ou les lignes) divisant chaque partie en deux : LL, LH, HL et HH pour le niveau 1. Ce procédé est appliqué sur la bande LL pour obtenir des niveaux supérieurs de décomposition (figure 3.5).

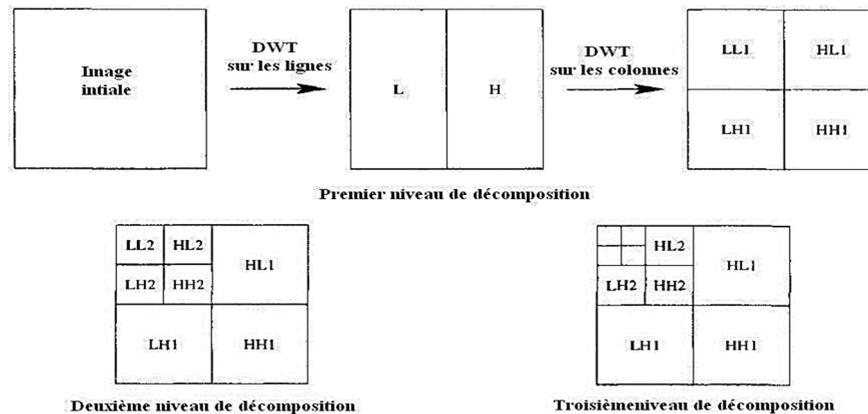


Figure 3.5 : Décompositions successives de la transformée en ondelettes

La bande LL est une approximation de l'image initiale contient les informations bases fréquences. Les trois autres bandes LH, HL et HH (horizontale, verticale et diagonale) contiennent les informations hautes fréquences figure 3.6.

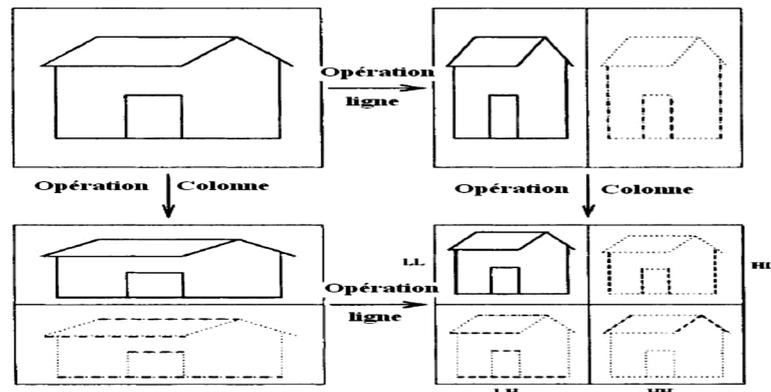


Figure 3.6 : Les quatre sous-bandes de l'image décomposée

3.4. Algorithme de décomposition –reconstruction de S. Mallat

Le principe de cet algorithme pour les ondelettes est présenté dans la figure 3.7 Soit $x(n)$ un signal échantillonné correspondant au signal d'origine. Ce signal est décomposé sur plusieurs niveaux de résolutions en deux bandes de fréquences (passe-haut et passe bas) à la manière approximation et détails.

3.4.1. Décomposition/reconstruction.

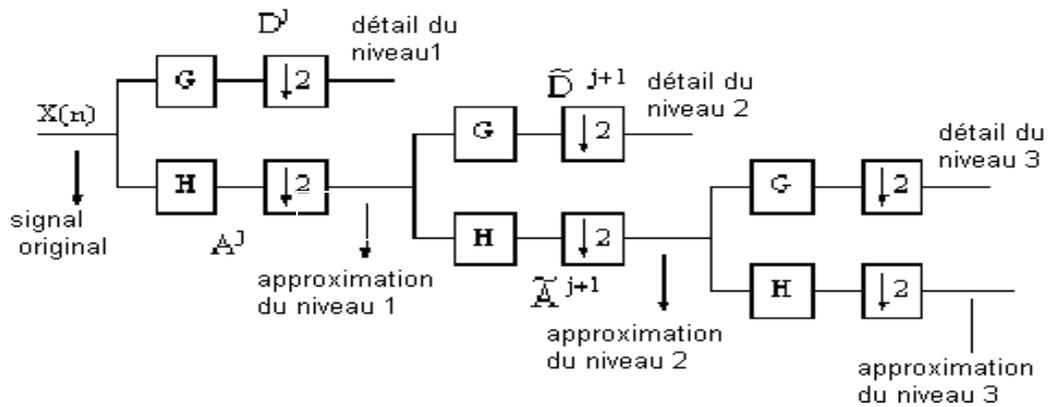


Figure 3.7 : Schéma de l'algorithme de décomposition de S.Mallat [89]

L'opération de décomposition passe par plusieurs itérations. Une itération est représentée par le schéma suivant :

A^j : signal d'approximation de niveau j

\tilde{A}^{j+1} : signal issu de la convolution de A^j avec le filtre H

D^j : signal de détail de niveau j .

\tilde{D}^{j+1} : signal issu de la convolution de A^j avec le filtre G

$$A^{j+1}(n) = \tilde{A}^{j+1}(2n) = \sum_k h(k-2n)A^j(k) \quad (3.3)$$

$$\tilde{D}^{j+1}(n) = \sum_k g(k-n)A^j(k) \quad (3.4)$$

3.4.1.1 Décimation :

L'opération de décimation consiste à prendre un échantillon sur deux d'un signal d'où :

$$A^{j+1}(n) = \tilde{A}^{j+1}(2n) = \sum_k h(k-2n)A^j(k) \quad (3.5)$$

$$D^{j+1}(n) = \tilde{D}^{j+1}(2n) = \sum_k g(k-2n)A^j(k) \quad (3.6)$$

Le signal A^{j+1} représente une version lissée du signal A^j

D^{j+1} représente la différence d'information entre A^j et A^{j+1} .

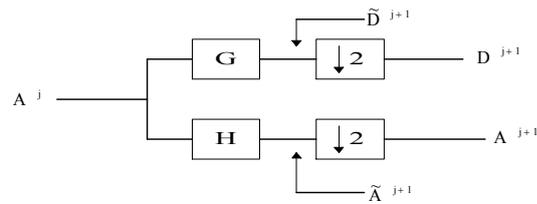


Figure 3.8 : Schéma de la décimation

3.4.1.2. Reconstruction.

La reconstruction est l'étape inverse de la décomposition avec une interpolation

Le schéma de reconstruction du signal original est le suivant :

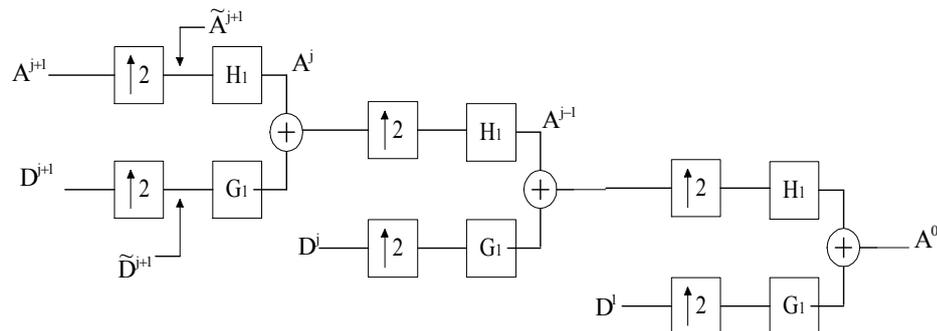


Figure 3.9 : Processus de reconstruction du signal

3.4.1.3. Interpolation par des zéros :

L'interpolation par des zéros consiste à insérer un zéro tous les deux échantillons successifs.

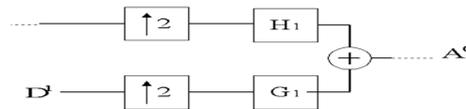


Figure 3.10 : Processus d'interpolation du signal

3.5. Calcul de complexité.

Le tableau 3.1 résume les différences entre les transformées et montre que le nombre d'opérations nécessaires pour effectuer la transformée en ondelettes sur n échantillons est directement proportionnelle à n [90].

Tableau 3.1 : Complexité algorithmique aux différents types de transformées sur n échantillons

Type de transformée	Opération
transformée de Fourier	$O(n^2)$
transformée de Fourier rapide	$O(n \log_2 n)$
transformée en ondelettes	$O(n)$

3.6. Carte auto-organisatrice (Carte de Kohonen)

Mis au point par Teuvo Kohonen, c'est un réseau de neurones, cet algorithme est basé sur un apprentissage non supervisé. Il donne en sortie une carte organisée adaptée à la topologie de l'espace d'entrée. C'est un réseau de neurones à une seule couche ou les neurones sont organisés sur une matrice $N \times M$. Chaque mot code du dictionnaire est représenté par un neurone de la carte : les valeurs des poids synaptiques sont donc les valeurs des mots codes. Après initialisation aléatoire, les neurones se déploient dans l'espace de sortie en respectant la topologie de l'espace d'entrée. Cet algorithme est utilisable en quantification scalaire comme en quantification vectorielle.

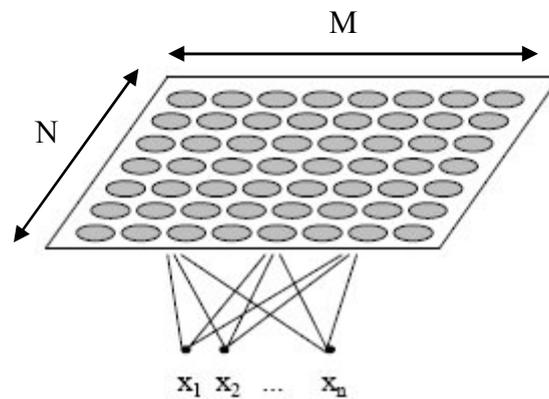


Figure 3.11 : Exemple d'une carte de Kohonen

3.6.1 Algorithme d'apprentissage

Dans cet algorithme, il faut bien faire la distinction entre la distance entre deux neurones utilisée pour le calcul du voisinage, et la distance euclidienne (relation 3.10) entre les neurones et les vecteurs d'apprentissage. La distance euclidienne est définie par la relation suivante :

$$D = \sqrt{\sum_{i=1}^n (W_i - X_i)^2} \quad (3.10)$$

L'algorithme se déroule comme suit :

1. Les poids synaptiques (W_i) sont d'abord initialisés avec des valeurs aléatoires.
2. A chaque itération, un vecteur d'apprentissage (X_i) est présenté à la carte, les distances euclidiennes entre ce vecteur et les poids synaptiques sont calculées et le neurone le plus proche est choisi comme étant le neurone gagnant.
3. Le neurone gagnant ainsi que ses voisins sont mis à jour par la relation (3.11).
4. La variable itération (t) est incrémentée et les mêmes étapes 2, 3, et 4 se reproduisent.

L'algorithme s'arrête en général après un certain nombre d'itérations choisi préalablement par l'utilisateur. La fonction de mise à jour des neurones est la suivante :

$$W_i(t+1) = W_i(t) + \beta_{i,k}(t) (X_i - W_i(t)) \quad (3.11)$$

$\beta_{i,k}(t)$ est une fonction d'apprentissage décroissante prenant en considération le nombre d'itérations et le voisinage. Il n'existe pas de règle générale définissant l'évolution des paramètres d'apprentissage, cependant Kohonen donne une forme particulière de la fonction β qui se révèle assez pratique. La relation de $\beta_{i,k}(t)$ est la suivante :

$$\beta_{i,k}(t) = \beta_0(t) \exp\left(-\frac{\|r_i - r_k\|}{\sigma(t)^2}\right) \quad (3.12)$$

Où $\beta_0(t)$ et $\sigma(t)$ sont des fonctions décroissantes appropriées, r_i représente les coordonnées du $i^{\text{ème}}$ neurone et r_k représente les coordonnées du $k^{\text{ème}}$ neurone qui est le neurone gagnant. $\beta_0(t)$ donne le coefficient d'apprentissage alors que $\sigma(t)$ diminue la taille du voisinage au cours du temps.

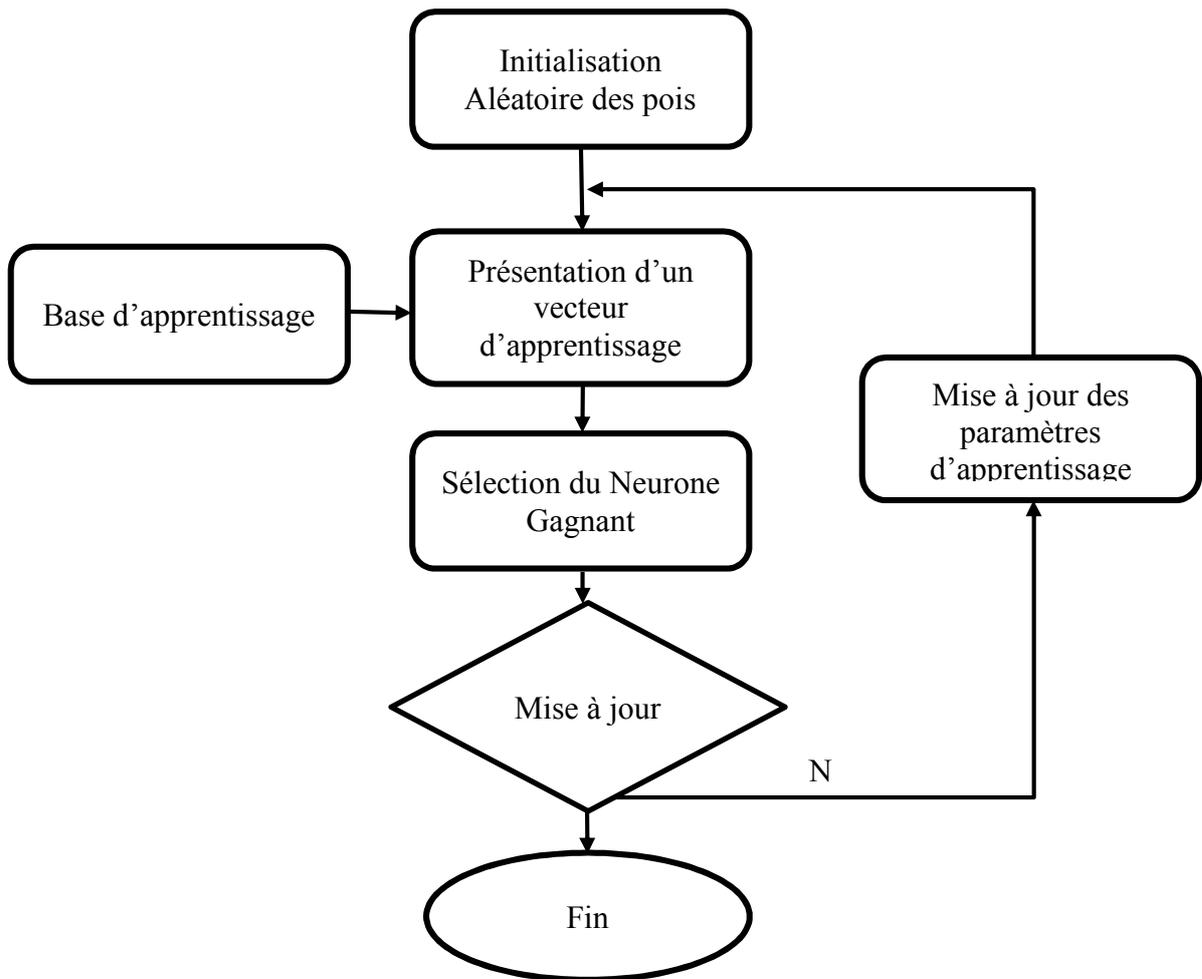


Figure 3.12 : Algorithme d'apprentissage de Kohonen

3.6.2. Propriétés de la carte auto-organisatrice

Elle permet de garder la topologie de l'espace d'entrée. On obtient une projection de ce dernier sur la matrice de sortie. La préservation topologique est la propriété principale de la carte auto-organisatrice de Kohonen.

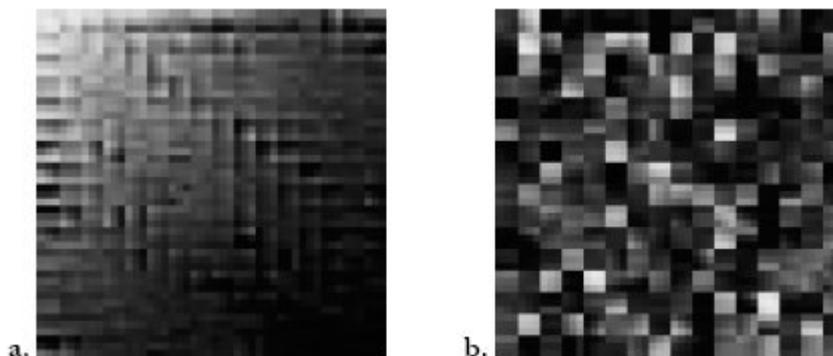


Figure 3.13 : Deux exemples de dictionnaire (a) : dictionnaire obtenu par l'algorithme de Kohonen (b) : dictionnaire obtenu par le GLA représenté sur une matrice.

- La carte de Kohonen est directement applicable pour la création de dictionnaires scalaires ou vectoriels. La base d'apprentissage est insérée comme espace d'entrée, et les valeurs des poids synaptiques obtenus en sortie représentent les mots codes du dictionnaire obtenu.
- Une conséquence de la préservation de la topologie de l'espace d'entrée est la robustesse lors de la transmission. Si un indice transmis est altéré durant son passage par un canal bruité, le vecteur choisi à la reconstruction sera erroné. Cependant, si le mot code correcte et le mot code erroné sont voisins donc pas très différents (grâce à la topologie maintenue), l'erreur sera moins perceptible visuellement.
- Le nombre de neurones choisi initialement doit être suffisant pour permettre la représentation correcte de l'espace d'entrée.

3.7. Evaluation de l'algorithme de S. Mallat par l'arithmétique en ligne

3.7.1. Introduction

Dans l'arithmétique, deux modes de calcul existent pour effectuer les opérations de base (addition, multiplication, etc.), chacun est caractérisé par la manière d'introduire les chiffres des opérands et de générer le résultat. Le premier est le mode parallèle ou classique dont les opérations s'effectuent de la même manière qu'à la main, c'est le plus utilisé, comme par exemple dans la famille des microprocesseurs qui utilise une seule unité arithmétique pour effectuer toutes les opérations. Cependant, lors de l'implémentation sur les circuits ASIC comme les FPGA on est souvent ramené à utiliser plusieurs circuits arithmétiques pour effectuer les différentes opérations séquentielles en parallèle ou avec des pipelines dans le but d'accélérer le processus de calcul, ce qui fait appelle au deuxième mode qui est le mode série.

Dans ce chapitre nous présentons brièvement le mode de calcul classique avec tous ces avantages et inconvénients. Puis, nous étudions le mode de calcul série en représentant son principe et ses avantages. Enfin, nous présentons l'arithmétique en ligne afin de modéliser l'algorithme de S MALLAT par cette arithmétique.

3.7.2. Mode parallèle

Les opérations et la génération de résultat s'effectuent de droite à gauche, c'est-à-dire de LSB jusqu'à MSB avec une possibilité de propagation de retenue à chaque calcul d'un chiffre ceci pose un problème. En effet, dans le cas d'une opération d'addition le calcul de rang $i + 1$ s'effectue après le calcul de rang i des opérands ce qui va ralentir la vitesse d'exécution donc une augmentation du temps d'exécution ; cette dernière varie linéairement avec la taille des opérands [91].

Dans ce mode, le traitement des opérands se fait d'une manière parallèle : tous les chiffres des opérands sont introduits en même temps et le résultat est généré de la même manière, ce qui va augmenter la taille de circuit d'addition lors de l'utilisation des opérands de grande taille dans le cas d'une réalisation matérielle. De plus, la taille des opérands est fixée donc une précision limitée, mais ce mode présente un avantage d'être efficace dans le cas de calcul avec des opérands de petit taille.

Pour surmonter tous ces inconvénients, les chercheurs se sont orientés vers l'utilisation du mode série qui utilise un seul circuit élémentaire pour effectuer toute les opérations des rangs ce qui tend à minimiser la surface.

3.7.3 Mode série

Le traitement se fait par introduction des opérandes chiffre par chiffre séquentiellement et de générer les chiffres de résultat à chaque itération. Ce mode est caractérisé par :

- Une communication série qui limite les lignes d'interconnexion interne et les pins d'entrées sorties.
- Un pipe ligne au niveau des bits, ce qui permet un calcul massif et simultané.
- La circulation des bits de données peut se faire avec MSB en tête ce qui permet une grande précision, c'est-à-dire de produire des résultats en même temps que les bits des données entrent.

3.7.3.1 Calcul avec LSB en tête

Ce mode de calcul s'effectue de la même façon que le calcul classique avec une propagation de retenue. Il est utilisé uniquement dans les opérations d'addition et de multiplication car ces dernières s'effectuent de droit à gauche ; les opérations de la division et de la comparaison ne peuvent pas être effectuées avec ce mode parce que la connaissance des bits dévidant et du diviseur est nécessaire pour la connaissance du bit de poids faible d'un quotient, de même pour la comparaison qui est faite à partir des bits MSB des opérandes.

3.7.3.2 Calcul avec MSB en tête

Dans ce mode de calcul, les chiffres des opérandes circulent en série, MSB en tête et de même pour le résultat à savoir MSB en tête, ce mode est appelé mode de calcul en ligne [92]. Générer le résultat à partir d'un MSB en tête nécessite une procédure de correction de chaque chiffre généré afin de converger vers le résultat le plus exacte, le principe sera détaillé dans ce qui suit.

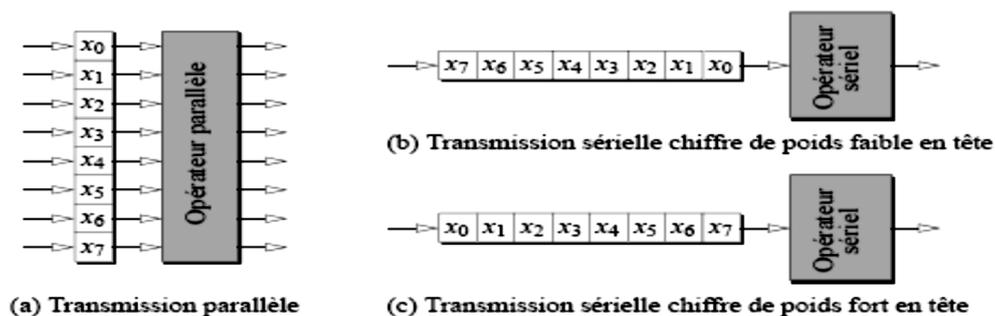


Figure 3.14 Divers modes de transmission d'un nombre entier de huit chiffres.

3.7.4 Arithmétique en ligne

Les premiers algorithmes de l'arithmétique en ligne ont été présentés par Trivedi et Ercegovic [92]. Ils présentent les caractéristiques suivantes :

- Les opérandes sont introduits, à chaque cycle, bit par bit, le bit de poids fort en tête,
- Les résultats sont obtenus de la même manière, bit par bit, poids fort en tête, mais avec un retard p , tel qu'au pas j , alors que le j (ième) digit des opérandes sont introduits, le $(j-p)$ ième digit du résultat est généré.
- Le retard p est très petit devant la taille des opérandes d'entrée.
- Les opérandes ainsi que les résultats sont écrits dans un système de notation redondant

Comme énoncé précédemment le mode série permet de réaliser un pipe ligne ; c'est-à-dire dans le cas des opérations successives le lancement d'une opération s'effectue dès que le premier bit de résultat de l'opération précédant est générée, la figure suivante montre ce principe avec la réalisation de l'opération $\sqrt{A + B + C \times D}$ en mode de calcul en ligne.

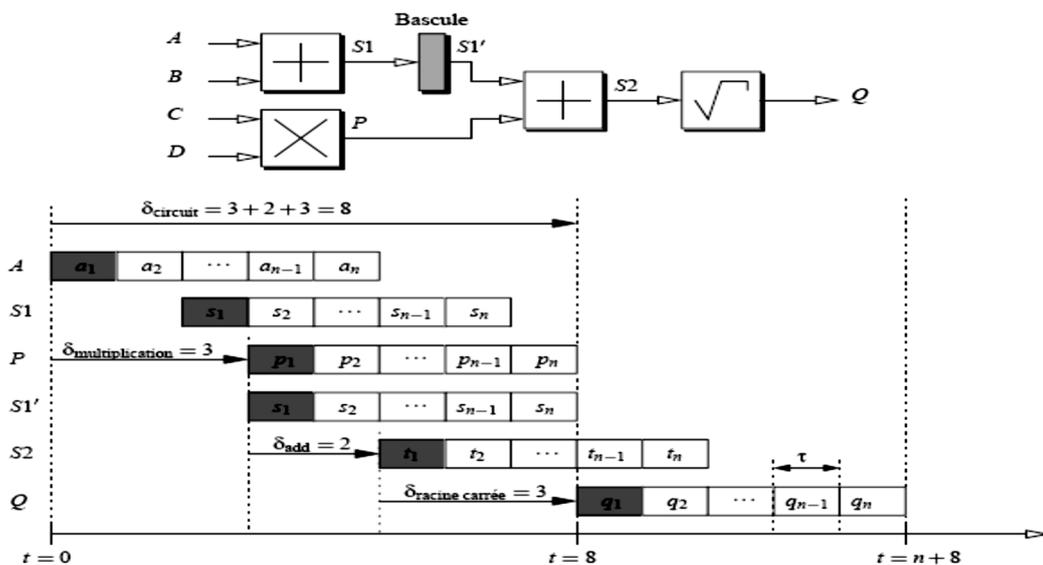


Figure 3.15 Calcul en ligne de l'expression $\sqrt{A + B + C \times D}$

3.7.4.1 Représentation des nombres dans les systèmes redondants

L'arithmétique en ligne utilise un système de notation et de calcul non conventionnel appelé système de notation d'Avizienis [93] ou notation redondante. Ce système introduit une multitude d'écriture, pour les nombres appelée redondance. Cette écriture offre deux aspects fondamentaux à cette arithmétique :

- 1- La possibilité de représenter un nombre sous différentes représentations d'où le concept de correction du nombre résultat au fur et à mesure de sa génération.
- 2- La possibilité d'effectuer des opérations sans propagation de retenue, soit chiffre de poids fort en tête.

3.7.4.1.1 Formulation mathématique

Les nombres sont représentés en base β avec des chiffres pris non pas dans l'ensemble $\{0, 1, \dots, \beta-1\}$ mais dans l'ensemble $\{-a, \dots, -1, 0, 1, \dots, a\}$, où a est un entier inférieur ou égal à $\beta-1$. Ces chiffres négatifs sont à l'origine de la désignation de ce système comme système de représentation des nombres à chiffres signés.

- Si $2a+1 \geq \beta \Rightarrow$ tous les nombres sont représentés dans ce système.
- Si $2a+1 < \beta \Rightarrow$ il existe un algorithme d'addition sans propagation de retenue.

Dans la suite et afin d'éviter toute confusion entre le signe des chiffres et l'opérateur de soustraction, nous noterons les chiffres négatifs avec une barre. Par exemple, le chiffre -1 est noté $\bar{1}$.

Exemple :

Le nombre 2345 peut s'écrire en base 10 modifier avec l'ensemble de chiffre

$\{-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5\}$ selon les codages suivants :

$$235(-5) = 2*1000 + 3*100 + 5*10 - 5 = 2345$$

$$24(-5) (-5) = 2*1000 + 4*100 - 5*10 - 5 = 2345$$

On dit alors que le système est redondant.

Autre exemple :

La représentation binaire du nombre $x=3/4$ est : 0,11 ; ce même nombre peut avoir les présentations suivantes :

$X= 1,1 \ 1$ ou $X= 1,01$ ou $X= 0,11$. Nous pouvons vérifier que :

$$\triangleright X= 1.2^0 - 1.2^{-1} + 1.2^{-2} = 3/4$$

$$\triangleright X= 1.2^0 + 0.2^{-1} - 1.2^{-2} = 3/4$$

$$\triangleright X= 0.2^0 + 1.2^{-1} + 1.2^{-2} = 3/4$$

L'intérêt que cette multitude d'écriture pour un même nombre représenté, pour le mode de calcul en ligne réside dans le fait que les digits résultats générés sont graduellement corrigés, de telle manière à ce que les digits générés convergent vers le résultat exact de l'opération. De plus il existe dans ces systèmes de représentation des algorithmes permettant d'effectuer des additions de façon totalement parallèle, sans propagation de retenue. En particulier, l'algorithme proposé par Avizienis.

3.7.4.1.2 Facteur de Redondance K

Le nombre g est le plus grand nombre représenté dans une base β avec une précision n :

$$K = \sum_{i=1}^n x_i \cdot \beta^{-i} = g / (\beta - 1) \quad (3.13)$$

Avec n tendant vers l'infini et les conditions sur g , [94]. Le système est défini comme suit :

Tableau 3.2 : type de facteur de redondance.

g	K	Système
$\beta / 2$	$\beta / (2 \cdot (\beta - 1))$	Redondance minimale
$\beta - 1$	1	Redondance Maximale

Les notations mathématiques utilisées dans ce travail sont présentées dans le tableau suivant :

Tableau 3.3 : Notations utilisées

Notation	Signification
K	Facteur de Redondance
β	Base de notation
g	Plus grand digit de la base
p	Retard en ligne
$\varepsilon_j = 1/2$	Erreur résiduelle
R[j]	Reste partiel
H[j]	Résidu total
S(H[j])	Fonction de sélection

3.7.5 Application à une fonction $Z=f(X, Y)$

La méthodologie d'application du traitement en ligne à une fonction :

- Définition de l'opération à effectuer.
- Définition des plages d'opérandes.
- Calcul du reste partiel.
- Décomposition de la relation de récurrence.
- Calcul du reste complet.
- Calcul de l'intervalle de choix.
- Détermination de la fonction de choix

3.7.6 Modélisation de l'algorithme de S.Mallat par l'arithmétique en ligne

3.7.6.1 Description mathématique

Les modèles des filtres à base d'ondelettes sont des filtres à réponse impulsionnelle finie d'ordre N , tel que :

$$Y_n^* = \sum_{i=0}^N h_i X_{n-i} \quad (3.14)$$

Où : Y_n^* : signal de sortie ; X_n : signal d'entrée ;
 h_i : Coefficients du filtre N : l'ordre du filtre.

A l'étape j , $X_n[j]$ et $Y_n^*[j]$ se présentent comme suit :

$$X_n[j] = X_n[j-1] + x_n[j]k^{-j} \quad (3.15)$$

$$Y_n^*[j] = Y_n^*[j-1] + y_n^*[j]k^{-j} \quad (3.16)$$

$$\text{Tel que : } 0 \leq |X_{n-i}| < 1 \quad (3.17)$$

Où b représente la base et r le coefficient de normalisation.

$$b^{-r} \sum_{i=0}^N |h_i| \leq \sum_{i=0}^N |h_i| |X_{n-i}| < \sum_{i=0}^N |h_i| \quad (3.18)$$

De l'équation (3.14), il en résulte :

$$b^{-r} \sum_{i=0}^N |h_i| \leq Y_n^* < \sum_{i=0}^N |h_i| \quad (3.19)$$

$$\text{Où, } Y_n^* \in \left[\sum_{i=0}^N |h_i| / b^r, \sum_{i=0}^N |h_i| \right] \Leftrightarrow Y_n^* / \sum_{i=0}^N |h_i| \in [1/b^r, 1[\quad (3.20)$$

L'algorithme en ligne est caractérisé par le retard p qui est défini par :

$$Y_n = b^{-p} \cdot Y_n^* / \sum_{i=0}^N |h_i| \quad (3.21)$$

Afin d'utiliser une architecture en pipeline, nous calculons le nouveau intervalle de convergence pour un délai nul $p=0$.

Donc, au lieu de calculer Y_n^* directement, on calcule :

$$Y_n = Y_n^* / \sum_{i=0}^N |h_i| \Leftrightarrow Y_n = \sum_{i=0}^N h_i X_{n-i} / \sum_{i=0}^N |h_i| \quad (3.22)$$

Posons $\sum_{i=0}^N |h_i| = D$

L'expression du nouveau filtre devient :

$$Y_n = \sum_{i=0}^N h_i X_{n-i} / D \quad (3.23)$$

3.7.6.2 Calcul de l'intervalle de convergence

Les opérandes appartiennent à l'intervalle,] $-b^{-r}$, b^{-r} [, donc :

$$|X_{n-i}| \in [0, b^{-r} [\quad (3.24)$$

$$\text{Et } |h_i| \cdot |X_{n-i}| \in [0, |h_i|. [\text{Pour les } (N+1) \text{ opérandes} \quad (3.25)$$

Donc :

$$\sum_{i=0}^N |h_i| |X_{n-i}| \in [0, \sum_{i=0}^N |h_i| \cdot b^{-r} [\quad (3.26)$$

Pour assurer la convergence de l'algorithme :

$$\sum_{i=0}^N |h_i| \cdot b^{-r} < 1 \quad (3.27)$$

$$\text{Il en résulte : } r > \log_b \left(\sum_{i=0}^N |h_i| \right) \quad (3.28)$$

3.7.6.3. Calcul en ligne

D'après l'équation (3.21) du nouveau filtre normalisé est :

$$Y_n = \sum_{i=0}^N h_i X_{n-i} / D \quad (3.30)$$

Prenons $b=2$.

La condition d'arrondissement symétrique est définie par :

$$Y_n[j] - \frac{2^{-j}}{2} \leq \sum_{i=0}^N h_i X_{n-i}[j] / D < Y_n[j] + \frac{2^{-j}}{2} \quad (3.31)$$

$$\Leftrightarrow -1/2 \leq 2^j \cdot \left(\sum_{i=0}^N h_i X_{n-i}[j] / D - Y_n[j] \right) < 1/2 \quad (3.32)$$

3.7.6.4. Le résidu partiel

Le résidu partiel $R[j]$ est défini comme suit :

$$R[j]=2^j \cdot [(\sum_{i=0}^N h_i X_{n-i}[j]/D) - Y_n[j]] \quad (3.33)$$

L'algorithme converge si et seulement si.

$$-1/2 < R[j] \leq 1/2 \quad (3.34)$$

En remplaçant les équations (3.15) ; On obtient :

$$R[j]=2^j [(\sum_{i=0}^N h_i X_{n-i}[j-1]/D) - Y_n[j-1]] + (\sum_{i=0}^n h_i x_{n-i}[j]/D) - y_n[j] \quad (3.35)$$

Après simplification de l'équation (3.33) on aura :

$$R[j]=2R[j-1] + L[j] - y_n[j] \quad (3.36)$$

$$\text{Posons } L[j] = (\sum_{i=0}^N h_i X_{n-i}[j]/D) \quad (3.37)$$

Comme étant des valeurs précalculées et stocker en mémoire.

3.7.6.5 Le résidu complet

Le résidu complet est décrit par :

$$H[j]=R[j] + y_n[j] \quad (3.38)$$

Des équations (3.35) et (3.37):

$$H[j]= 2.H[j-1] + L[j] - 2y_n[j-1]$$

Sachant que de l'équation (3.33):

$$-1/2 \leq R[j-1] \leq 1/2 \quad (3.39)$$

Utilisant (3.37) et (3.39)

$$-1/2 + y_n[j] \leq H[j-1] \leq 1/2 + y_n[j] \quad (3.40)$$

Avec $y_n[j] \in \{-1, 0, +1\}$: Redondance maximale utilisée

Finalement, le résidu complet définit la valeur du bit résultat par l'inégalité :

$$-3/2 \leq H[j-1] < 3/2 \quad (3.41)$$

Des équations (3.39) et (3.41), la fonction de sélection est donnée par $S(H[j])$, et définie par :

$$\begin{cases} H[j] = 2H[j-1] + L[j] - 2y_n[j-1] \\ y_n[j] = S(H[j]) \end{cases} \quad (3.42)$$

$$\text{Avec } y_n[j] \quad \left\{ \begin{array}{l} -1 \text{ si } -3/2 \leq H[j] < -1/2 \\ 0 \text{ si } -1/2 \leq H[j] < 1/2 \\ 1 \text{ si } 1/2 \leq H[j] < 3/2 \end{array} \right. \quad (3.43)$$

Notre travail consiste alors à implémenter les équations suivantes :

$$H[j] = 2.H[j-1] + L[j] - 2y_n[j-1] \quad (3.44)$$

$$\text{Avec: } L[j] = \sum_{i=0}^N h_i x_{n-i}[j] / D \quad (3.45)$$

3.7.7 Représentation en complément à deux

Dans ce mode de calcul les nombres sont représentés en système redondant tel que, en base 2 les chiffres peut prendre les valeurs -1, 0 ou 1, cette représentation est limitée uniquement au niveau logiciel. En effet, le passage à l'implémentation matérielle nécessite une représentation de ces valeurs en binaire, car les circuits numériques travaillent uniquement en deux états (l'état haut et l'état bas). Pour cela, nous avons adopté un codage en complément à deux ; son principe est de coder chaque chiffre x_i codé en système redondant en deux bit x_i^+ et x_i^- comme indiqué dans le tableau 3.2.

Tableau 3.4 : le codage en complément à deux

x_i	x_i^+	x_i^-
0	0	0
-1	1	1
1	0	1
interdit	1	0

3.8. Conclusion

Dans ce chapitre nous avons présentés l'algorithme de Stéphane Mallat qui présente un caractère séquentiel, qui réduit considérablement les performances globales. De ce fait, l'amélioration des performances de son exécution consiste à accélérer le processus de calcul de ces opérateurs sériels et à améliorer leurs performances en utilisant des architectures et des arithmétiques adéquates. Sa modélisation par l'arithmétique en ligne permet de résoudre le problème de rapidité et d'engendrer des architectures pour une implémentation hardware.

La partie de notre travail fut l'objet de plusieurs communications et d'une publication [95] - [99].

L'algorithme de Kohonen nous offre la possibilité de créer des quantificateurs scalaires ou vectoriels efficaces et robustes donnant une représentation précise de l'espace d'entrée et permettant de garder une bonne qualité à la réception lors de transmissions. L'inconvénient principal de cette méthode est qu'elle prend un temps considérable pour le calcul et son implémentation sur FPGA reste une tâche difficile pour un grand nombre de neurones.

Ces transformées présentent de nouveaux défis pour être implémentées et optimisées. Vu les possibilités offertes par ces transformées, elles sont potentiellement candidat pour être adoptés pour un grand nombre d'applications telles que l'imagerie médicale, le cinéma numérique, la surveillance, les satellites, les applications mobiles (appareil photo, PDA, téléphone cellulaire,).

Ces différentes applications présentent plusieurs exigences et contraintes : le temps, la qualité, ou la consommation d'énergie, Vu cette variété d'applications et d'exigences, trouver une implémentation et une architecture pouvant satisfaire l'ensemble de ces contraintes se révèlent très difficile. Quelle implémentation et quelle architecture choisir ?

CHAPITRE 4

IMPLEMENTATION SOFTWARE ET HARDWARE

DE LA TRANSFORMEE D'ONDELETTES ET LA CARTE SOM

4.1. Introduction

Les progrès technologiques continus dans le domaine des circuits intégrés ont permis la réduction des coûts de consommation et de la taille des systèmes numériques ; ainsi que la réalisation de circuits de plus en plus complexes, tout en améliorant leurs performances et leurs fiabilités. Aujourd'hui, les techniques de traitement numérique occupent une place majeure dans tous les systèmes électroniques modernes grand public, professionnels ou de défense. De plus, les techniques de réalisation de circuits spécifiques, tant dans les aspects matériels que dans les aspects logiciels font de la microélectronique une des bases indispensables pour la réalisation de systèmes numériques performants. Elle impose néanmoins une méthodologie de développement en conception assistée très structurée.

Les concepteurs des circuits intégrés utilisent depuis longtemps la simulation, aucun ne peut se permettre de fabriquer une puce, de tester son fonctionnement, puis en cas de problème, d'effectuer la modification. Un simulateur permet de détecter des erreurs et les corriger avant la fabrication ; La microélectronique est un des domaines où les erreurs coûtent cher.

Lorsqu'un circuit est disponible, l'expérimentation consiste à prendre des composantes, composer une maquette, la tester, l'adapter, la tester à nouveau. Les méthodes changent et de nouveaux outils apparaissent, il fallait un langage descriptif de haut niveau pour la modélisation et la simulation. Avec l'apparition des outils de synthèse et des langages de description du matériel, la méthodologie de conception a beaucoup évolué. Cette nécessité s'imposa au département de la défense des Etats-Unis (DoD) au début des années 80 et donna naissance aux langages de description hardware comme le VHDL (Very high Speed integrated circuits Hardware Description Language).

Dans ce chapitre, nous allons présenter la description Software et Hardware et la simulation de notre architecture.

PARTIE SOFTWARE

4.2 Méthode proposée : modélisation et conception

Notre travail se divise en deux blocs :

- Bloc 1 : partie prétraitement par la transformée en ondelettes, utilisation de l'algorithme de S.Mallat.
- Bloc 2 : Après l'obtention des 4 bandes (approximation et détails). Nous avons choisie de faire une QS sur les bases fréquences ou l'entropie de l'image et concentré et faire une QV sur les trois bandes détails non perceptible à l'oil nu.

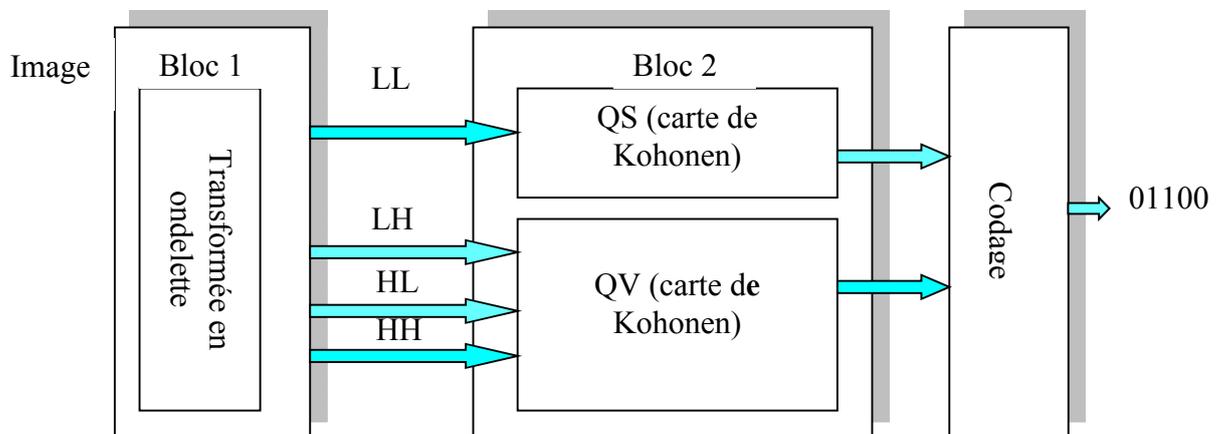


Figure 4.1. Schéma bloc de l'architecture proposée

4.2.1 Bloc 1 : partie prétraitement par ondelettes

Dans cette partie, la programmation a été faite à l'aide de MATLAB 7.7 (R2008b). Nous présentons les différentes étapes pour la compression des images médicales choisie dans [100]. Nous avons plusieurs images pour le test parmi lesquelles les trois images présentées respectivement (figure 4.2). Ce sont des images de 256* 256 de type Bmp.



Image 1

Image 2

Image 3

Figure 4.2. Représentation des images originales pour le test

4.2.1.1 Le choix des coefficients de filtre

Nous utilisons un banc de filtre 5/3 appelé filtres de LE Gall qui permet d'avoir une décomposition et une reconstruction sans perte d'information

Les coefficients de ce filtre sont donnés dans le tableau 4.1.

Tableau 4.1 : Coefficients du filtre de LE Gall

Filtre de décomposition	Filtre passe bas G0	$[-1/8 \ 2/8 \ 6/8 \ 2/8 \ -1/8]$
	Filtre passe haut H0	$[-1/2 \ 1 \ -1/2 \ 0 \ 0]$
Filtre de reconstruction	Filtre passe bas G1	$[1/2 \ 1 \ 1/2 \ 0 \ 0]$
	Filtre passe haut H1	$[-1/8 \ -2/8 \ 6/8 \ -2/8 \ -1/8]$

Ceci, sachant que nous avons introduit des zéros pour les filtres de degré inférieur (deux zéros pour les filtres de reconstruction), afin d'assurer une lecture commune de l'image pour tous les filtres en bloc de 5x5 pixel. En raison de simplification, on factorise par $1/8=2^{-3}$ et $1/2=2^{-1}$ afin d'éliminer la virgule de ces coefficients qui peuvent être considéré comme des décalages à droite dans un registre de taille fixe pour faciliter l'implémentation. On obtient les coefficients du tableau 4.2.

Tableau 4.2 : Coefficients du filtre après simplification

Filtre de décomposition	Filtre passe bas G0	$[-1 \ 2 \ 6 \ 2 \ -1]*1/8$
	Filtre passe haut H0	$[-1 \ 2 \ -1 \ 0 \ 0]*1/2$
Filtre de reconstruction	Filtre passe bas G1	$[1 \ 2 \ 1 \ 0 \ 0]*1/2$
	Filtre passe haut H1	$[-1 \ 2 \ 6 \ 2 \ -1]*1/8$

Après la décomposition des images par ondelettes, chaque image se divise en quatre parties, (figure 3).

- Partie approximation (bloc de 128*128 pixels).
- Partie détail (trois blocs de 128*128 pixels).

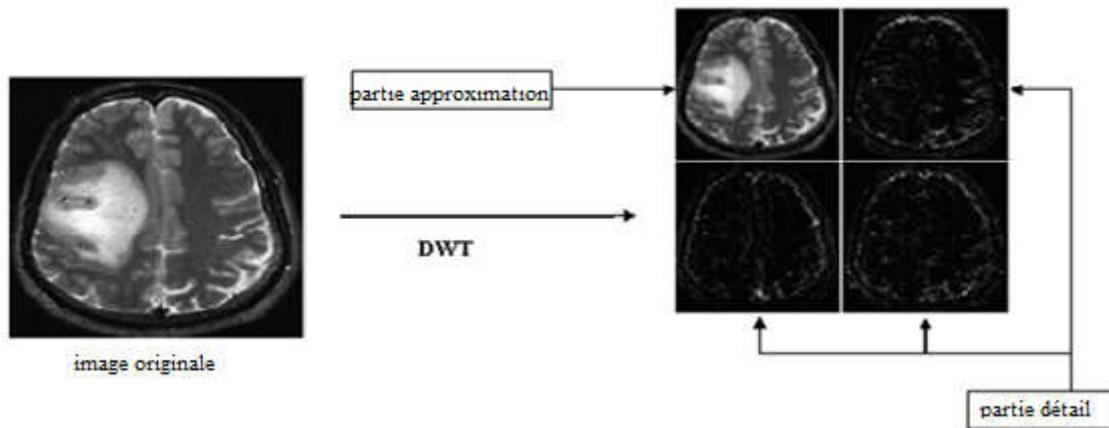


Figure 4.3 : Décomposition d'une image en quatre bandes

La figure 4.4 présente les résultats obtenus des images de la figure 4.2 après décomposition

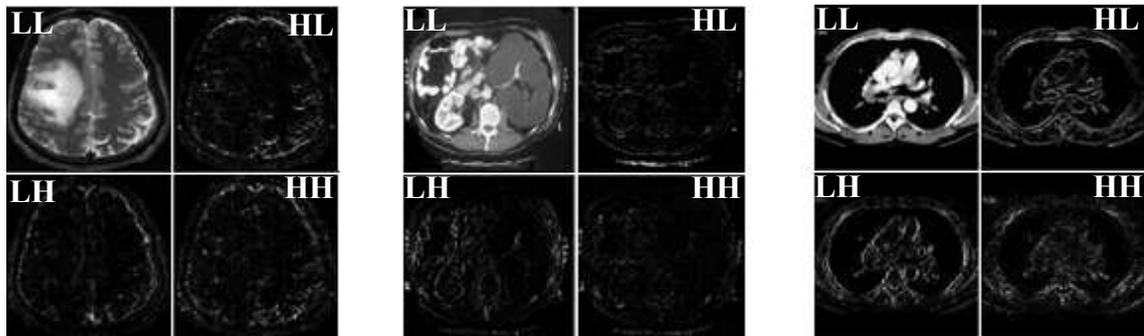


Figure 4.4. Résultat de décomposition des images en quatre bandes

4.2.2 Bloc 2 : La quantification

Dans ce travail, nous avons utilisés la bande LL qui est une approximation de l'image initiale et peut être considérée comme sa version sous-échantillonnée (à la fois horizontalement et verticalement) ou (coefficients des basses fréquences) qui sont concentrés pour une quantification scalaire. Les trois bandes LH, HL et HH (horizontale, verticale et diagonale) ou détails (coefficients hautes fréquences) seront exploitées pour la quantification vectorielle (figure 4.5) :

- Partie détail pour une Quantification Vectorielle.
- Partie approximation pour une Quantification Scalaire.

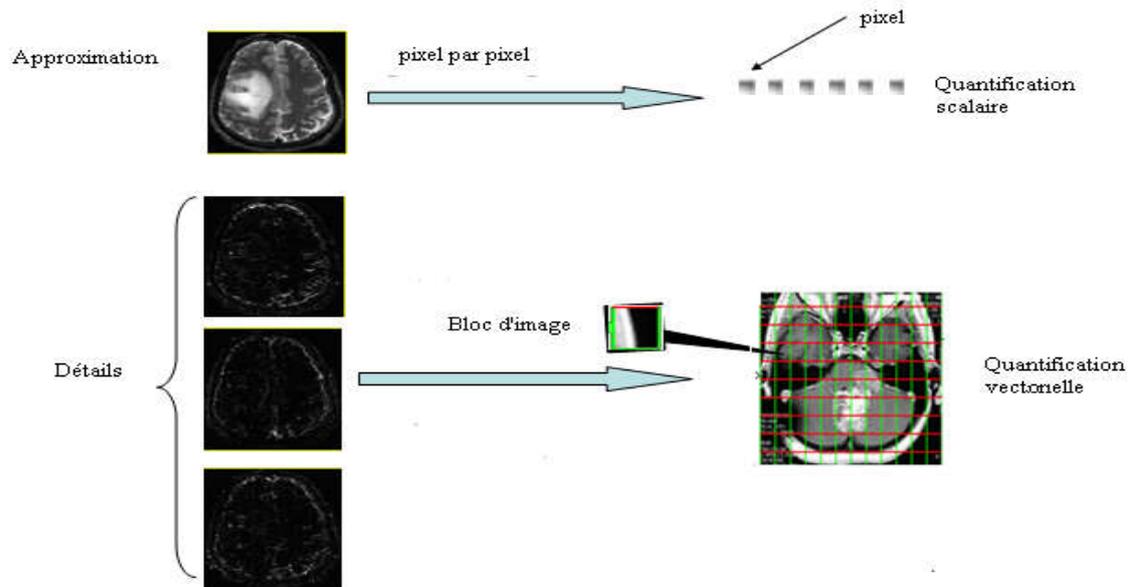


Figure 4.5. Procédé de quantification de la bande approximation et détails

L'utilisation du réseau de Kohonen est sensible au changement des paramètres internes tel que le facteur d'apprentissage, le nombre d'itérations, le nombre de neurones et le voisinage. La manière de fixer ces paramètres est critique pour la convergence du réseau. Néanmoins, ceci n'exclut en aucun cas sa fiabilité dans les domaines de la quantification.

Pour cela, il faut choisir la taille des blocs pour la quantification vectorielle et le nombre de neurones minimal pour l'implémentation et le nombre d'itérations.

L'algorithme appliqué est le même pour les deux parties. Cependant, elles se distinguent par la nature des données en entrée :

- Quantification vectorielle : l'image est découpée en blocs (vecteurs d'entrée) de $(2*2)$, $(4*4)$ ou $(8*8)$.
- Quantification scalaire : on prend l'image pixel par pixel.

A la fin de cet algorithme on obtiendra deux dictionnaires :

- Dictionnaire quantification scalaire de dimension $(1, M)$.
- Dictionnaire quantification vectorielle de dimension (N, M) .

Avec N : la taille des vecteurs d'entrée.

M : le nombre de neurones.

4.2.2.1. Détermination des différents paramètres du réseau de Kohonen

4.2.2.1.1. Test pour 16 neurones et 50 itérations

La détermination des paramètres repose sur un compromis entre le nombre de neurones et le nombre d'itérations. Nous avons fait plusieurs essais sur l'image 1 de la figure 4.2 ; sur le nombre de neurones et le nombre d'itérations. Pour la taille des blocs nous avons pris les trois cas suivants :

- Blocs 2*2 pixels.
- Blocs 4*4 pixels.
- Blocs 8*8 pixels.

Le tableau suivant présente les résultats des tests

Tableau 4.3 : Test pour 16 neurones et 50 itérations

Taille du bloc	PSNR	MQE	RC	NPB
2*2	30.9863	51.8138	63.7199	1
4*4	31.0119	51.5103	69.1619	0.25
8*8	30.9999	51.6524	70.5224	0.0625

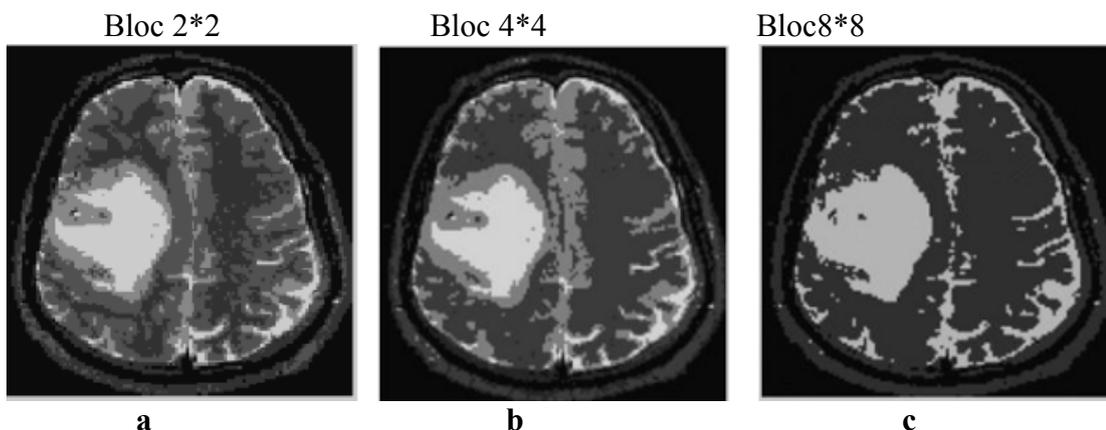


Figure 4.6. Reconstruction des images pour 16 neurones et 50 itérations.

4.2.2.1.2. Test pour 25 neurones et 50 itérations

La figure 4.7 présente la reconstruction de l'image 1 de la figure 4.2 pour différentes tailles des blocs ; pour 25 neurones et 50 itérations.

Tableau 4.4: Test pour 25 neurones et 50 itérations

block	PSNR	MQE	RC	NPB
2*2	31.0249	51.33564	68.75	1
4*4	31.0462	51.1048	73.4375	0.25
8*8	31.0513	51.0415	74.6094	0.0625

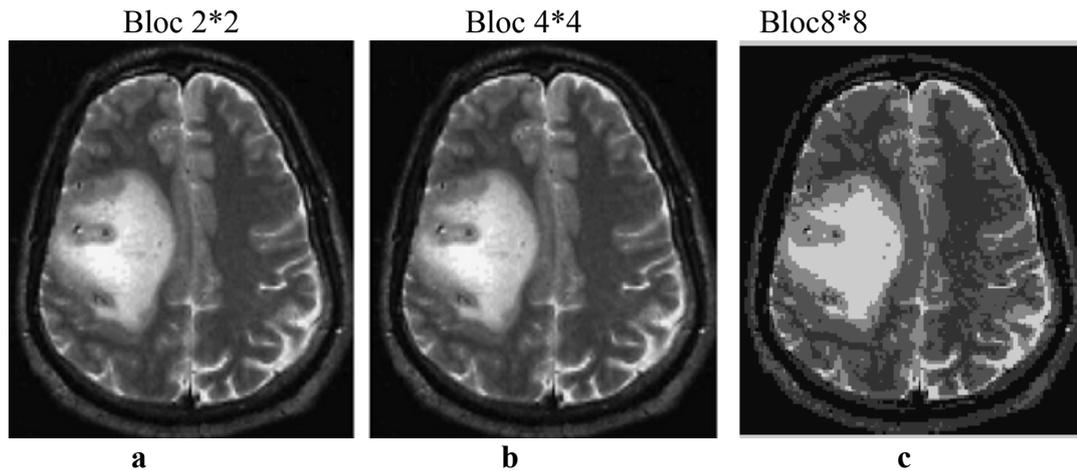


Figure 4.7. Reconstruction d'images pour 25 neurones et 50 itérations.

Dans les différents tests effectués, nous avons fixés en premier lieu le nombre de neurones à 16 et nous avons fait varier le nombre d'itérations. En augmentant le nombre d'itérations, la qualité de l'image est meilleure, puisque EQM diminue. Nous savons aussi qu'en augmentant le nombre de neurones, on obtient une bonne qualité de l'image. Afin d'avoir un compromis entre les deux paramètres (nombre de neurones et nombre d'itérations), nous avons augmenté le nombre de neurones à 25 et nous avons constaté que la qualité de l'image augmente et EQM diminue. En appliquant ces essais pour différentes tailles de blocs 2*2, 4*4 et 8*8, nous avons constaté que le taux de compression augmente avec la taille des blocs. Ce taux de compression atteint son maximum pour les blocs 8*8. Cependant, l'effet de bloc est visible et la qualité de l'image reconstruite est moyenne. En conclusion, le meilleur résultat est celui de la figure 4.7 b obtenu à partir des paramètres suivants :

- décomposition de l'image en bloc 4*4.
- Nombre de neurones 25 .
- Nombre d'itération 50.

4.2.3. Génération des dictionnaires

Nous avons appliqués l'algorithme de Kohonen sur un ensemble d'images avec les paramètres suivants : une taille de blocs (4x4), une carte de Kohonen 5×5 (25 neurones) et 50 itérations.

4.2.3.1. Calcul du facteur d'apprentissage $\beta_j(t)$

Il n'y a pas de règle générale quant à l'évolution des paramètres d'apprentissage, mais Kohonen propose une forme particulière de la fonction $\beta_j(t)$: C'est une fonction qui définit le pas d'apprentissage, elle décroît avec le temps. Dans notre implémentation nous avons choisi, $\beta_j(t)$ de la forme suivante :

$$\beta_j(t) \in [0,1] \quad \text{Avec } j \in V_{ind}(t) \quad (4.1)$$

$V_{ind}(t)$: L'ensemble des neurones qui constituent le voisinage du neurone gagnant.

$$\beta_j(t) = n_t(t) \times Gv_j(t) \quad (4.2)$$

Où les fonctions n_t et Gv_j gèrent respectivement l'étendue du voisinage. Elles sont définies comme suit :

$$n_t(t) = n_0 \times \left(1 - \frac{t}{t_{max}}\right) \quad (4.3)$$

$$Gv_j(t) = \max\left(0, 1 - \frac{dist_j}{gt}\right) \quad (4.4)$$

Avec
$$g(t) = g_0 \times \left(1 - \frac{t}{t_{max}}\right) \quad (4.6)$$

Le paramètre t_{max} étant le nombre total d'itérations.

$dist_j$ est la distance entre les coordonnées du neurone gagnant et ses voisinages.

$$dist = \|r_i - r_g\| \quad (4.7)$$

r_i : représente la position du $i^{ème}$ neurone sur la carte avec les coordonnées L_j, M_j .

r_g : représente la position du neurone gagnant sur la carte avec les coordonnées P, Q .

Les paramètres n_0 et g_0 définissent les voisinages et s'expriment par :

$$g_0 = 1 + \frac{(M + N)}{3} / 2 \quad (4.8)$$

$$n_0 = t_{max} \left(1 - \frac{1}{\sqrt{g_0}}\right) \quad (4.9)$$

4.2.3.2. Génération du dictionnaire pour la quantification scalaire

Le dictionnaire généré a une dimension de (1, 25) représenté à la figure 4.8.

1 : Le pixel d'entrée.

25 : le nombre de neurones.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 21 22 23 24 25

Figure 4.8 : Représentation du dictionnaire pour la quantification scalaire de dimension de (1, 25)

4.2.3.3. Dictionnaire pour la quantification vectorielle

Le dictionnaire généré a une dimension de (16,25) est représenté à la figure 4.9.

16 : taille du vecteur d'entrée.

25 : nombre de neurones.

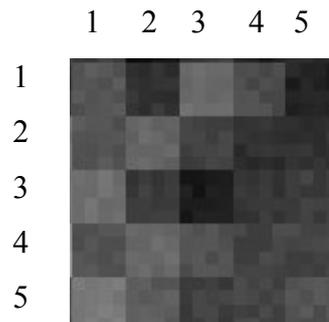


Figure 4.9 : Représentation du dictionnaire pour la quantification vectorielle de dimension de (16,25)

4.2.4. La décompression ou restitution des images

Les matrices d'indice obtenues sont codées par le codage arithmétique pour la transmission et la décompression des images. À la transmission, c'est uniquement l'adresse du code vecteur qui est transmise. Ensuite, le décodage consiste à récupérer ces adresses et les associer aux contenus des dictionnaires. Ceci dans le but de reconstituer les images des deux parties (détail et approximation), en appliquant l'ondelette inverse avec le filtre LE Gall.

La figure 4.10 représente la reconstruction des images prévues pour le test des images de la figure 4.2.

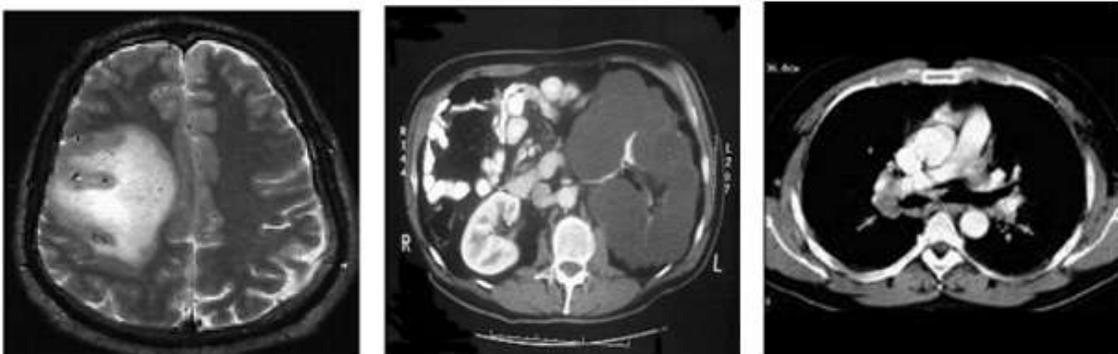


Figure 4.10. Reconstruction des images par ondelettes inverse

4.2.5. Calcul des performances

Nous avons développés notre méthode de compression sur les images du fichier Dicom avec sélection des régions d'intérêts (Region of Interest) [101] , et réalisé une interface ergonomique avec les exemples d'images présenter en Annexe 2.

Tableau 4.5. Calcul de performance pour les images Dicom

Type d'Image	Taille du fichier original	Taille du fichier compressé	Taux de compression	Rapport de compression	PSNR	SSIM
IRM de cerveau	130 Ko	9 Ko	93.08%	14.44 :1	27.48 dB	85.38%
IRM Abdomen	514 Ko	20 Ko	96.11%	25.7 :1	39.96dB	96.07%
IRM Abdomen	514 Ko	15 Ko	97.08%	34.26 :1	44.22 dB	97.09%

Les valeurs du PSNR et du SSIM sont assez importantes et prouve que la dégradation n'est pas très significatives. Plus la taille du fichier augmente, plus le taux de compression augmente.

PARTIE HARDWARE

4.3. Architecture globale :

Après la modélisation mathématique de l'algorithme de décomposition et de reconstruction de Stéphane Mallat par le mode de calcul en ligne et le développement des architectures correspondantes, une description en VHDL et plusieurs simulations sous Matlab Modelsim nous ont permis de juger de la faisabilité de ces architectures.

Le processus d'implémentation convoite certaines phases de vérification. En premier lieu, les architectures doivent être codées en langage VHDL. Par la suite, une phase de simulation va permettre une vérification fonctionnelle de ces dernières. Une fois la simulation fonctionnelle validée, une phase de synthèse va permettre de donner un rapport détaillé sur les ressources du circuit FPGA consommées et sur la fréquence maximale de fonctionnement. Enfin, la phase de placement et routage va éclater schématiquement la surface consommée, en deux parties constituant notre architecture et configurer le routage des interconnexions entre les différents blocs des ressources du circuit FPGA.

4.3.1 Bloc1 de décomposition par ondelettes :

La décomposition est basée sur l'opération de convolution et de décimation. Cette dernière a été accomplie d'une part dans la procédure de lecture de l'image avec des sauts des lignes et des colonnes. D'autre part, l'opération de convolution est réalisée à l'aide des filtres LE Gall.

L'application de mode de calcul en ligne nous ramène au :

- calcul de l'accumulateur partiel $L[j]$.
- calcul de résidu complet $H[j]$.
- calcul du chiffre résultat $Y[j]$.

Ceci revient donc à implémenter l'équation suivante :

$$H[j] = 2H[j-1] + L[j] - 2y_n[j-1] \quad (4.1)$$

$$\text{Avec: } L[j] = \sum_{i=0}^N h_i x_{n-i}[j] / \sum_{i=0}^N h_i \quad (4.2)$$

A partir des équations (4.1) et (4.2), on définit une nouvelle quantité $L_p[j]$ telle que:

$$L_p[j] = \sum_{i=0}^N h_i x_{n-i}[j] / \sum_{i=0}^N |h_i| - 2 \cdot y_n[j-1]$$

Des équations (3.39) et (3.41), on déduit la fonction de sélection donnée par $S(H[j])$ et définie par :

$$\begin{cases} H[j] = 2H[j-1] + L[j] - 2y_n[j-1] \\ y_n[j] = S(H[j]) \end{cases} \quad (4.3)$$

Avec
$$y_n[j] = \begin{cases} -1 & \text{si } -3/2 \leq H[j] < -1/2 \\ 0 & \text{si } -1/2 \leq H[j] < 1/2 \\ 1 & \text{si } 1/2 \leq H[j] < 3/2 \end{cases} \quad (4.4)$$

Nous pouvons diviser l'architecture en deux blocs principaux (figure 4.16) :

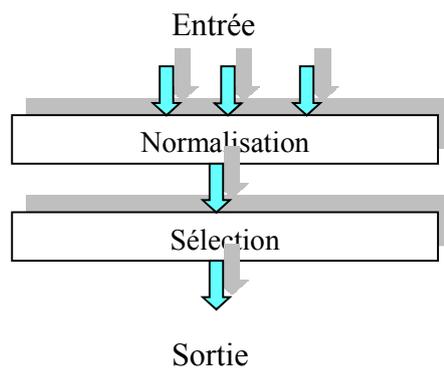


Figure 4.11: Structure générale de l'architecture du bloc 1

- Bloc de normalisation pour le calcul de $L_p[j]$
- Bloc de sélection pour le calcul de $H[j]$ et la génération du bit résultat.

L'architecture de la cellule de base en utilisant le mode en ligne est illustrée par la figure 4.12.

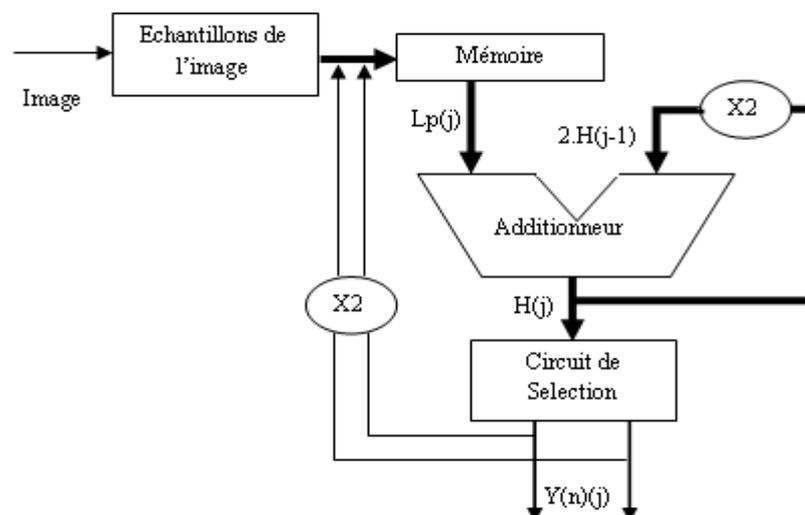


Figure 4.12 : Architecture de la cellule de base du bloc 1

4.3.1.1 Bloc de normalisation

Il s'agit de générer la valeur de $L_p[j]$. L'expression de cette dernière devient :

$$L_p[j] = [h_0 x_n[j] + h_1 x_{n-1}[j] + h_2 x_{n-2}[j] + h_3 x_{n-3}[j] + h_4 x_{n-4}[j]] / \sum_{i=0}^N |h_i| - 2 \cdot y_n[j-1] \quad (4.5)$$

Cette expression est difficile à implémenter sur un circuit FPGA, vue sa complexité.

En effet, pour calculer une valeur de $L_p[j]$, il faut réaliser 5 multiplications, 4 additions, une soustraction et une division. De ce fait, le procédé de calcul est très lent.

D'autre part, dans l'expression de l'accumulateur partiel $L_p[j]$, les coefficients h_i sont multipliés par des bits $x_{n-i}[j]$ variant entre -1, 0, ou 1. Cette opération est appelée multiplication bit-valeur. De plus, le terme $y_n[j-1]$ peut prendre aussi les valeurs -1, 0, 1 ; donc l'ensemble des valeurs de $L_p[j]$ est fini. Chaque valeur est calculée à l'aide d'une combinaison des bits $x_{n-i}[j]$ et $y_n[j-1]$. Ceci consiste à stocker les valeurs pré calculées de $L_p[j]$ dans la ROM et l'adresser avec $x_{n-i}[j]$ et $y_n[j-1]$. Ce principe est illustré dans le schéma suivant :

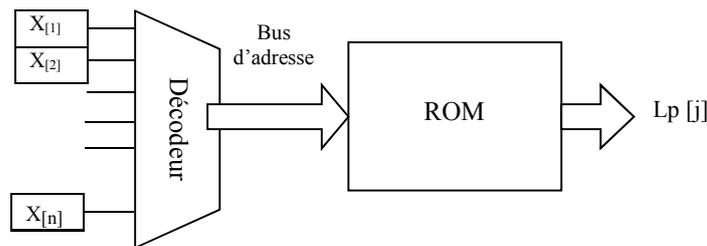


Figure 4.13: Utilisation d'une ROM lors du calcul de $L_p[j]$

Avec les différentes combinaisons d'entrées nous réalisons une table de vérité afin de définir l'état de sortie de chaque décodeur, en faisant correspondre chaque combinaison d'entrée à l'adresse de la valeur $L_p[j]$ pré calculée avec cette combinaison.

4.3.1.1.2 Taille des mémoires

Le nombre des valeurs pré calculées de $L_p[j]$ dépend du nombre des coefficients non nul. Pour les filtres à n coefficients, on a 3^{n+1} combinaison (en tenant compte des valeurs de $y_n[j-1]$). Si on prend en considération la symétrie de ces coefficients, ce nombre diminue, ce qui induit à une réduction de la taille des mémoires. Les différentes tailles sont indiquées dans le tableau ci-dessous :

Tableau 4.6 : Taille des mémoires pour chaque filtre

filtre	Nombre des valeurs de $Lp[j]$ pré calculées	Taille de mémoire	Taille de bus d'adresse
G0	75	128	7
H0	27	32	5
G1	27	32	5
H1	75	128	7

Les valeurs de $Lp[j]$ sont représentées en binaire et sont stockées en complément à 2.

Le nombre de valeur possible $Lp[j]$ nous informe sur sa taille en bits. Cette dernière doit répondre à la précision voulue. Donc, il faut choisir une taille qui permet de représenter toutes les valeurs possibles de $Lp[j]$ et d'avoir des résultats satisfaisant en termes de qualité.

Dans notre cas, les différentes tailles sont citées ci-dessous :

Tableau 4.7 : Taille des résidus partiels $Lp[j]$

Filtre	Taille de $Lp[j]$
G0, H1	15 bit
G1, H0	10 bit

4.3.1.2 Bloc de sélection :

On rappelle que ce bloc réalise les tâches suivantes :

- Evaluation du résidu complet $H(j)$.
- Evaluation de l'intervalle de l'appartenance de $H(j)$.
- Génération des bits résultats.

L'expression de $H(j)$ contient deux opérations, une addition avec récursivité et une multiplication par 2. La structure de ce bloc fait appel à un additionneur tel que le résultat est rebouclé vers l'une de ces entrées afin de réaliser une addition avec $H(j-1)$ et l'autre entrée est connectée au bus de donnée qui porte la valeur de $Lp[j]$. La multiplication par 2 est un simple décalage d'un bit vers la gauche.

4.3.1.2.1 Additionneur à propagation de retenue.

Il existe plusieurs circuits qui réalise l'opération d'addition, chacun possède des avantages et des inconvénients. Le choix dépend de l'application envisagée.

Dans le cas d'un additionneur à propagation de retenue, l'opération s'effectue avec des chiffres du poids faible vers le poids fort, en calculant à chaque fois la retenue pour l'additionner avec les chiffre de poids suivant. L'avantage de cette méthode vient du faite qu'elle utilise des simples fonctions logiques, donc facile à implémenter sur un FPGA.

De plus, ce dernier possède des canaux spéciaux pour la propagation de retenue ce qui va augmenter la rapidité de calcul. De ce fait, nous avons choisi ce circuit pour notre architecture.

4.3.1.2.2 Circuit de sélection.

La valeur de $H[j]$ est comprise entre $-3/2$ et $3/2$ donc la détermination de l'intervalle d'appartenance de cette dernière est assurée avec seulement 3 bit comme l'indique le tableau suivant.

Tableau 4.8 : Les Bits de sélection

Bits de H[J]	Signification
H_s	bit signe
H_{-1}	bit de poids « -1 »
H_0	bit de poids « 0 »

Nous avons fixé 3 bits après la virgule pour la valeur de $H[j]$, le MSB représente le bit signe et les deux autres représentent successivement le bit de poids 2^1 et 2^0 . Comme $H[j]$ est compris entre $-3/2$ et $3/2$, donc le bit de poids 2^1 est toujours égal à zéro ; par conséquent il sera inutilisé dans l'architecture (aucune connexion). La sélection se fait de la manière suivante :

Tableau 4.9: Fonction de sélection

H_s	H_0	H_{-1}	Y	y_nMsb	y_nLSB
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	-	-	-
1	0	0	-	-	-
1	0	1	-1	1	1
1	1	0	-1	1	1
1	1	1	0	0	0

Après simplification, on obtient le circuit suivant :

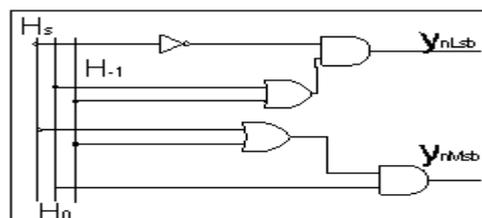


Figure 4.14: Schéma logique du circuit de sélection

4.3.2. Bloc 2 : Implémentation du quantificateur vectorielle et du quantificateur scalaire par l'algorithme de Kohonen

Dans cette partie, nous présentons l'implémentation du quantificateur vectoriel et du quantificateur scalaire par l'algorithme de Kohonen, sur un circuit FPGA, en se basant sur les résultats des paramètres de performances issus de l'implémentation software sous Matlab.

L'architecture proposée est composée de deux parties :

- Partie pour la quantification scalaire, illustrée sur la figure 4.15.
- Partie pour la quantification vectorielle, montrée sur la figure 4.29

4.3.2.1 Implémentation du quantificateur scalaire QS

Cette partie nous permet l'exécution de l'algorithme de Kohonen pour une quantification scalaire pour la partie approximation figure 4.15.

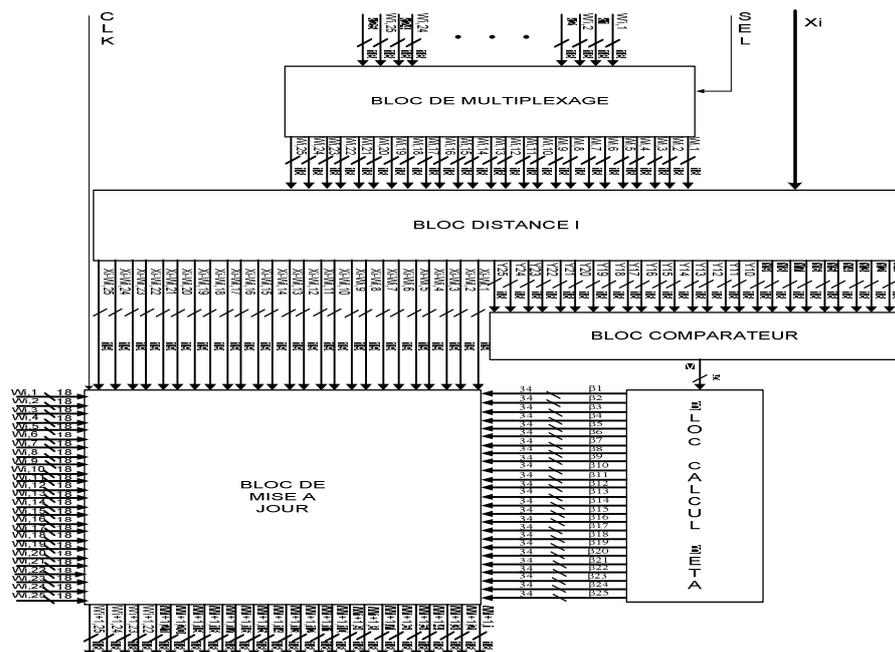


Figure 4.15. L'architecture globale de la quantification scalaire.

Notre architecture est composée des blocs suivants :

- Bloc de multiplexage qui permet de sélectionner les poids synoptiques initiaux W_{ij} par des constantes K_j . Pour le reste du traitement, il laisse passer les nouveaux poids synoptiques $W_{(i+1),j}$.

- Bloc de calcul de la distance (I): pour le calcul de $S_j = (X_i - W_{i,j})$ et $Y_j = (X_i - W_{i,j})^2$.
- Bloc comparateur : il permet de détecter parmi les différents Y_j l'indice du neurone gagnant, ce dernier correspond à la valeur minimale de Y_j .
- Bloc de calcul du facteur d'apprentissage β .
- Bloc de mise à jour.

Ces deux derniers blocs permettent de déterminer les nouveaux poids synoptiques $W_{(i+1),j}$.

4.3.2.1.1 Bloc du multiplexage

Ce bloc contient 50 entrées et 25 sorties codées sur 18 bits et une entrée de sélection notée SEL. Il est constitué de 25 multiplexeurs dont chacun possède deux pins d'entrée (pin1, pin2), un pin de sélection et une sortie, la table de vérité de ces multiplexeurs est montrée sur le tableau 4.12.

Tableau 4.10: Table de vérité du multiplexeur

SEL	Sorties
0	Pin 1
1	Pin 2

Le principe de fonctionnement de ce bloc est comme suit :

Au début du processus, chaque multiplexeur reçoit sur sa pin1 la valeur initiale K_j de $W_{i,j}$ du tableau 4.11.

Pour le reste du traitement, chaque multiplexeur reçoit sur sa pin2 la boucle de retour qui porte les nouvelles valeurs de $W_{i,j}$. Le schéma de ce bloc est illustré sur la figure 4.16.

Tableau 4.11 : Les différentes constantes des poids synoptiques initiaux.

Constante K_j	Nombre en binaire	Constante K_j	Nombre en binaire
K1	00.1111001100111011	K14	00.1011110011111011
K2	00.0011101100101011	K15	00.0010110100011111
K3	00.1001101101011010	K16	00.0110011111011100
K4	00.0111110001101001	K17	00.1110111101111010
K5	00.1110010000101100	K18	00.1110101010111010
K6	00.1100001100011000	K19	00.0110100100000111
K7	00.0111010011011011	K20	00.1110010011000110
K8	00.0000010010111100	K21	00.0000111011010001
K9	00.1101001001000111	K22	00.0101101001010101
K10	00.0111000111011000	K23	00.1101000000101011
K11	00.1001110110001100	K24	00.0000001010000110
K12	00.1100101010111100	K25	00.0010001110001110

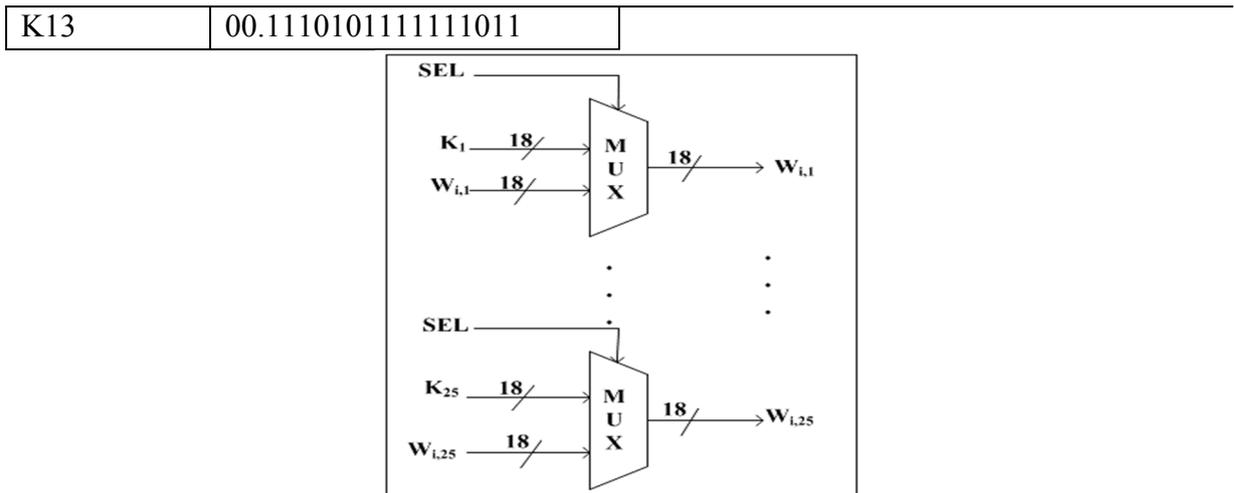


Figure 4.16: Bloc du multiplexage.

4.3.2.1.2 Bloc de calcul de la distance euclidienne

Ce bloc possède 26 entrées et 50 sorties :

- une entrée pour les données X_i codée sur 18 bits.
- 25 entrées pour les poids synoptiques $W_{i,j}$ qui proviennent du bloc précédent, codées sur 18 bits.
- 25 sorties pour $(X_i - W_{i,j})$, codées sur 19 bits.
- 25 sorties pour les Y_j , avec $Y_j = (X_i - W_{i,j})^2$, codées sur 18 bits.

Chaque cellule contient un soustracteur noté (SOUS) et un multiplieur noté (MULT) telle que montrée par la figure 4.17.

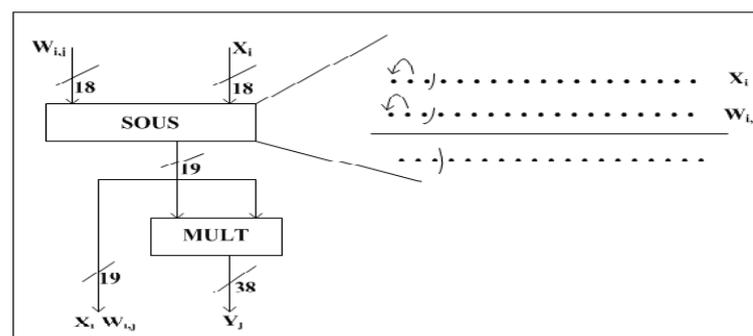


Figure 4.17 . Cellule de base du bloc de calcul de la distance Euclidienne.

Les entrées X_i et $W_{i,j}$ sont sur 18bits et leurs bits du poids fort (bit signe) est translaté au poids supérieur; ceci afin d'éviter le débordement de capacité. Le multiplieur permet d'élever au carré le résultat fourni par le soustracteur, $(X_i - W_{i,j})^2$ codé sur 38 bits. Ce bloc est constitué de 25 cellules identiques. La cellule élémentaire est montrée sur la figure 4.17.

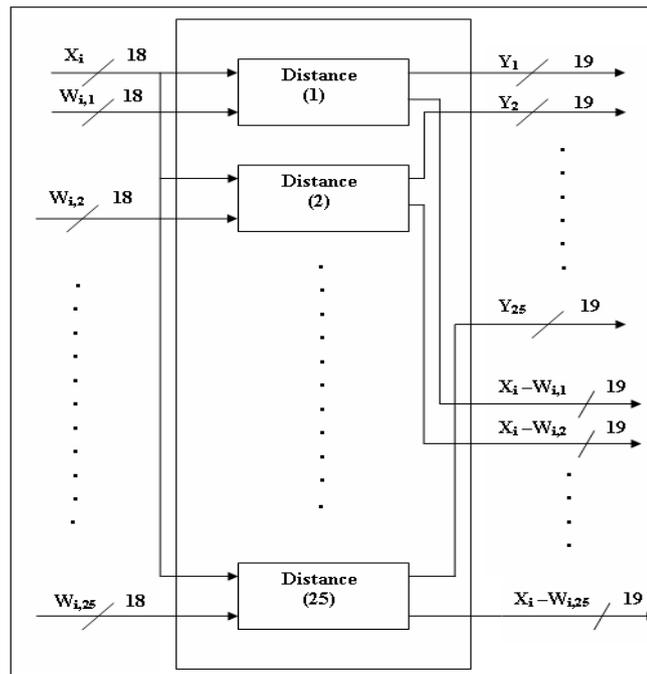


Figure 4.18 : Schéma synoptique du bloc de calcul de la distance Euclidienne.

4.3.2.1.3 Bloc comparateur

Ce bloc a pour tâche la détection de l'indice (j) de la valeur minimale parmi les 25 Y_j obtenus du bloc précédent. Ce bloc reçoit les 25 Y_j et leurs indices respectifs et génère en sortie un vecteur noté (ind). Il est composé de 4 modules notés, module A, module B, module C et module D. figure 4.19:

- 25 entrées des Y_j obtenus du bloc précédent, codées sur 19 bits.
- 25 entrées représentant respectivement les indices des Y_j ($j=1, \dots, 25$), codées sur 5 bits.
- Une sortie qui représente l'indice du $Y_{j,\min}$, codée sur 5 bits.

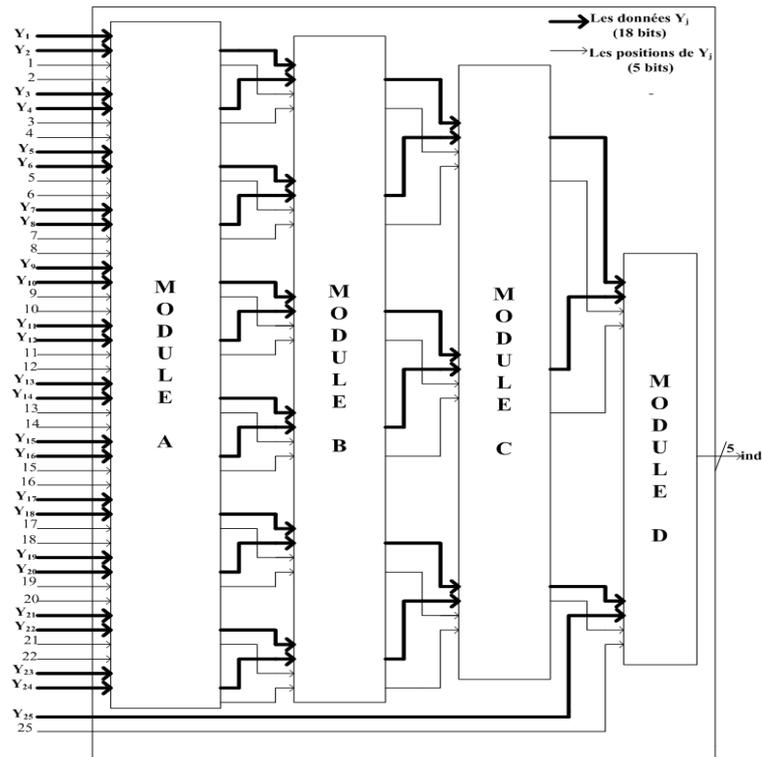


Figure 4.19 : Schéma synoptique du bloc comparateur.

4.3.2.1.3.1 Module A

Le module A possède 48 entrées, (24 Y_j , et leurs indices respectifs). Il est constitué de 12 cellules identiques. Chaque cellule contient un comparateur noté (COMP) et 2 multiplexeurs (MUX). La cellule élémentaire est représentée sur la figure 4.26.

Son principe de fonctionnement est le suivant : Le comparateur compare deux valeurs de Y_j qui se succèdent afin de trouver la valeur minimale entre celles-ci. Ensuite, nous avons deux multiplexeurs dont les pins de sélection sont commandées par la sortie du comparateur. Le 1^{er} MUX sélectionne la donnée, tandis que le 2^{ème} sélectionne sa position ; on obtient:

- 12 sorties pour les Y_j obtenues après comparaison.
- 12 sorties pour les indices.

4.3.2.1.3.2 Module B

Ce module a pour entrées les sorties du module A. Ce dernier est constitué de 6 cellules identiques aux précédentes. Son principe de fonctionnement reste le même. Dans ce cas on obtient 12 sorties.

4.3.2.1.3.3 Module C

Le principe de fonctionnement reste inchangé pour ce module. La cellule élémentaire est dupliquée en 3 fois. Il a l'entrée des sorties du module précédent et on obtient 6 sorties.

4.3.2.1.3.4 Module D

Ce module est constitué de la même cellule élémentaire dupliquée en 2, et d'une 3^{ème} qui se compose d'un comparateur et d'un MUX. A la sortie de la 3^{ème} cellule on obtient l'indice du $Y_{j,\min}$ noté (ind) .

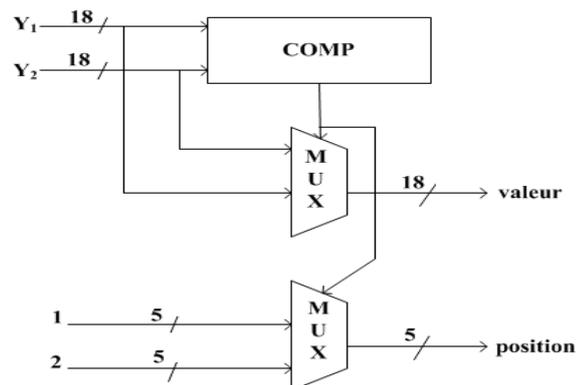


Figure 4.20 : Cellule de base pour les modules du comparateur.

4.3.2.1.4 Bloc de calcul du facteur d'apprentissage β

Ce bloc nous permet de calculer les différentes valeurs de bêta nécessaires à la mise à jour. Il possède une entrée pour l'indice (ind) qui provient du bloc précédent et génère en sortie 25 valeurs (β_j). Ce bloc est constitué de quatre modules : deux tables notées P et Q et de deux modules β_1 et Gv représentés en figure 4.21.

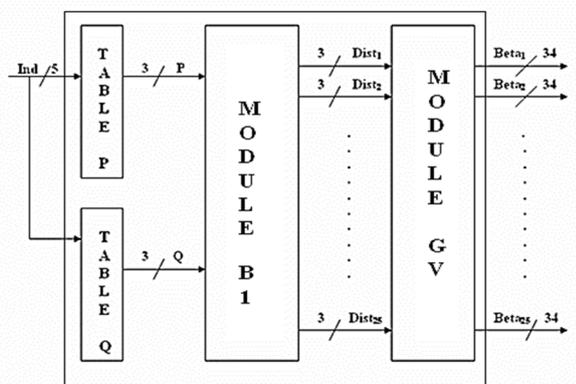


Figure 4.21. Schéma synoptique du bloc de calcul du facteur d'apprentissage β

Les différentes étapes pour le calcul du facteur d'apprentissage β .

Pour ind trouvé	(ind=1,...25)	
$P = \text{fix} ((\text{ind}-1) / 5) + 1$		
$Q = \text{ind}-5 * \text{fix} ((\text{ind}-1) / 5)$		
Pour j=1 a 25		
$M_j = \text{fix} ((j-1) / 5) + 1$		
$L_j = j-5 * \text{fix} ((j-1) / 5)$		
Distance $j = \max (L_j - P , M_j - Q)$		(5.6)
$Gv_j = \max (0, (1 - (\frac{Dist_j}{gt})))$		(5.7)
$\beta_j = n_t * Gv_j$		(5.8)

Avec $g_t = g_0 (1 - t/T_{\max})$ et $n_t = n_0 (1 - t/T_{\max})$

fix : permet de prendre la partie entière.

P, Q : elles dépendent de l'indice trouvé (ind).

L_j, M_j : elles dépendent de l'indice du neurone.

Tableau 4.12 : Les différentes valeurs pré-calculées de la distance j.

Ind	P _i	Q _i	Ind	P _i	Q _i
00001	001	001	01110	011	100
00010	001	010	01111	011	101
00011	001	011	10000	100	001
00100	001	100	10001	100	010
00101	001	101	10010	100	011
00110	010	001	10011	100	100
00111	010	010	10100	100	101
01000	010	011	10101	101	001
01001	010	100	10110	101	010
01010	010	101	10111	101	011
01011	011	001	11000	101	100
01100	011	010	01110	101	101
01101	011	011			

Les tables P et Q possèdent une entrée d'adressage pour l'indice ind codé sur 5 bits et une sortie pour les données pré-calculées de P et Q codée sur 3 bits.

Tableau 4.13 : Les différentes valeurs pré-calculées de L_j et M_j .

J	L_j	M_j	J	L_j	M_j
00001	001	001	01110	011	100
00010	001	010	01111	011	101
00011	001	011	10000	100	001
00100	001	100	10001	100	010
00101	001	101	10010	100	011
00110	010	001	10011	100	100
00111	010	010	10100	100	101
01000	010	011	10101	101	001
01001	010	100	10110	101	010
01010	010	101	10111	101	011
01011	011	001	11000	101	100
01100	011	010	01110	101	101
01101	011	011			

4.3.2.1.4.1 Module B1

Ce module permet l'exécution de l'équation (4.6). Il possède 2 entrées et 25 sorties, codées sur 3 bits. Ce module est composé de 2 sous modules notés (L-M) et MAX. Le schéma synoptique de ce module est représenté sur la figure 4.22.

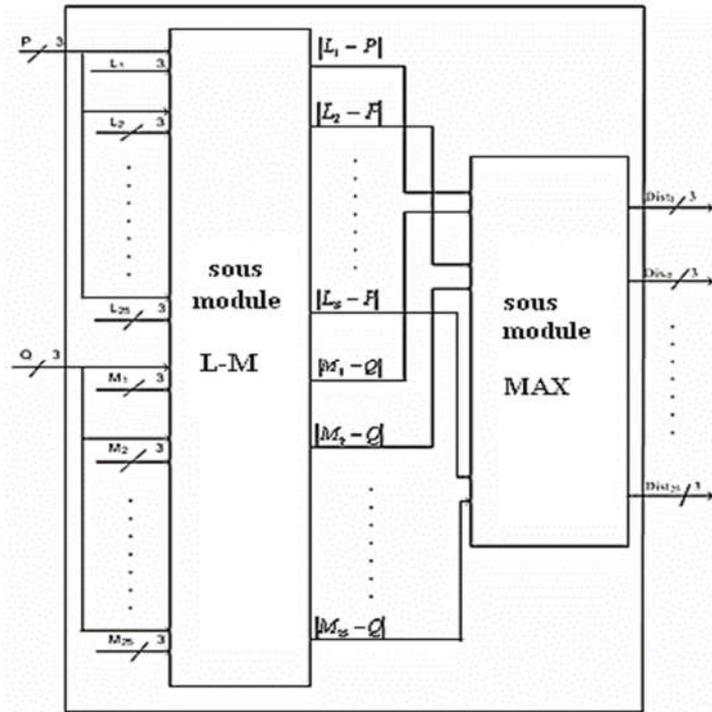


Figure 4.22. Schéma du module B1.

4.3.2.1.4.2 Sous module (L-M)

Ce sous module permet de soustraire respectivement P et Q de L_j et M_j respectivement, et d'extraire par la suite les valeurs absolues des différents résultats trouvés. Il contient 52 entrées :

2 entrées pour P et Q, 25 pour les L_j et 25 pour les M_j . (L_j et M_j sont des constantes dont les valeurs sont représentées dans le tableau 4.9). Il génère 50 sorties, 25 pour les $|L_j - P|$ et 25 pour les $|M_j - Q|$.

Ce sous module est constitué de 50 cellules identiques. La cellule élémentaire est représentée par la figure 4.23.

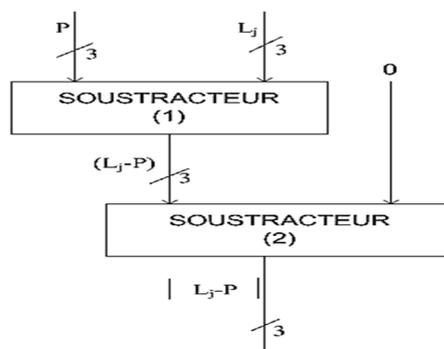


Figure 4.23 . Cellule de base du sous module (L-M).

Chaque cellule contient deux soustracteurs. Le soustracteur (1) permet de calculer $(L_j - P)$. Le second est utilisé pour la génération de la valeur absolue. Si $(L_j - P)$ est positif, il est injecté directement à la sortie du sous module ; sinon, on utilise le 2^{ème} soustracteur, dont la pin1 reçoit zéro et la deuxième pin reçoit la valeur obtenue du soustracteur 1, soit $(0 - (L_j - P))$. La sélection de $(L_j - P)$ ou du résultat du second soustracteur en sortie est commandée par le signe de $(L_j - P)$.

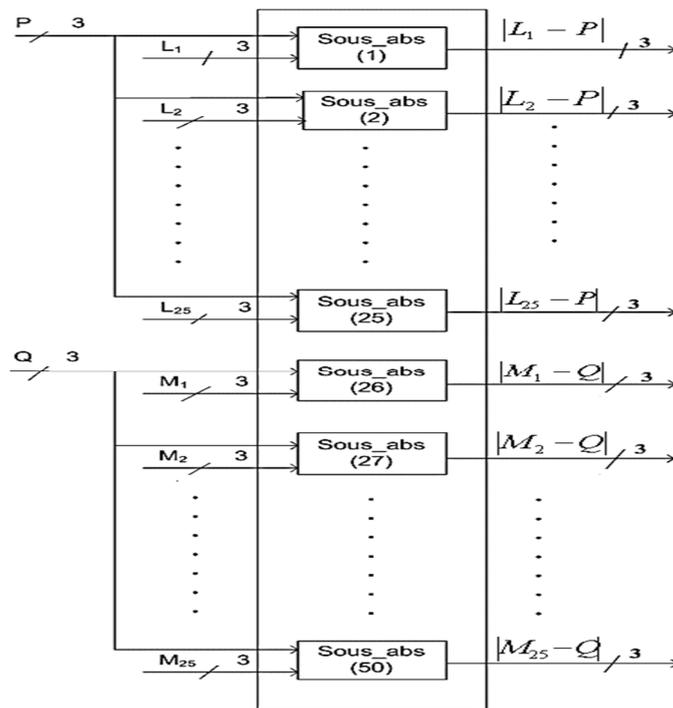


Figure 4.24. Schéma du sous module (L-M).

4.3.2.4.3 Sous module MAX

Ce sous module permet de trouver le maximum entre $|L_j - P|$ et $|M_j - Q|$ qui proviennent du bloc précédent ; il contient 50 entrées et 25 sorties :

- 25 sorties pour le calcul de l'équation (4.6).

Ce bloc est constitué de 25 cellules identiques. Chaque cellule contient un comparateur et un multiplexeur ; son fonctionnement de base est identique à celui du bloc comparateur.

La cellule de base et le schéma du sous module MAX sont représentés respectivement sur les figures 4.25 et 4.26.

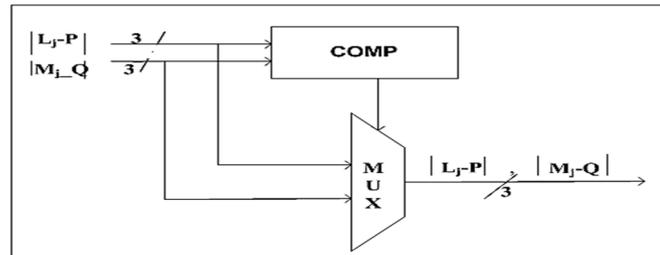


Figure 4.25: Cellule de base du sousmodule MAX.

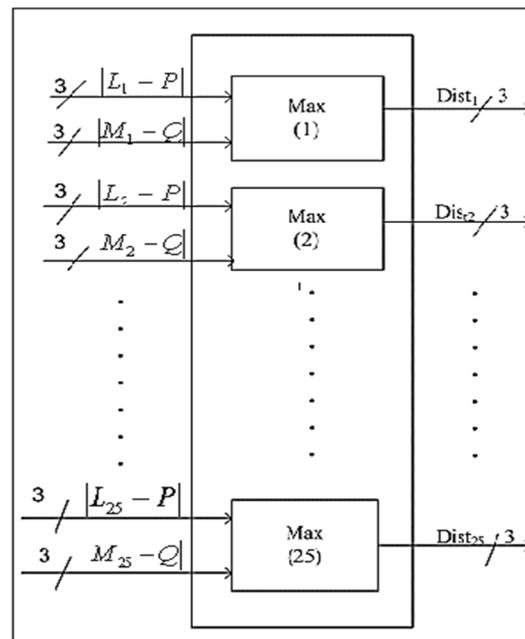


Figure 4.26 : Schéma du sous module MAX.

4.3.2.1.4.4 Module de calcul de G_v

Ce module permet l'exécution de l'équation (4.7) et (4.8). Il possède 25 entrées associées aux valeurs des distances notées ($dist_j$) calculer par le module précédent et fourni 25 valeurs de β qui sont codées sur 39 bits. Pour l'exécution de ce module, nous avons utilisé un programme VHDL qui calcule ces équations.

4.3.2.1.5 Bloc mise à jour

Ce bloc permet la génération des nouveaux points synoptiques $W_{(i+1),j}$, il contient 75 entrées et 25 sorties :

- 25 entrées pour les poids synoptiques initiaux (K_j), codées sur 18 bits.
- 25 entrées pour les ($X_i - W_{i,j}$) qui proviennent du bloc distance (I), codées sur 19 bits.
- 25 entrées pour les bêtas provenant du bloc précédent, codées sur 34 bits.
- 25 sorties pour les nouveaux poids synoptiques $W_{(i+1),j}$ codées sur 18 bits.

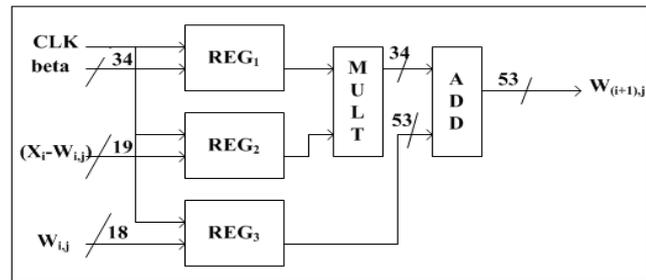


Figure 4.27 : Cellule de base du bloc mise à jour.

Les valeurs de β , les $(X_i - W_{i,j})$ et des poids synoptiques $W_{i,j}$ sont synchronisés par une horloge CLK. Pour cela nous avons introduit des registres. En premier lieu, ce bloc fait la multiplication entre le β et $(X_i - W_{i,j})$, à sa sortie il donne un résultat sur 53 bits. Avant l'addition avec les $W_{i,j}$, il est concaténé avec des zéros pour avoir la même taille que $\beta * (X_i - W_{i,j})$ qui est sur 53 bits. Ainsi, le résultat obtenu est sur 53 bits ; figure 4.27

Comme les $W_{i,j}$ sont représentés dès le début par un format fixe (18 bits), le résultat fourni par l'additionneur $W_{(i+1),j}$ doit être tronqué pour qu'il soit de la même taille que les $W_{i,j}$ initiaux . Le schéma de ce bloc est représenté sur la figure 4.28.

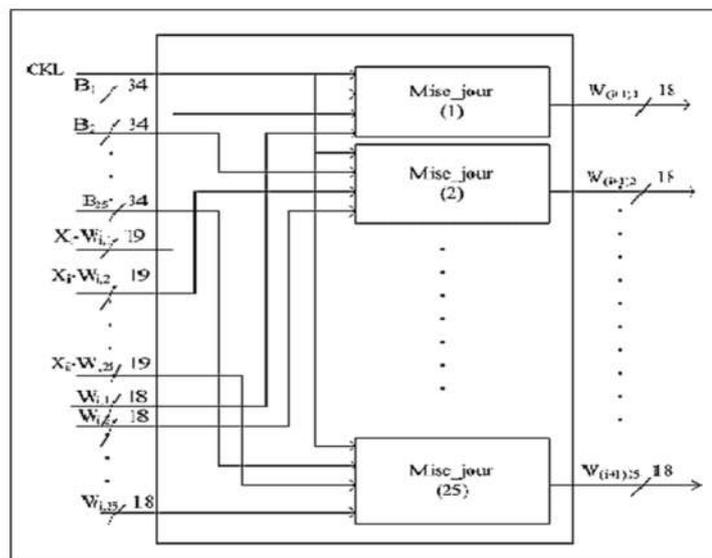


Figure 4.28: Schéma du bloc mise à jour.

4.3.2.2 Implémentation de la quantification vectorielle (QV)

Cette partie permet l'exécution de l'algorithme de Kohonen pour une quantification vectorielle sur la partie détail. L'architecture est identique à celle de la quantification scalaire, sauf qu'en entrée un vecteur de 16 valeurs, rentrent séquentiellement figure 4.29.

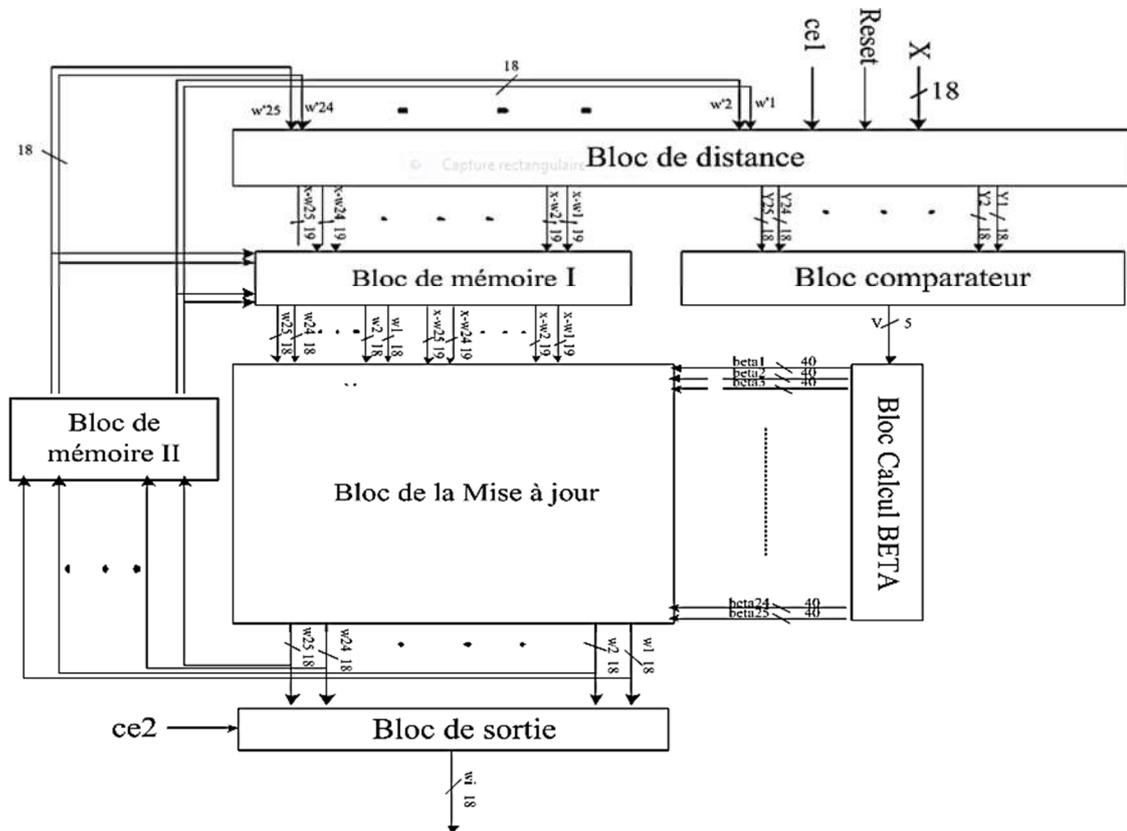


Figure 4.29: L'architecture globale de la quantification vectorielle.

Cette architecture reçoit en entrées :

- Les poids synoptiques $W_{i,j}$ provenant des RAMs.
- Une entrée pour les données X_i qui représente les vecteurs de la partie détail de l'image traitée.
- Les signaux de commandes WE1 et WE2 pour la sélection des RAMs en mode d'écriture ou de lecture.
- Les signaux CE et Reset utilisés respectivement pour le contrôle du Tristaite et l'initialisation de l'accumulateur.
- Une horloge CLK pour synchroniser les registres.

Cette architecture fournit en sortie, les nouveaux poids synoptiques, $W_{(i+1),j}$.

L'architecture est constituée de plusieurs blocs :

1- Bloc de calcul de distance II, permet de calculer les équations suivantes :

$$S_j = (X_i - W_{i,j}) \quad (4.9)$$

$$Y_{i,j} = \sum_{i=1}^{16} (X_i - W_{i,j})^2 \quad (4.10)$$

2- Bloc mémoire(1) : permet de stocker les poids synoptiques et les valeurs de la soustraction $S_j = (X_i - W_{i,j})$ pour les utiliser ultérieurement.

3- Bloc comparateur : recherche l'indice du neurone gagnant noté ind.

4- Bloc de calcul β et Bloc de la mise à jour permettent de calculer les nouveaux poids synoptiques $W_{(i+1),j}$.

5- Bloc mémoire (2) : il permet de stocker les nouveaux poids synoptiques.

4.3.2.2.1 Bloc de calcul de distance II

Ce bloc permet de calculer les équations (4.9) et (4.10) ; il est constitué de 25 cellules identiques, dans chaque cellule de base on retrouve deux composants supplémentaires : un accumulateur et un tristate. L'accumulateur est utilisé dans le but de

calculer la somme : $Y_{i,j} = \sum_{i=1}^{16} (X_i - W_{i,j})^2 \quad (4.11)$

Par contre le tristate permet d'extraire uniquement le 16^{ème} résultat de l'accumulateur ($Y_{16,j}$). Dans ce bloc, on retrouve 26 entrées codées sur 18 bits, une entrée pour les données X_i et 25 pour les poids synaptiques $W_{i,j}$ qui proviennent du bloc mémoire (2). Chaque entrée de $W_{i,j}$ possède une succession de 16 valeurs. Rajoutant à ces entrées des signaux de commande RESET et CE respectivement utilisés pour l'initialisation de l'accumulateur et le contrôle du tristate. En sorties, on retrouve 50 signaux (25 sorties pour Y_j et 25 pour $(X_i - W_{i,j})$ codées respectivement sur 18 et 19 bits). Pour chaque sortie de $(X_i - W_{i,j})$, on retrouve une succession de 16 valeurs .

L'accumulateur possède un signal de mise à zéro (reset) qui est actif à chaque fin du traitement de la 16^{ème} valeur. A la sortie du tristate les résultats sont sur 38 bits, et vont subir une troncation pour avoir 18 bits. Le format en virgule fixe de ce résultat est de 5 bits pour la partie entière, et 13 bits pour la partie fractionnelle. Le circuit d'une cellule de base est représenté sur la figure 4.30.

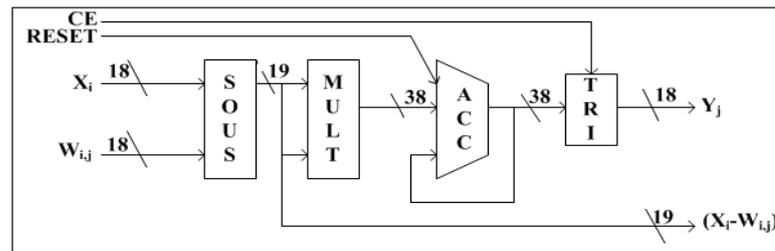


Figure 4.30: Cellule de base du bloc de calcul de distance (II).

4.3.2.2.2 Bloc comparateur et bloc de calcul du facteur d'apprentissage β

Ces deux blocs sont identiques à ceux utilisés dans la quantification scalaire.

4.3.2.2.3 Bloc mémoire (1)

Ce bloc est constitué de deux modules et d'un compteur. Ces derniers permettent le stockage respectif des valeurs intermédiaires W_{ij} provenant du bloc mémoire(2) et $(X_i - W_{ij})$ généré par le bloc distance 2. Puisque on dispose de $16 \cdot 25 \cdot W_{ij}$, il s'agit de calculer $16 \cdot 25 (X_i - W_{ij})$ à chaque traitement, d'où il est nécessaire d'utiliser 25 mémoires RAMs dans chaque module. Ce qui engendre une totalité de 50 mémoires, leurs tailles sont définies comme suit :

Pour le 1^{er} module, cette dernière est de $2^4 \cdot 18$ bits, (2^4 : le nombre d'adresse correspond à 16 valeurs de W_{ij} , 18 bits : la taille des W_{ij}). La taille des mémoires utilisées dans le second module est de $2^4 \cdot 19$ bits.

Ces RAMs peuvent être en mode lecture ($WE=0$) ou en mode d'écriture ($WE=1$).

Il est à noter que dans chaque mémoire, il est question de stocker pendant la phase d'écriture séquentiellement 16 données, et les récupérer de la même manière durant la phase de lecture. On représente sur la figure 4.31 une mémoire élémentaire du 1^{er} module.

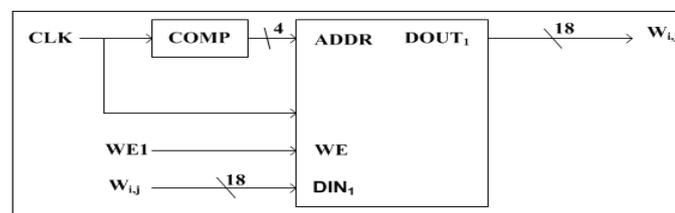


Figure 4.31. Mémoire élémentaire de 1^{er} module.

Le schéma synoptique de ce bloc est représenté sur la figure 4.32.

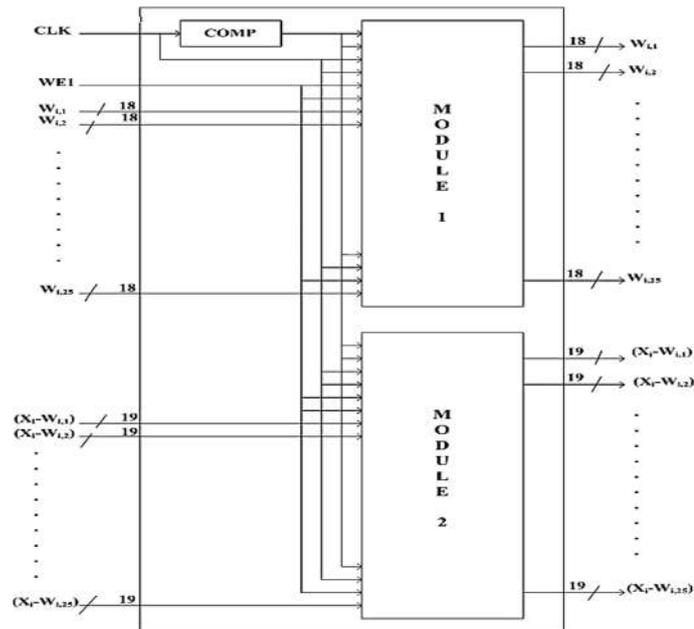


Figure 4.32 . Schéma synoptique du Bloc mémoire (1).

4.3.2.2.4 Bloc mise à jour 2

Ce bloc permet l'exécution de l'équation (4.12).

$$W_{ij}(t+1) = W_{ij}(t) + \beta_j (X_i - W_{ij}(t)). \quad (4.12)$$

Il contient 75 entrées et 25 sorties :

- 25 entrées pour les bêtas qui proviennent du bloc de calcul de bêta, codées sur 34 bits.
- 25 entrées pour les poids synoptiques initiaux $W_{i,j}$ qui proviennent du bloc mémoire (2), codées sur 18 bits.
- 25 entrées pour les $(X_i - W_{i,j})$ qui proviennent du bloc mémoire (1), codées sur 19 bits.
- 25 sorties pour les nouveaux poids synoptiques $W_{(i+1),j}$, codées sur 18 bits.

Pour ces trois derniers, chaque entrée et sortie est une succession de 16 valeurs. La figure 4.33 représente la cellule élémentaire de ce bloc.

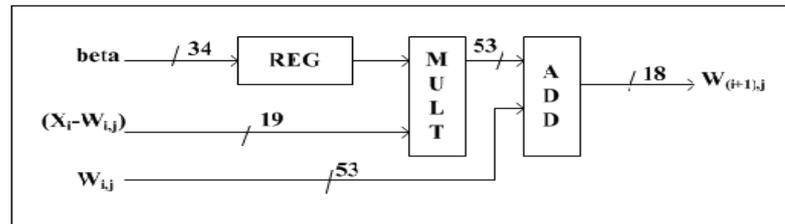


Figure 4.33 : Cellule de base du bloc mise à jour.

4.3.2.2.5 Bloc mémoire (2)

Ce bloc a le même fonctionnement que le bloc mémoire (1), il possède 25 entrées et 25 sorties pour les poids synoptiques $W_{i,j}$, chaque entrée et sortie possède 16 valeurs codées sur 18 bits, plus un signal de commande WE2 qui assure la sélection de la mémoire en mode de lecture ou d'écriture. Ce bloc est constitué de 25 cellules élémentaires dont le schéma synoptique est montré sur la figure 4.34.

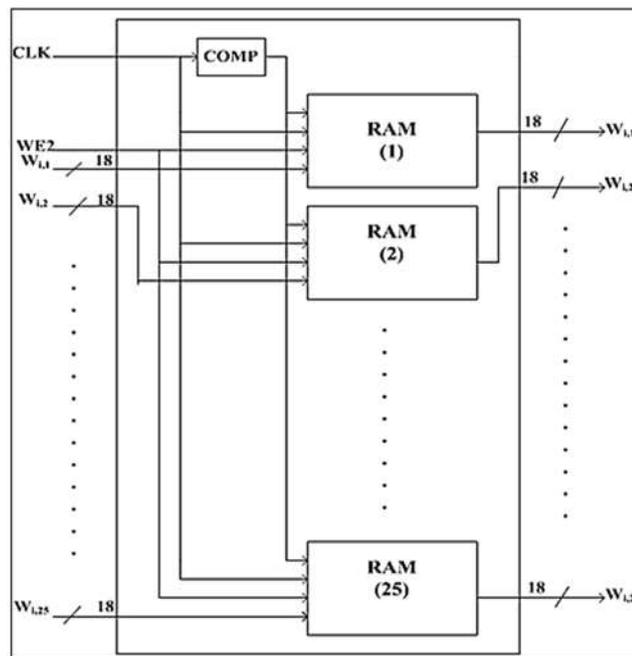


Figure 4.34. Schéma synoptique du Bloc mémoire (2).

4.4. Résultats d'implémentation et de simulation sur circuit Virtex II

L'implémentation de l'architecture proposée est réalisée sur le circuit Virtex II circuits 2v8000ff1152-5. Les performances en taux d'occupation et délais d'exécution sont illustrées sur le tableau 4.14.

Tableau 4.14 : Les performances en taux d'occupation sur FPGA.

Circuit: 2v8000ff1152-5

Bloc 1 : Décomposition ondelette			
Number of Slices:	2191	out of 3072	71 %
Number of Slice Flip Flops:	189	out of 6144	3 %
Number of 4 input LUTs:	4143	out of 6144	67 %
Number of bonded IOBs:	61	out of 172	35 %
Number of GCLKs:	1	out of 16	6 %
Bloc2 : Quantification scalaire			
Number of Slices:	26972	out of 46592	57 %
Number of Slice Flip Flops:	2939	out of 93184	3 %
Number of 4 input LUTs:	49879	out of 93184	53 %
Number of bonded IOBs:	479	out of 1108	43 %
Number of GCLKs:	3	out of 16	18 %
Quantification vectorielle			
Number of Slices:	14806	out of 46592	31 %
Number of Slice Flip Flops:	2410	out of 93184	2 %
Number of 4 input LUTs:	27259	out of 93184	29 %
Number of bonded IOBs:	474	out of 1108	42 %
Number of BRAMs:	75	out of 168	44 %
Number of GCLKs:	2	out of 16	12 %

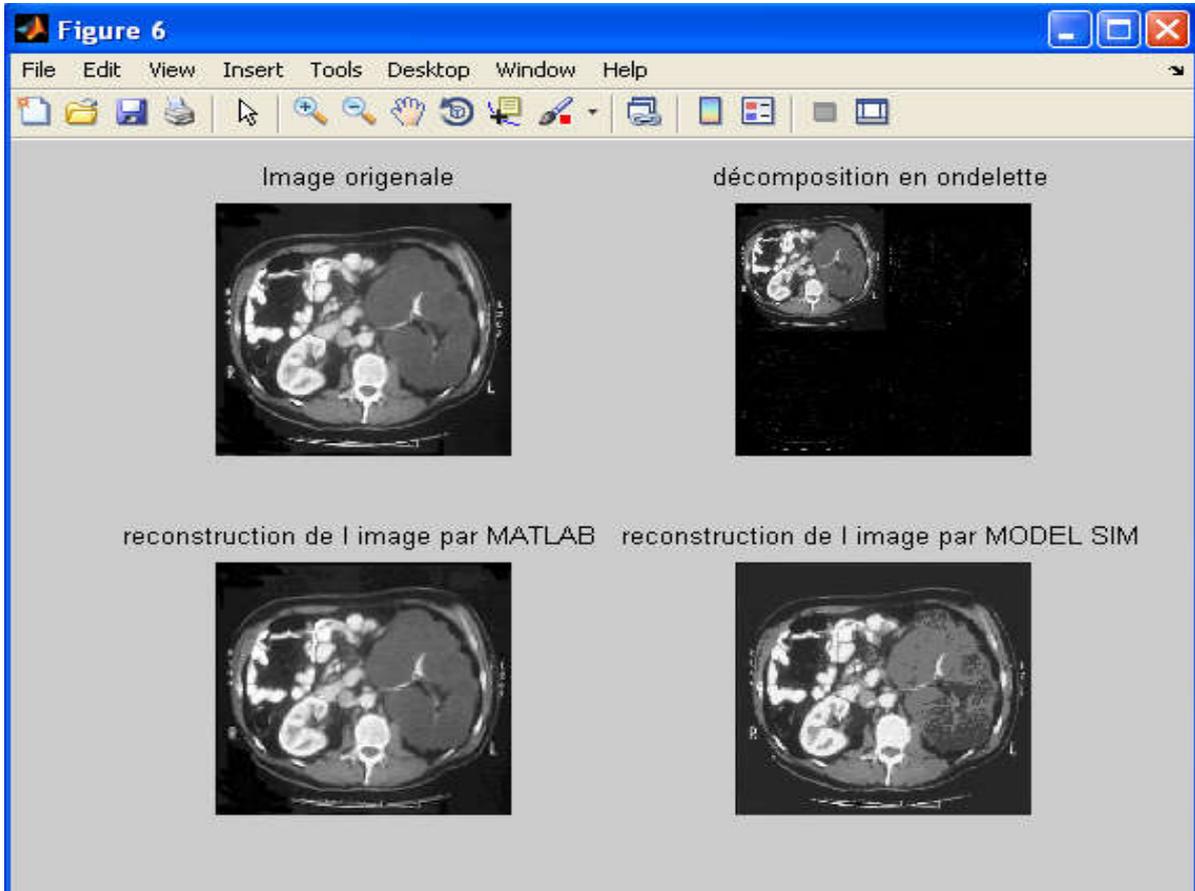
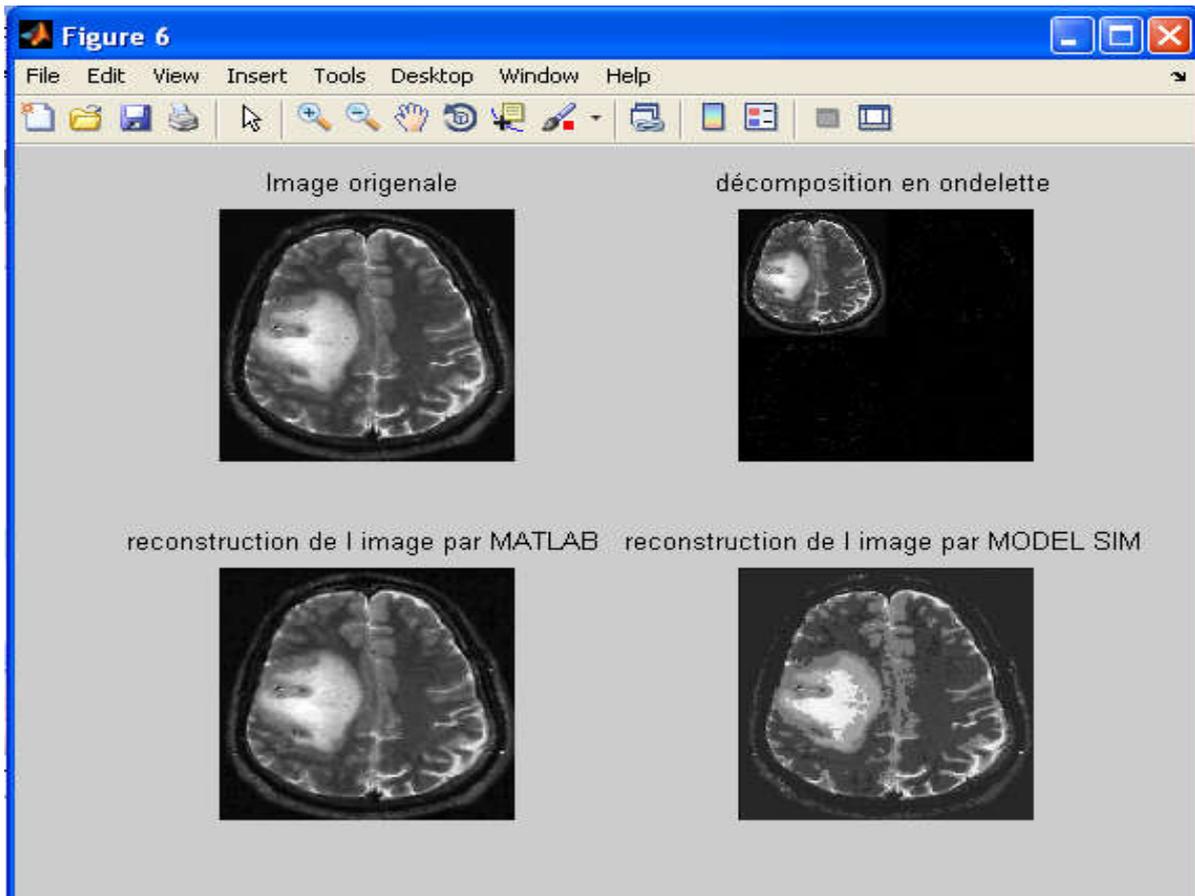
Tableau 4.15 : Les performances fréquentiels sur FPGA.

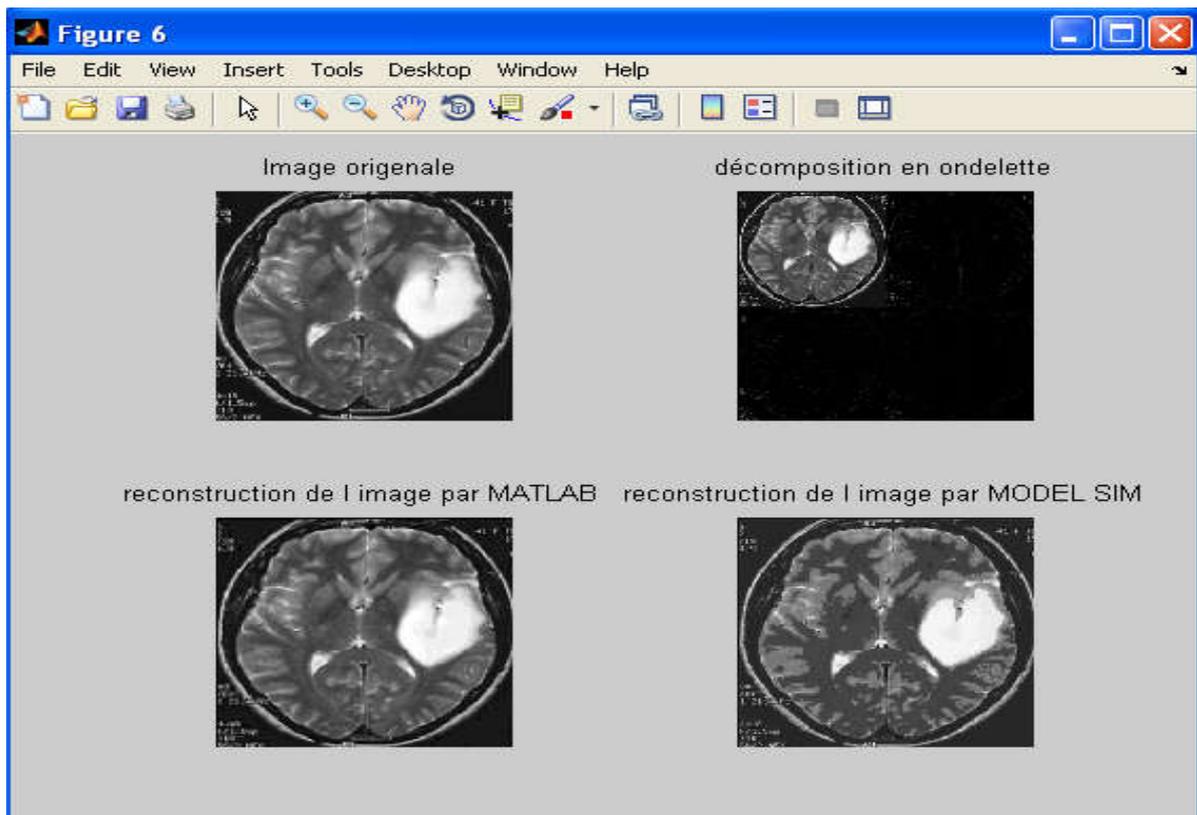
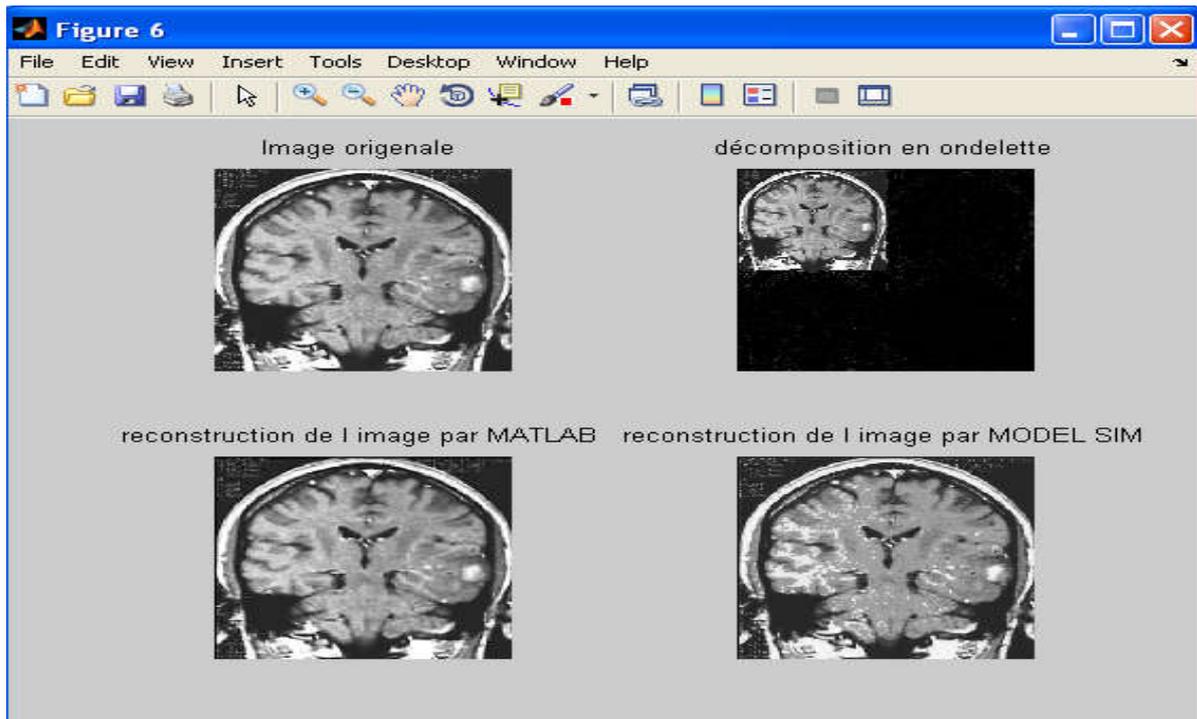
Circuit: 2v8000ff1152-5

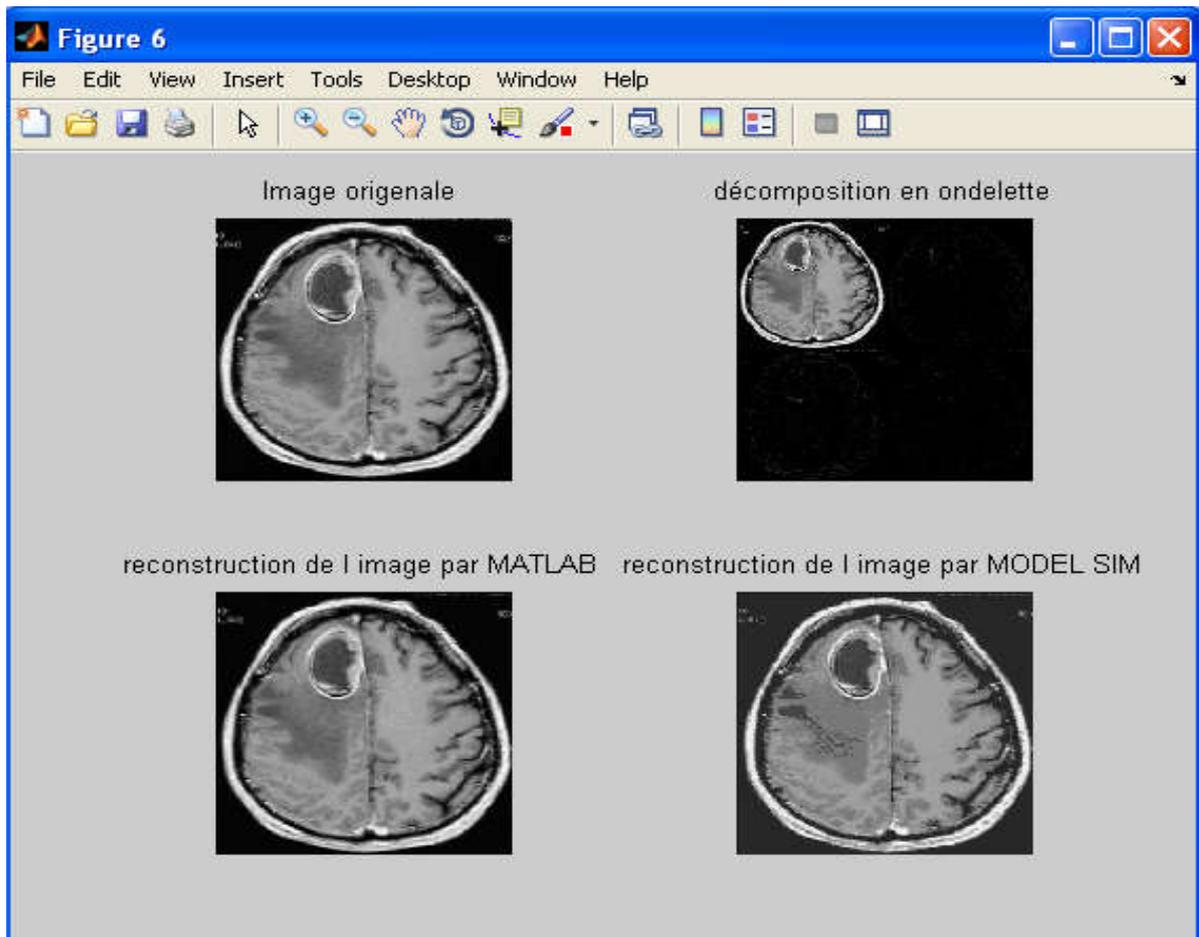
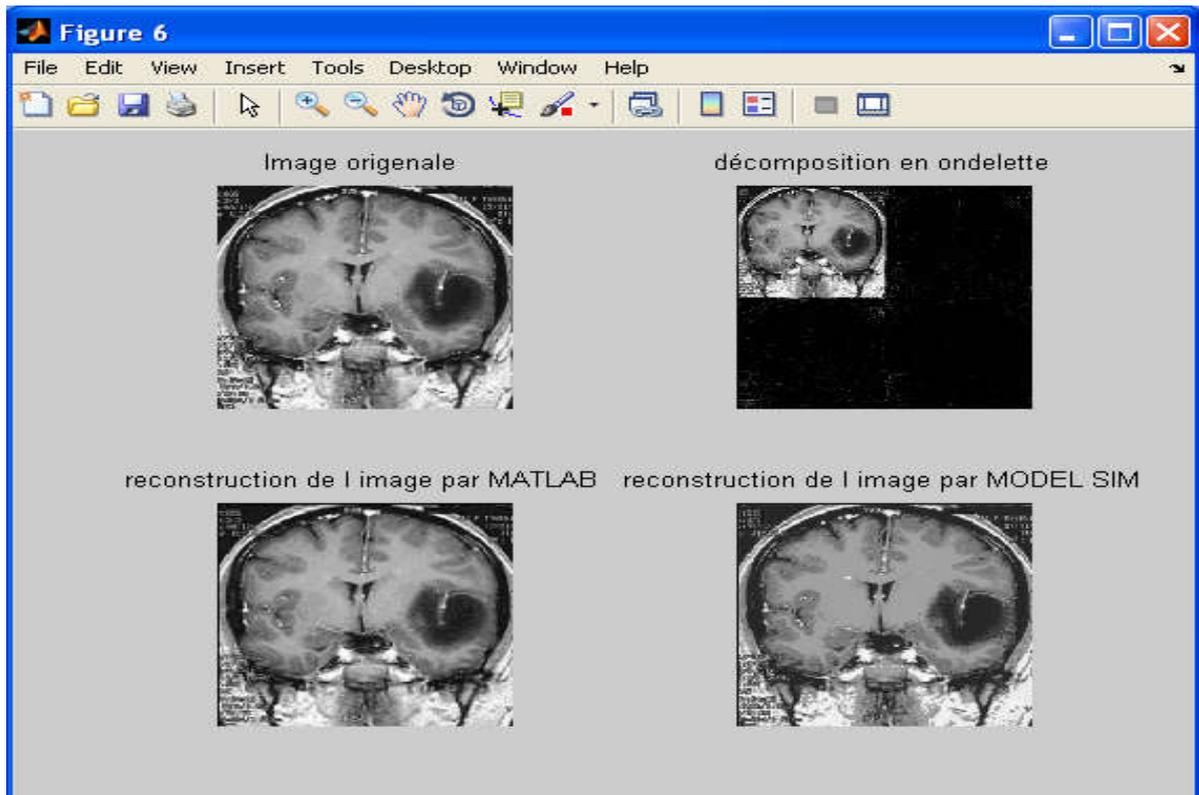
	BLOC1	BLOC2 /QS	BLOC2/QV
Fréquence de fonctionnement	49.9 Mhz /Décomposition 34.829 Mhz /Reconstruction	12.95 Mhz	12.89 Mhz
Délai d'exécution pour une image		63.258 ms	190.659 ms

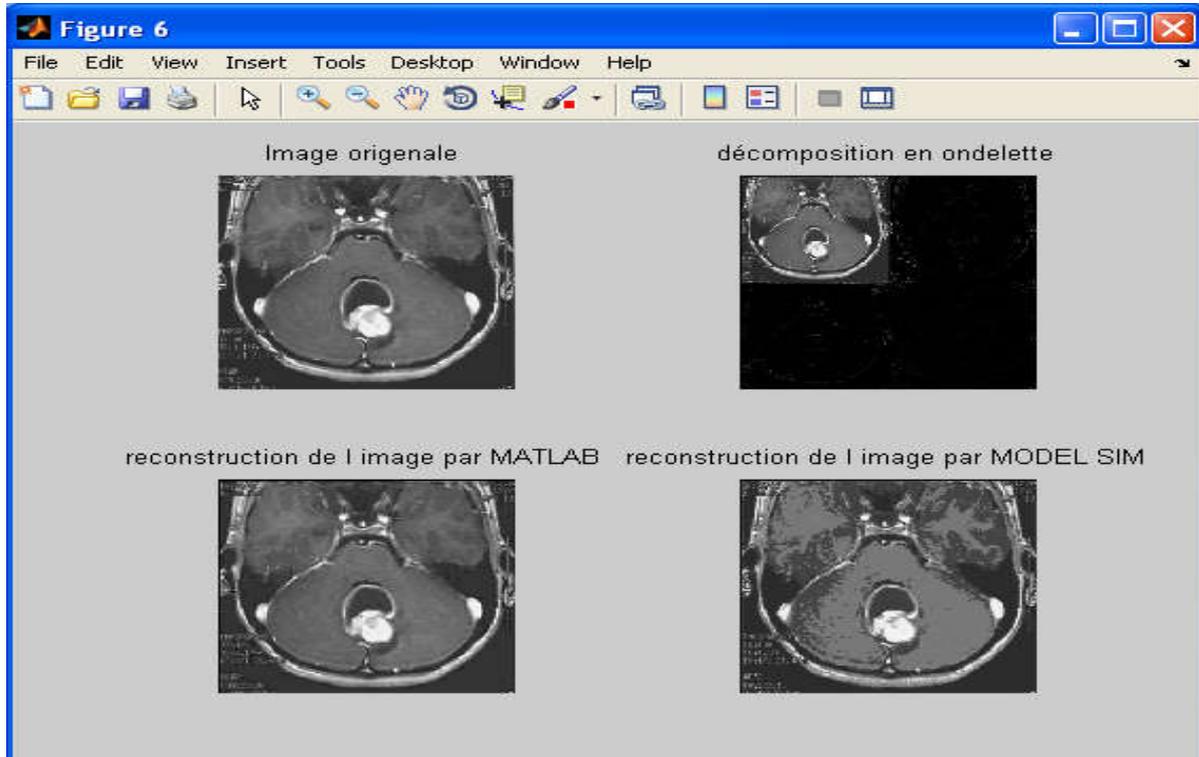
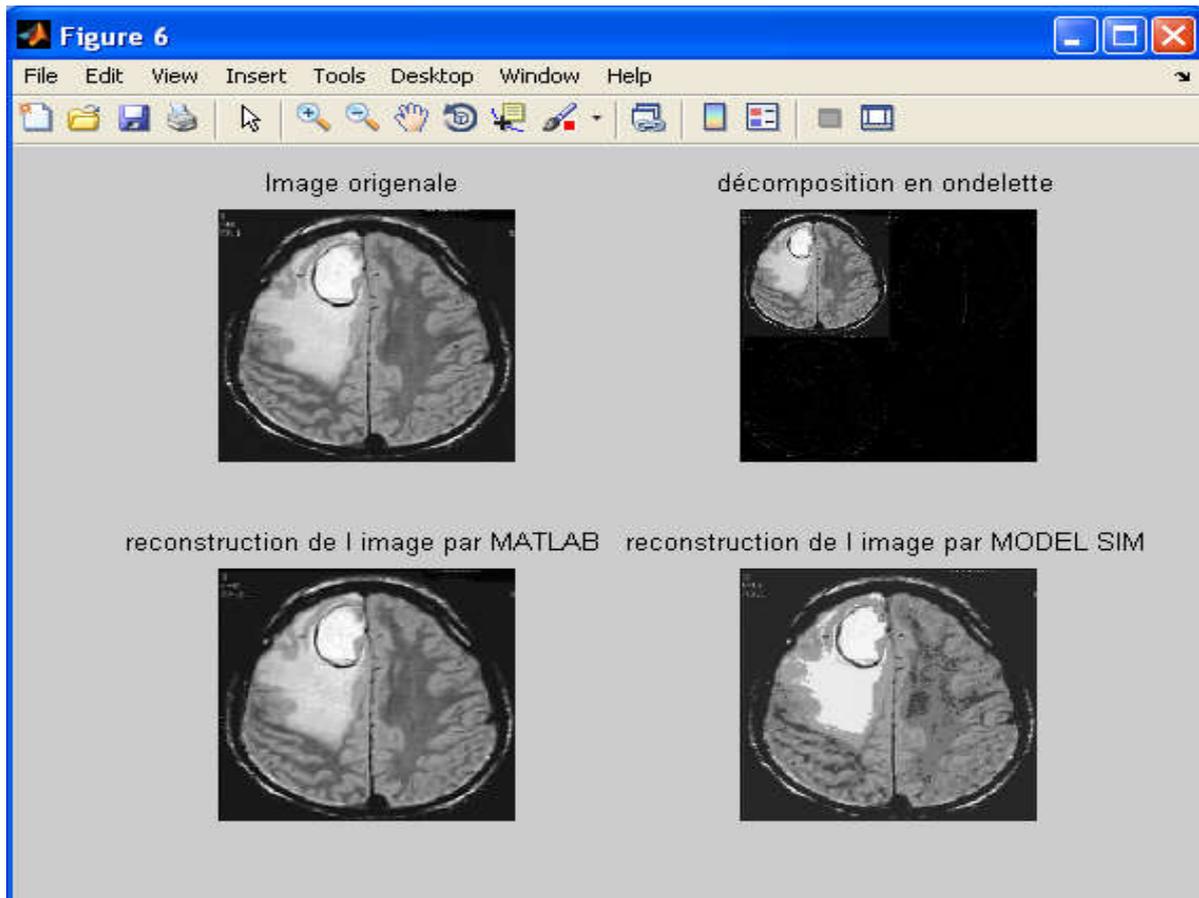
4.4.1. Résultats des différentes images testées sur circuit Virtex II

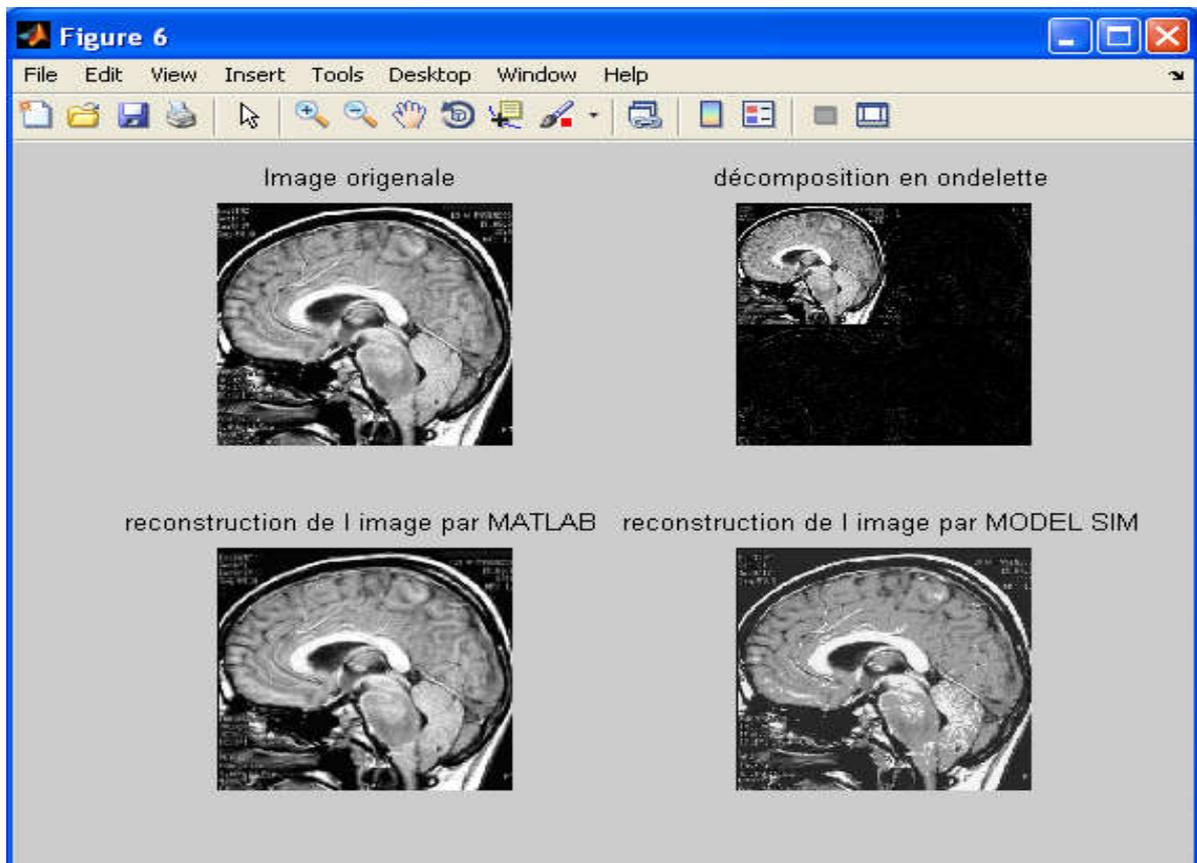
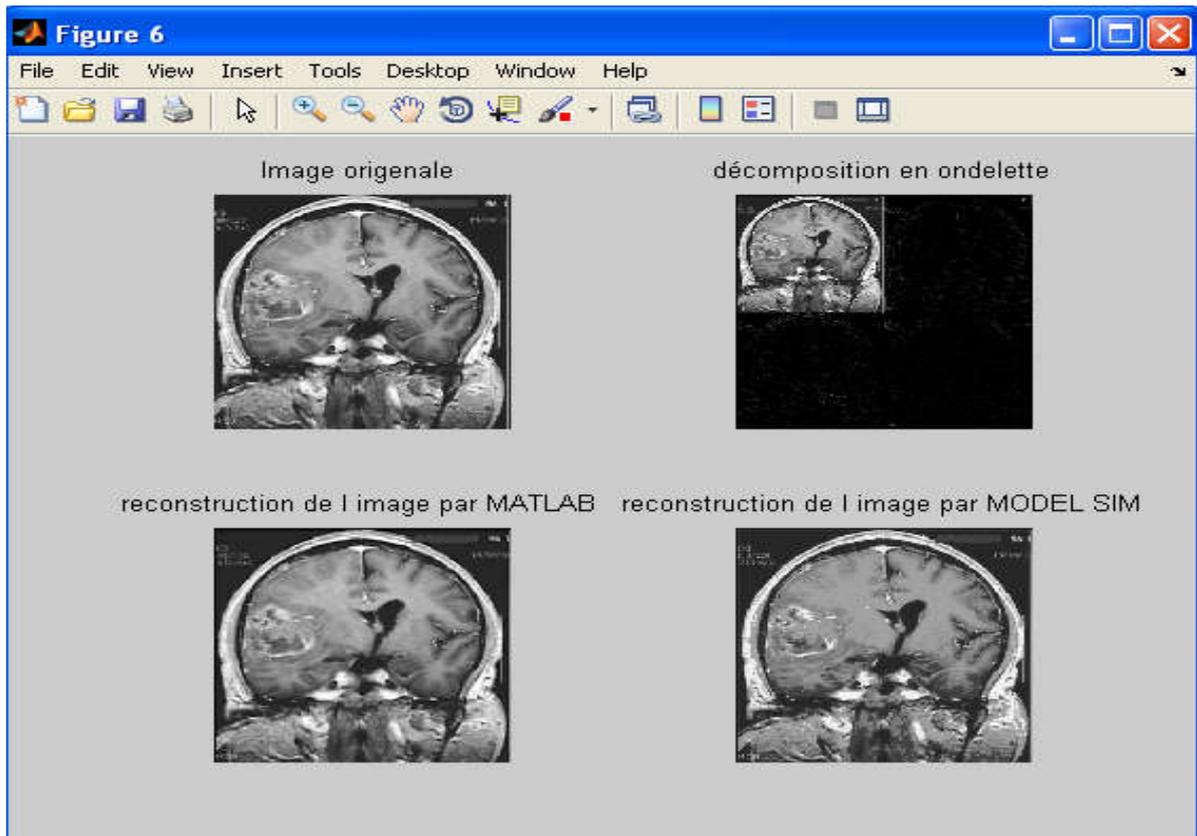
Les figures suivantes, représentation les résultats de test reconstruites en software (par Matlab) et en hardware (par Modelsim).

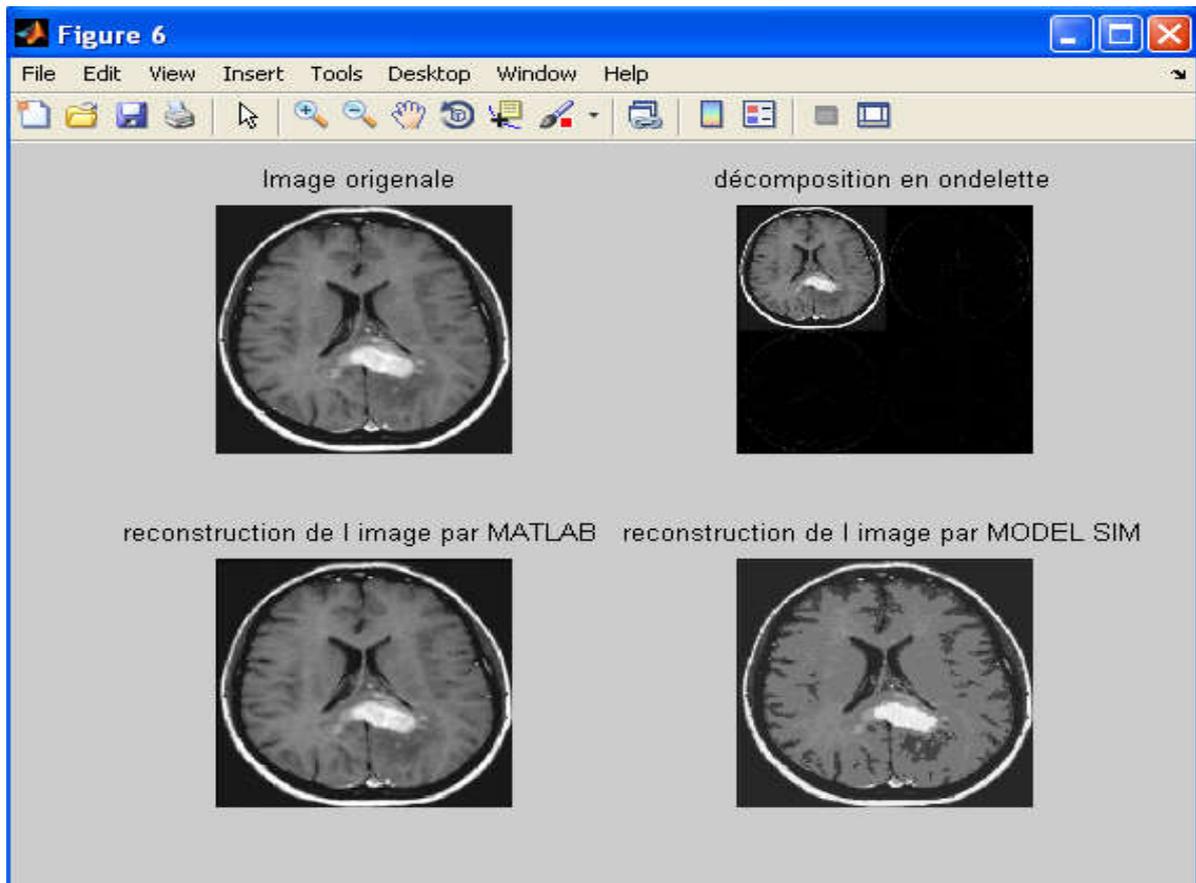
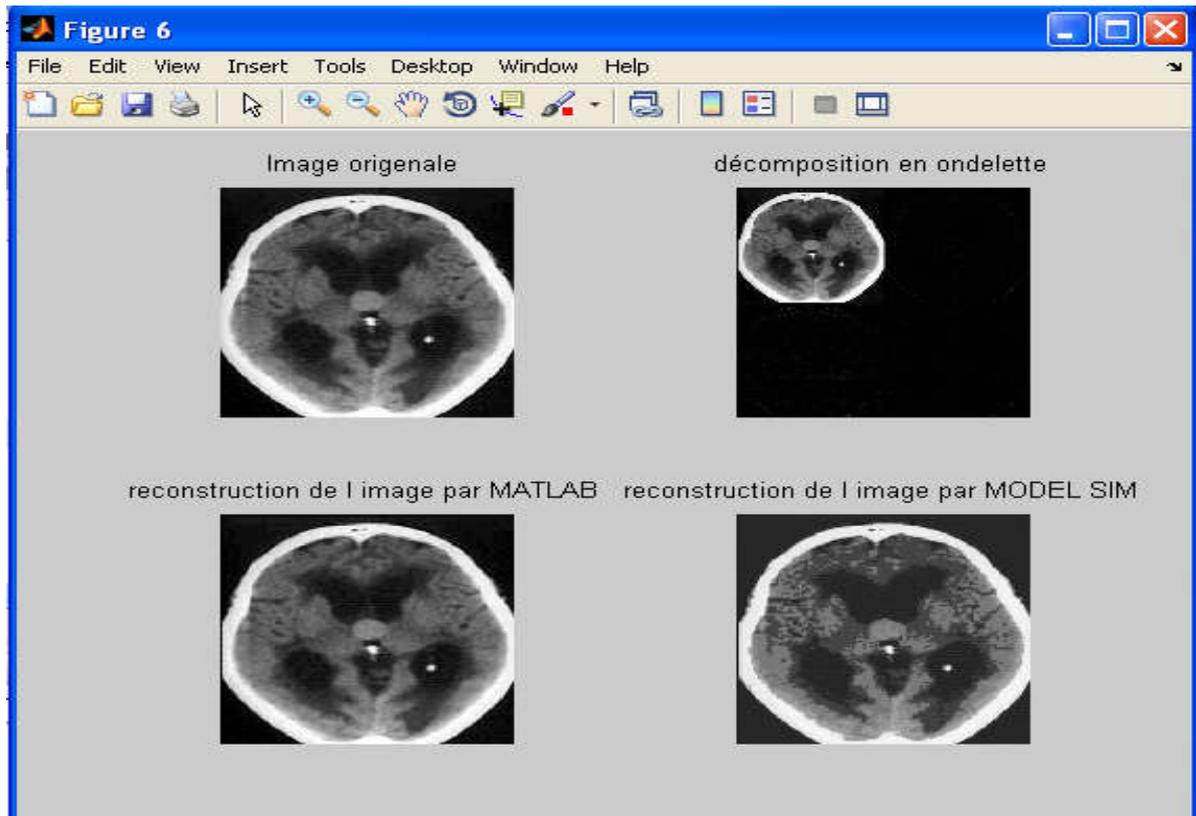


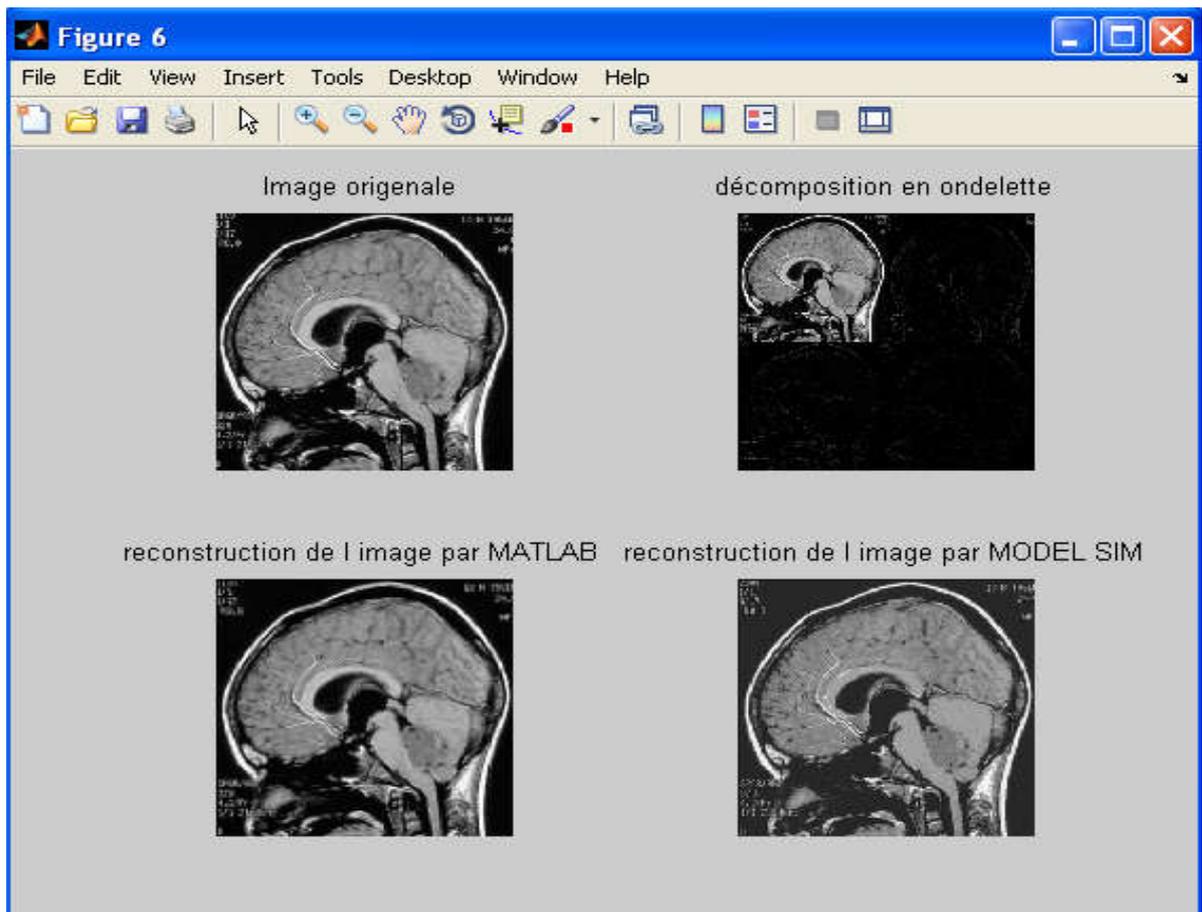
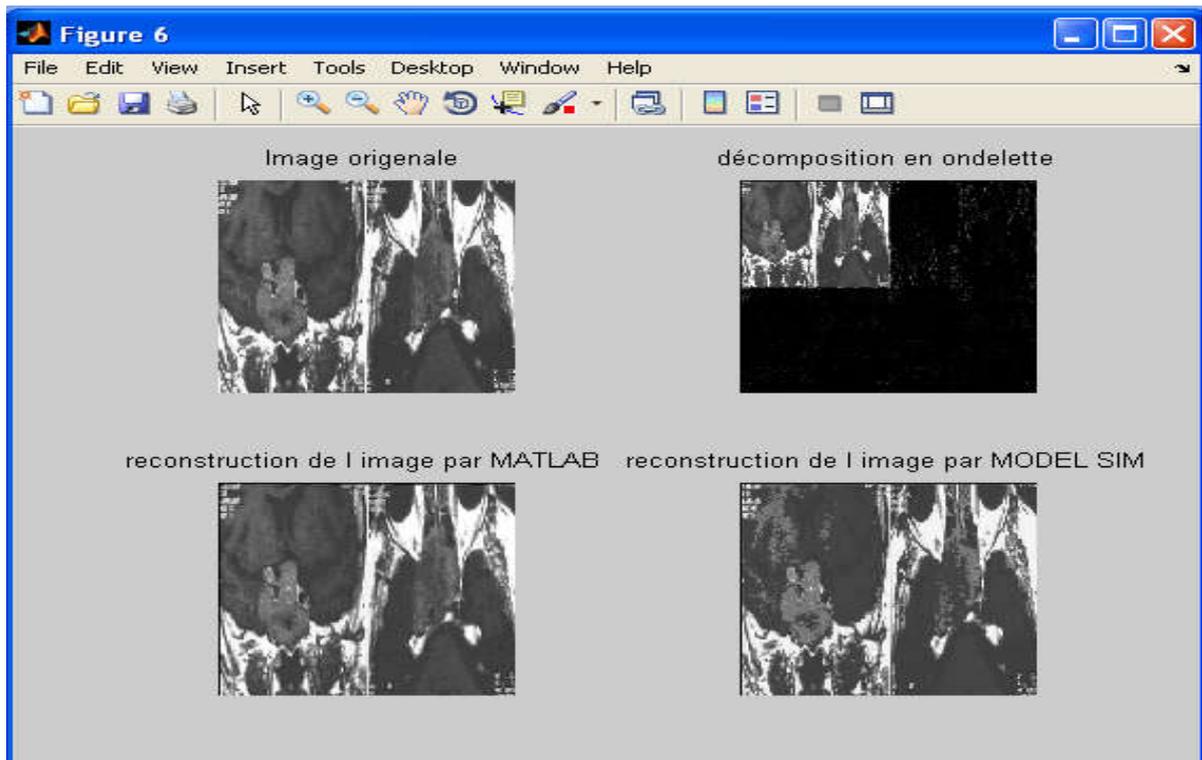


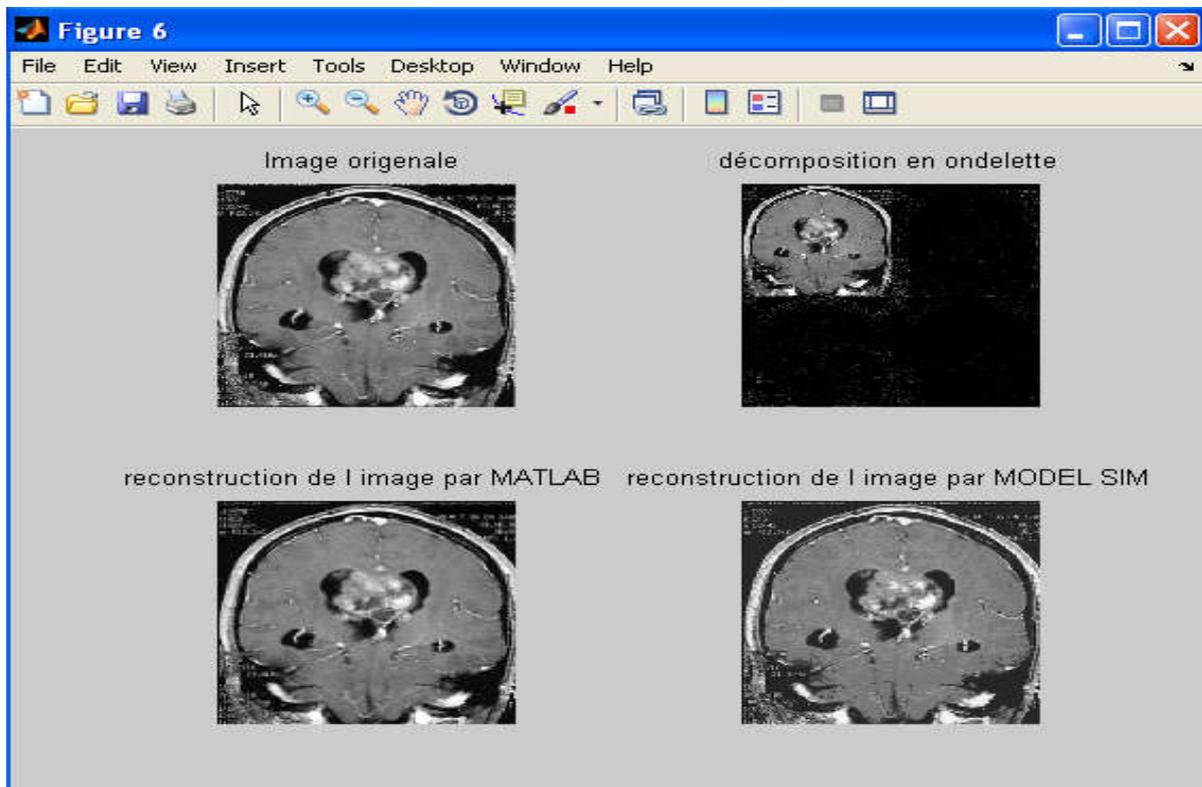
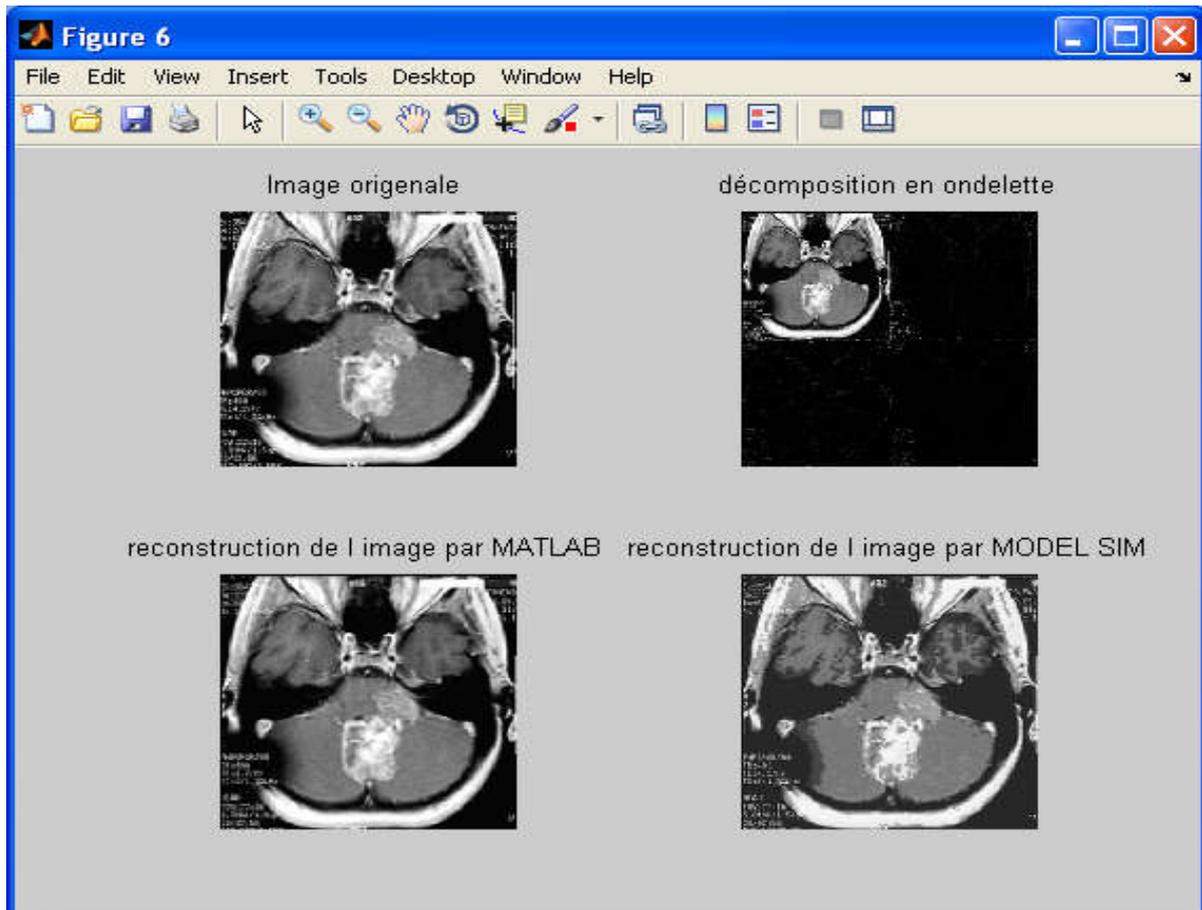


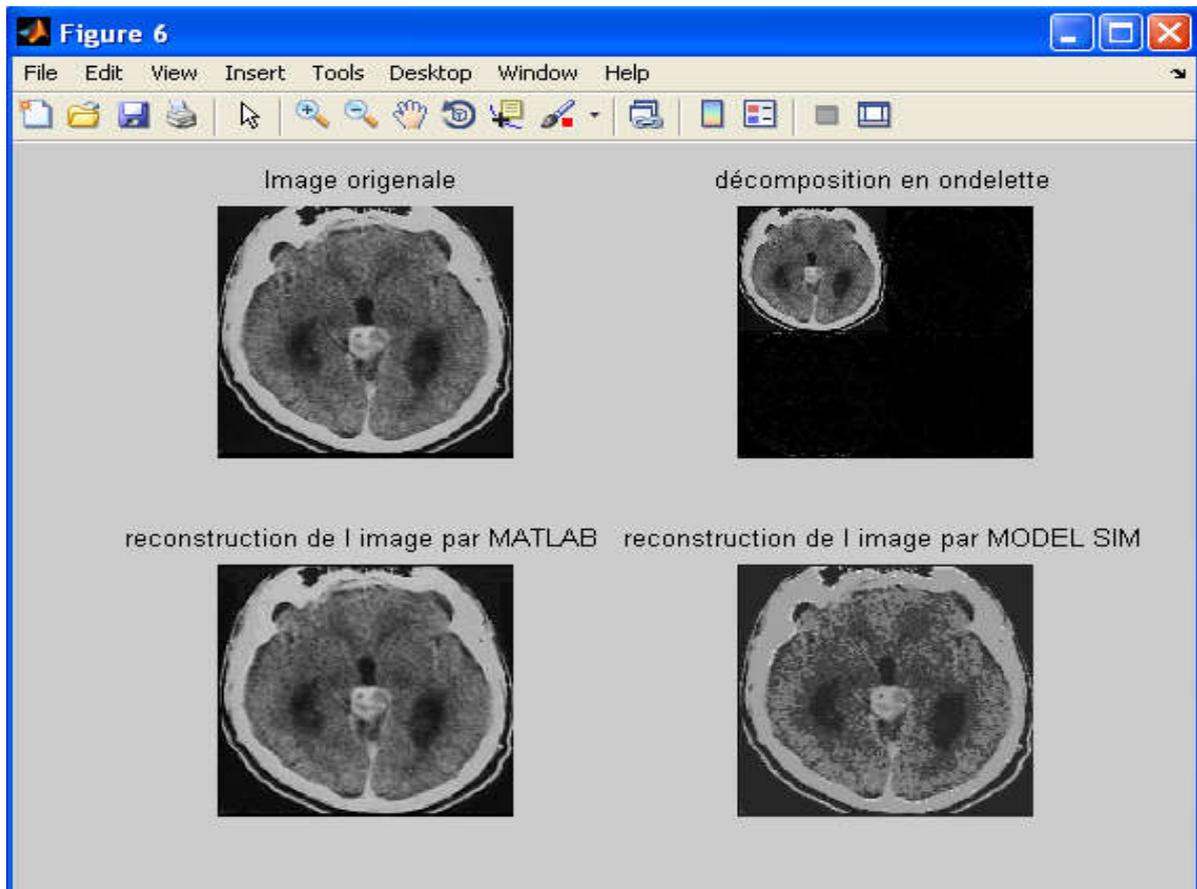
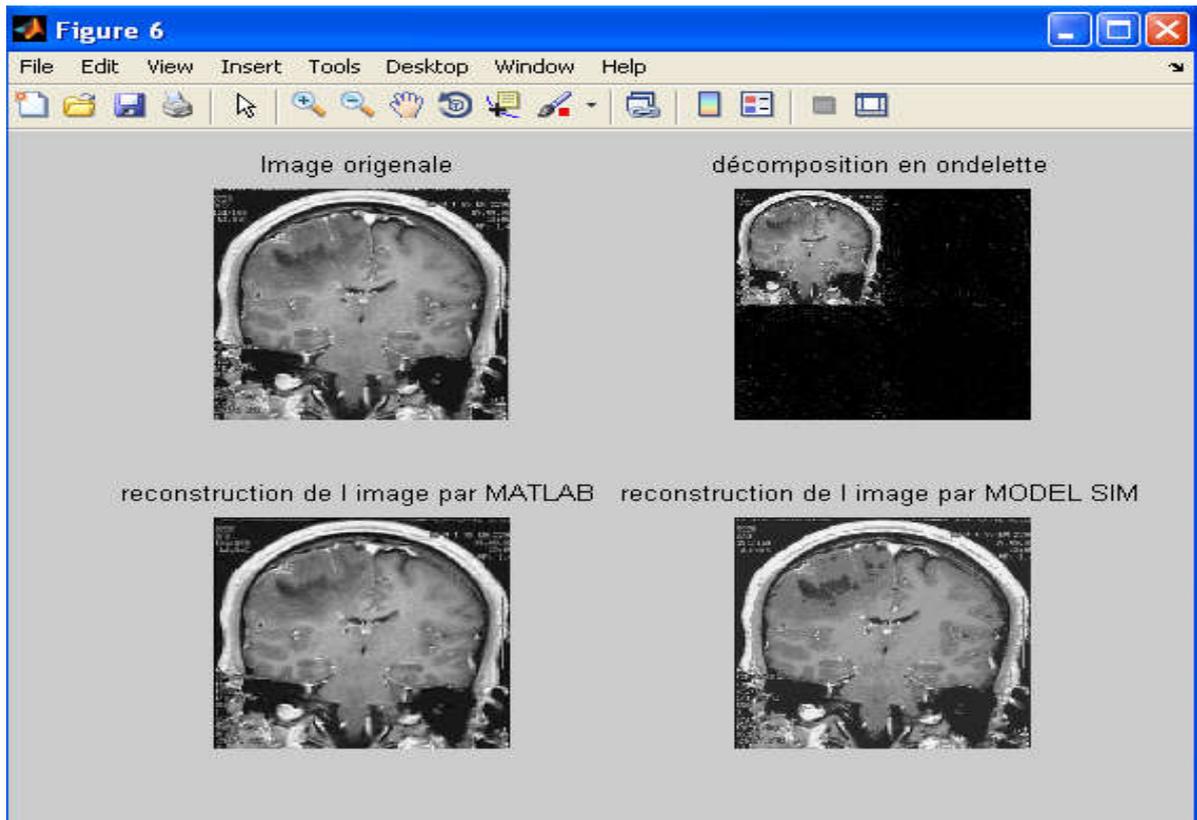












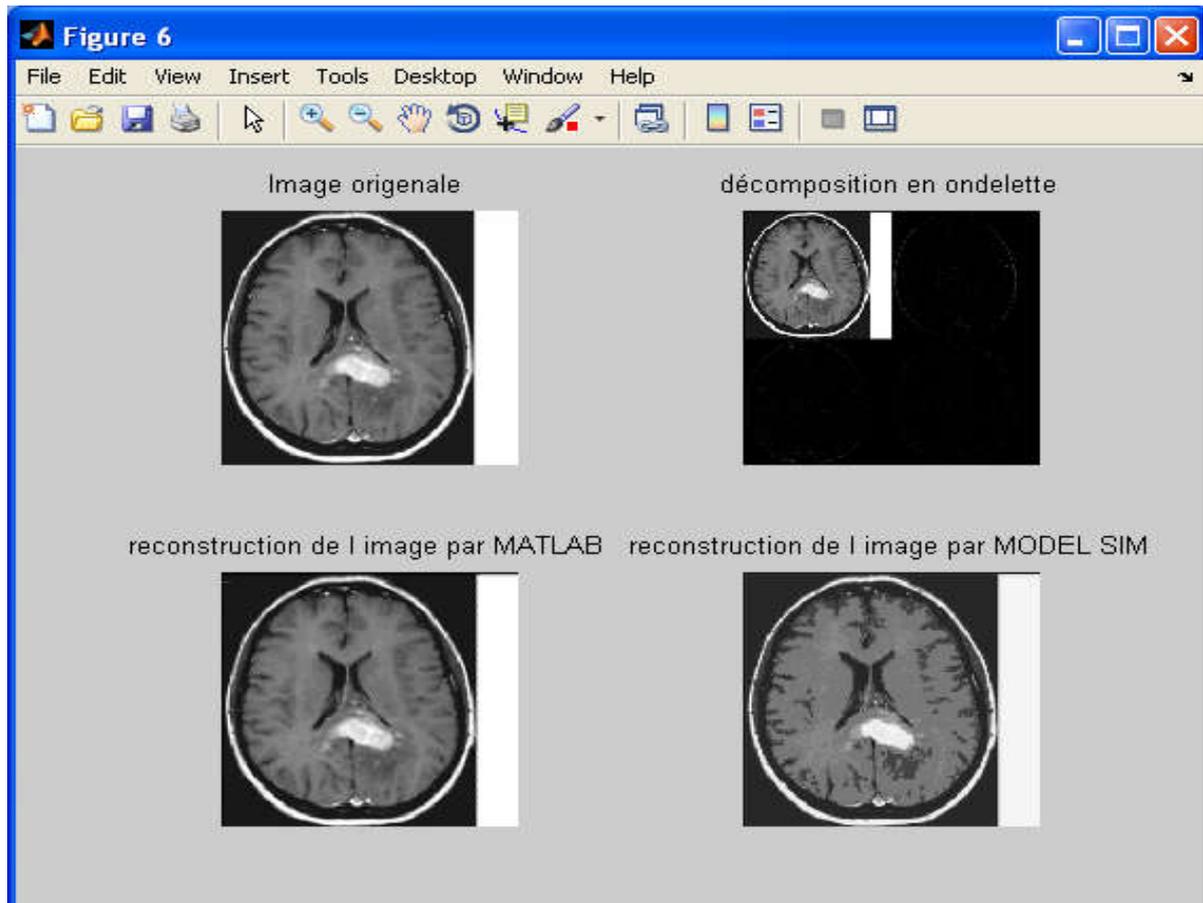


Figure 4.35 : représentation des images test reconstruites en software (par MATLAB) et en hardware (par MODELSIM).

4.5. Résultats d'implémentation et de simulation sur la carte Zedboard

La même méthode été d'écrit en langage C est implémentée sur une nouvelle carte Zedboard présenté annexe C. La Zynq est une famille de FPGA qui contient en plus de la partie logique programmable (Programmable Logic PL) un système microprocesseur ARM double cœur complet avec ses Contrôleurs mémoire et ses périphériques (Processing System PS).

4.5.1. Résultats des différentes images testées sur la carte Zedboard

The screenshot shows the Vivado 2015.4 Project Manager interface for a project named 'project_1'. The 'Implementation Complete' status is shown at the top right. The main window is divided into several sections:

- Synthesis:** Status: Complete. Messages: 3 warnings. Part: xc7z020dgg484-1. Strategy: Vivado Synthesis Defaults.
- Implementation:** Status: Complete. Messages: No errors or warnings. Part: xc7z020dgg484-1. Strategy: Vivado Implementation Defaults. Incremental compiler: None.
- DRC Violations:** No DRC violations were found.
- Timing:** Worst Hold Slack (WHS): NA. Total Hold Slack (THS): NA. Number of Failing Endpoints: NA. Total Number of Endpoints: NA. Implemented Timing Report: [View Report](#). Setup: Hold Pulse Width.
- Utilization - Post-Synthesis:**

Resource	Estimation	Available	Utilization %
LUT	144	53200	0.27
FF	10	106400	0.01
BUFG	1	32	3.12
- Power:** Total On-Chip Power: 1.675 W. Junction Temperature: 44.3 °C. Thermal Margin: 40.7 °C (3.4 W). Effective θ_{JA} : 11.5 °C/W. Power supplied to off-chip devices: 0 W. Confidence level: High.

Below the screenshot, a 2x4 grid of MRI brain scan images is shown, labeled 'Image Originale', 'LL', 'LH', 'HL', 'HH', and 'Image Reconstitue'.

Figure 4.36 : représentation des images test reconstruites en hardware sur la carte Zedbord

4.6. Conclusion

Ce travail a été conçu en utilisant l'environnement de conception de Xilinx (ISE7.1). Tous les blocs ont été décrits en code VHDL. Les tables ont été générées par l'outil core generator. Un intérêt particulier a été accordé au calcul du facteur d'apprentissage β . Pour garantir le bon fonctionnement de notre architecture, celle-ci a été simulée en utilisant l'outil ModelSim ; notre architecture a été implémentée sur le circuit 2v8000ff1152-5. Cette architecture présente un bon compromis qualité et surface d'occupation.

Nous avons proposé et implémenté une technique de compression avec perte des images médicales avec des paramètres optimaux ; ces transformées sont couramment utilisés dans la communauté scientifique pour la compression des images soient ensemble ou séparées [102] - [106] et implémentées sur des systèmes programmables ; la problématique de la compression peut être considérée de différents points de vue. Dans notre travail, nous cherchons à établir un compromis entre le nombre de neurones utilisés et le nombre d'itérations optimaux pour une implémentation sur FPGA en gardant une bonne qualité visuel de l'image originale ; afin d'éviter les erreurs médicales liées à une mauvaise reconstruction de l'image.

CONCLUSION GENERALE

Les images brutes ne peuvent être étudiées telles qu'elles se présentent et doivent subir une étape de transformation nécessaire à l'augmentation de l'efficacité du traitement. Cette étape de transformation consiste, notamment à extraire les données nécessaires pour représenter l'information de manière plus compacte, de nombreuses méthodes et transformées existent. Dans ce sujet, on propose d'intégrer une méthode hybride entre les transformées en ondelettes et les réseaux de neurones pour la compression des images médicales et son implémentation sur FPGA. Nous sommes donc passés d'un niveau abstrait de représentation mathématique puis algorithmique de la transformée en ondelette et des réseaux de neurones à un niveau plus concret d'implémentation sur FPGA. Les étapes d'exécution de ce projet sont:

- La modélisation mathématique de l'algorithme de décomposition et de reconstruction Stéphane Mallat par le mode de calcul en ligne.
- Le choix des paramètres optimaux du réseau de Kohonen (le nombre de neurones et le nombre d'itération).
- La construction d'un quantificateur scalaire de la partie approximation des images médicales.
- La construction d'un quantificateur vectorielle des parties détails des images médicales.
- La validation de notre méthode sous Matlab 7.7.
- La réalisation d'une interface pour la compression des fichiers Dicom avec sélection des zones d'intérêt sur les images.
- Une description de notre architecture en langage VHDL par l'utilisation de l'outil ISE 7.1 de Xilinx et la simulation par l'outil Modelsim; par la suite, une phase de simulation nous a permis une vérification fonctionnelle de cette dernière. Une fois la simulation fonctionnelle validée, une phase de synthèse nous a donné un rapport détaillé, sur les ressources consommées du circuit FPGA et sur la fréquence maximale de fonctionnement. Enfin, la phase de placement et routage a donné schématiquement la surface consommée.

Bien que la transformée en ondelettes et les réseaux de neurones sont des transformées efficaces et relativement complexes à mettre en œuvre, elles présentent l'inconvénient d'être gourmandes en ressources, aussi bien en mémoire qu'en surface. Nous nous sommes particulièrement intéressés aux propriétés de précision et complexité du système. Nos travaux ont eu pour objectif constant de réduire la surface donc la complexité

d'implémentation ; tout en préservant la qualité visuelle de l'image ; afin d'éviter les erreurs médicales liées à une mauvaise reconstruction de l'image.

Ce travail été expertisé par un professeur en neurochirurgie et un physicien médical en radiothérapie.

Synthèse et perspective

La comparaison des performances fournies par toutes les méthodes de compression n'est pas une tâche facile. Contrairement à la compression d'images naturelles, il n'existe pas de bases de données dites de référence pour tester les méthodes. Ainsi, les chercheurs utilisent leurs propres images médicales pour évaluer leurs algorithmes. Or, un résultat valable pour une modalité et un organe ne l'est pas nécessairement pour une autre modalité ou un autre organe. Quoiqu'il en soit, même s'il existe de nombreux outils statistiques d'analyse de performances, l'appréciation de la qualité des images reste du ressort des médecins, ce qui rend les tests à grande échelle difficiles à réaliser. Le défi de l'utilisation de la compression avec perte dans le monde médical est de taille puisqu'il nécessite de contrôler les pertes afin de ne pas affecter le diagnostic.

Les transformées en ondelettes et les réseaux de neurones demeurent des outils largement utilisés ; de ce fait, l'amélioration des performances et de leur exécution constitue une préoccupation majeure dans le contexte d'un traitement massif de l'information et la réalisation des IPs génériques pour d'autres applications telle que la segmentation d'images.

Ce travail aura pour perspective une étape de prétraitement des images médicales bruitées et l'intégration dans un système embarquée pour la télémédecine et sa sécurisation ; afin de contribuer à une amélioration d'un accès équitable aux soins.

PRODUCTION SCIENTIFIQUE

LES PUBLICATIONS

1. F.Alim.Ferhat, H.Bessalah S.Seddiki, O.Kerdjij H.Salhi, M.Issad Yakhlef "Wavelet Transform and Neural Network combined method for medical images Compression and implementation on FPGA ". The Mediterranean Journal of Computers and Networks, Vol. 5, No. 1, pp 34-41, 2009.
<http://www.medjcn.com/papers%20archive/index.html>
http://www.medjcn.com/papers%20archive/Vol11-2015/vol11_4body.html
2. F.Alim.Ferhat-Taleb;H.Bessalah ;S.Seddiki;O.Kerdjij,"Optimal Implementation on FPGA'S based on wavelet transform and self organizing map for medical images compression ",the Journal of Electric Engineering : ISSN :1582-4594 ,Volume 13 / 2013 - Edition : 1
<http://www.jee.ro/covers/art.php?issue=WS132800008W4f27ac0814676>
3. H.Bessalah, F.Alim-Ferhat, M.Bencherif, S.Seddiki: « Toward an Embedded Image Wavelet Transform Implementation Approach». Shaker Verlag ; Transactions on Systems, Signals & Devices Vol. 2, No. 3,pp 291-310. TSSD - 1861-5252/ c 2006.
<http://www.shaker.de/de/index.asp?lang=de>

LES COMMUNICATIONS

1. F.Alim-Ferhat , H.Bessalah, S.Seddiki, O.Kerdjij H.Salhi ,” WT-SOM network Implementation on FPGA for the medical images compression”, IEEE/ACM International Workshop on Computational Intelligence in Medical Imaging (CIMI 2008) October 27-31, 2008, Paris, France.
<http://portal.acm.org/citation.cfm?id=1456318>
2. F.Alim-Ferhat , H.Bessalah, S.Seddiki, O.Kerdjij H.Salhi “On line wavelets transform on a Xilinx FPGA Circuit to medical images compression” ,at the International Conference on Digital Image Processing (ICDIP 2009) published by IEEE computer society Mars 2009 Bangkok Thailand.
<http://ieeexplore.ieee.org/document/5190596/>
3. F.Alim-Ferhat , H.Bessalah, S.Seddiki, O.Kerdjij H.Salhi« Interface pour la compression des fichiers DICOM par une méthode hybride ondelettes et réseaux de neurones ». Third International Conference: E-Medical Systems EMedisys'2010 May 12-14, 2010 –Morocco.
<http://www.setit.rnu.tn/E-medisys/Proceeding.pdf>
4. F. Alim-Ferhat, S. Seddiki, B.Hachemi, A.Boudjellal: « Wavelet energy Embedded into a Level Set Method for Medical Images Segmentation in the Presence of Highly Similar Regions », The International Conference on Mathematics and Computers in Sciences and Industry (MCSI 2014) Varna , Black Sea, Bulgaria, September 13-15, 2014.
<https://csdl.computer.org/csdl/proceedings/mcsi/2014/4324/00/4324a149-abs.html>

APPENDICE A : Les standards

A-1. Les standards.

Il existe plus d'une cinquantaine de types de formats d'image [RCG02]. Pour chacun d'entre eux la structuration des données et les attributs sont différents. La standardisation d'un format d'image permet de régler l'utilisation, la divulgation et la production de logiciels et de hardware compatibles avec le format standard. Le format standard JPEG est le format d'image le plus populaire, et il est devant la scène depuis quelques années. Son successeur, le JPEG2000, semble s'établir dans le domaine de l'image numérique. Le JPEG2000 possède des fonctionnalités supplémentaires par rapport au format JPEG. Cependant, la plupart des appareils numériques (appareils photos, caméscopes, téléphones portable, etc) et les logiciels qui capturent et traitent les images sont au format JPEG.

A.1.1. JPEG (Joint Photographic Expert Group).

Le comité JPEG a été créé en 1986 par la jonction de plusieurs groupes qui travaillaient sur la photographie. Ce comité a produit la norme de compression d'images photographiques qui a été standardisée 1994 et a reçu son nom JPEG. Il est devenu le format le plus populaire très rapidement parce qu'il a été conçu avec différentes contraintes. L'algorithme JPEG doit être implémentable sur une grande variété de types de CPU (unité centrale de calcul) et sur des cartes plus spécialisées (appareil photo numérique et téléphone portable par exemple).

Il doit pouvoir compresser efficacement tout type d'images réelles (images photographiques, médicales) avec pertes et sans perte.

Il possède quatre modes de fonctionnement : séquentiel, progressif, sans perte, hiérarchique. Entre les 4 modes de compression de la norme JPEG, le séquentiel est le mode principal le plus répandu. Il est basé sur la transformation DCT (Discret Cosinus Transform), quantification scalaire et le codage d'Huffman sur pixels de 8 bits par plan de couleur.

A.1.2. JPEG-LS (Lossless JPEG)

La compression JPEG en mode sans perte n'est pas optimisée. Les objectifs du comité étaient de concevoir un mode réversible permettant une compression de l'image à 50%. Malheureusement, un code réversible est quasiment impossible avec l'utilisation de la DCT. Les erreurs d'arrondis dues à la précision limitée de calcul sont toujours présentes dans JPEG-

LS le procédé de compression est composé de trois parties : la prédiction de la valeur du pixel qui est faite par rapport aux pixels voisins en utilisant l'approche de prédiction MED (Median Edge Detection) ; la détermination d'un contexte. Ce contexte représente l'environnement du pixel à coder et ses voisins. L'idée est de prendre le meilleur environnement qui affine la prédiction avant le codage et de réduire le nombre de paramètres de l'erreur de prédiction ; le codage de l'erreur de prédiction dont l'approche est de réinsérer l'erreur de prédiction dans le système puis de la comparer à d'autres mesures d'erreur. Le JPEG-LS possède une option de haut taux de compression, mais cette option est quasiment sans perte.

A.1.3. JPEG 2000

Le JPEG2000 remplace le JPEG comme le format standard pour la compression des images. Il a été réalisé dans la perspective de répondre aux exigences des nouvelles applications les plus diversifiées, comme la multirésolution.

La compression JPEG2000 est composée de plusieurs étapes avec pertes et sans perte.

Tout d'abord, un changement d'échelle est effectué dans chaque composante couleur RGB, Cette transformation peut être réversible (sans perte) ou irréversible (avec pertes). L'image est découpé en petites images appelées tuiles, chaque tuile est considérée comme une image et est traitée de façon indépendante. Du fait de la complexité du mécanisme du JPEG2000, la décomposition de l'image en tuiles rend possible l'application du JPEG2000 sur des images de taille importante.

La transformation de domaine est faite à l'aide d'une décomposition par transformation en ondelettes Chaque tuile de chaque composante subit des transformations selon les types compressions réversible et irréversible. L'image est décomposée en plusieurs composantes. L'image et ses composantes sont partitionnées en blocs qui seront compressés séparément. Les valeurs de chaque bloc sont centrées par rapport à la moyenne du bloc.

- Une transformation discrète d'ondelette DWT (pour Discrete Wavelet Transform) est opérée sur chaque bloc. Le filtre choisi peut être réversible ou non-réversible, pour permettre la possibilité de compresser sans respectivement avec distortion. Le standard utilise la paire (5,3) réversible de Le Gall et la paire (9,7) irréversible de Daubechies.
- Les sous-bandes issues de la DWT sont quantifiées selon une méthode scalaire. La quantification est indépendante pour chaque sous-bande. Puis, ces sous-bandes sont divisées en blocs, généralement de 64×64 , nommés blocs de code.
- Les blocs de code sont encodés par une méthode arithmétique. Le codage se fait de telle sorte qu'une région d'intérêt (ROI) définie puisse être codée avec une meilleure qualité que le

reste de l'image. Des marqueurs sont insérés dans le flux de bits, renforçant la robustesse aux erreurs de transmission. Un en-tête décrit l'image originale et ses différentes décompositions. Ce type de codage permet de localiser, extraire, et reconstruire l'image en choisissant la résolution, la qualité et une région d'intérêt. Il permet également de contrôler le débit après l'étape de compression.

A-2.3 comparaison entre les algorithmes.

Bien que des tests montrent la supériorité de Jpeg2000 en terme de taux de compression sur les autres standards Jpeg. Pour une compression presque sans perte, Jpeg2000 est moins performant que Jpeg-Ls. Par contre d'autre étude on prouver la supériorité de l'algorithme de kohonen pour la compression des images naturelle référence comme celle de « Lena » figure A.1.



Figure A.1 comparaison de différents algorithmes de compression sur une image test.

A.3. Les standards des images médicales

Un standard est une spécification adaptée, pour décrire un procédé de façon cohérente. Dans le cas de la compression de données, l'utilisation de standards est indiquée puisque le processus de compression vise à la création d'une forme intermédiaire de l'information, plus compacte et facilitant sa transmission sur des réseaux de communication, son stockage, sa récupération. Il est donc nécessaire que cette forme intermédiaire et les mécanismes d'accès à cette forme, soient conformes à des spécifications qui garantissent l'interfonctionnement harmonieux des applications.

L'utilisation des standards dans le domaine de la santé est incontournable, pour répondre aux besoins d'interopérabilité et de pérennité.

A.3.1. La norme DICOM

DICOM est un protocole d'échange d'images, par réseau ou par l'intermédiaire de supports physiques. Compte tenu de la diversité des modalités d'images, le standard est organisé de façon à permettre à la fois le respect des spécificités de chaque modalité et la mise en commun de nombreux éléments de données. Tous les grands constructeurs de l'imagerie médicale contribuent activement au développement du standard. Il existe aujourd'hui 26 groupe de travail, réunissant environ 750 experts techniques ou médicaux. Le standard est modulaire, afin de répondre aux besoins communs ou spécifiques qui concernent les multiples modalités d'imagerie et des domaines de spécialités médicales faisant appel aux images Le standard DICOM couvre de nombreux aspects parmi lesquels :

- La communication des images et des données associées (en mode connecté et par l'utilisation de supports physiques), pour l'ensemble des techniques existantes et modalités d'imagerie.
- L'impression des images sur des supports physiques.
- La gestion des activités liées à l'acquisition, au traitement et à l'interprétation des images.
- La sécurisation des échanges, et différents mécanismes de signature des documents.

A.3.2 Health Level Seven (HL7)

Health Level Seven_HL7 le nom du standard fait référence à la 7^{ème} couche « application » du modèle OSI (Open System Interconnection) de l'ISO (International Standards Organisation). La raison était le développement de message standardisé permettant aux applications des systèmes d'information de santé de communiquer entre elles. Il existe un groupe de travail commun entre le comité DICOM et HL7 pour harmoniser les échanges de données touchant aux examens d'imagerie et aux comptes rendus structurés entre ces deux standards.

De plus en plus, les médecins ont besoin de visualiser plusieurs fois les clichés, de les communiquer à des confrères pour partager leur avis et leur diagnostic. Ainsi, le stockage, la visualisation et la communication sont les trois principes qui ont forcé l'arrivée du format numérique des images dans le milieu médical. Pour que les données puissent coïncider, les différentes techniques d'acquisition d'images doivent être compatibles, au moins par leur support numérique. De ce fait découle une évidence : les différents appareils d'imagerie, ou les différentes marques doivent pouvoir communiquer entre eux, et doivent pouvoir supporter le même standard de fichier.

A.4. Comparaison entre les méthodes.

La compression des images fixes a été réalisée par plusieurs techniques parmi les plus connues : les standards JPEG et JPEG2000, ont beau fournir d'excellentes performances en termes de rapport débit-distorsion, mais leur complexité est telle que cela coûte plus d'énergie de compresser l'image que de la transmettre sans compression. De plus, ces méthodes résistent très mal aux erreurs (perte d'information) apportées par le canal de transmission ; l'absence de quelques bits d'information au récepteur suffit pour entraîner une dégradation brutale de la qualité de l'image décodée, voire même l'impossibilité de décoder l'image dans le contexte particulier de l'imagerie médicale

La figure A.2 présente la robustesse d'une image médicale aux erreurs de transmission de la carte de Kohonen

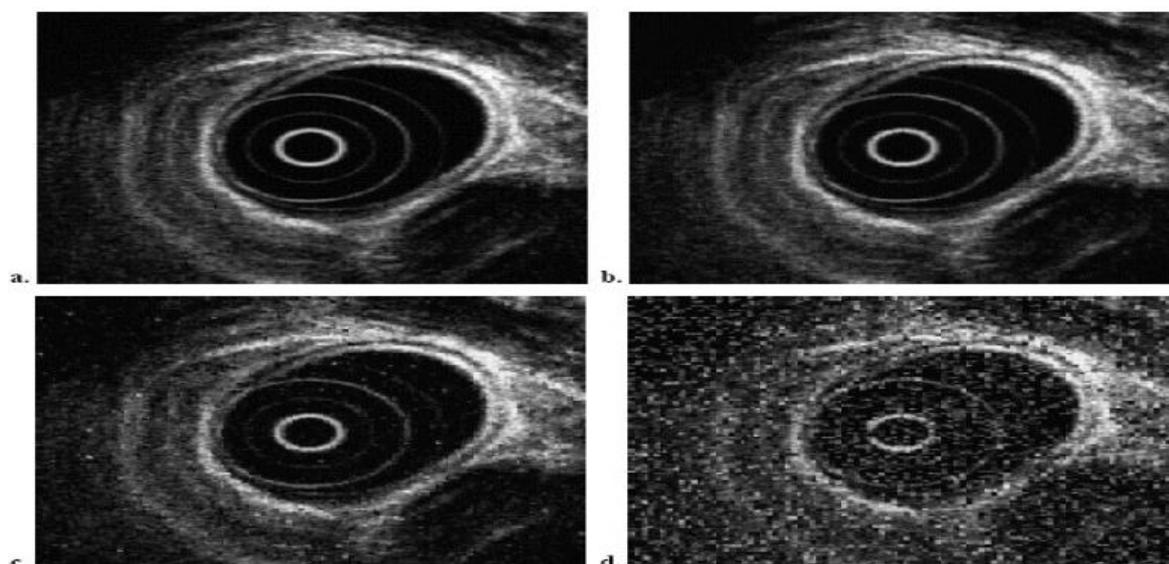


Figure A.2 : Robustesse aux erreurs de transmission de la carte de Kohonen

a : image originale, b : Image reçue à travers un canal non bruité PSNR= 29.76 dB,
 c : Image reçue à travers un canal bruité avec un dictionnaire de Kohonen PSNR= 24.48 dB, d
 : Image reçue à travers un canal bruité avec un dictionnaire autre que celui de Kohonen
 PSNR= 17.31 dB.

APPENDICE B : L'interface graphique compression DICOM

B.1. La norme DICOM

La norme DICOM est l'abréviation de « Digital Imaging and COmmunication in Medicine ». C'est une norme définissant une méthode de communication entre les différents équipements utilisés en imagerie médicale. Elle est aujourd'hui adoptée par la plupart des constructeurs. Avec chaque machine respectant cette norme est émise une déclaration de conformité (Conformance Statement) facilitant l'interopérabilité avec d'autres dispositifs. C'est aussi le format d'image faisant référence, dans le domaine médical, à cette norme. Il permet l'archivage aisé des images et des informations qui y sont associés et facilite leur transmission.

Nous avons développé notre méthode de compression sur les images du fichier DICOM avec sélection des régions d'intérêts (Region of Interest), suivant les étapes suivantes :

B.2. L'interface graphique compression DICOM

L'interface que nous avons réalisée donne aux médecins des choix multiples figure B.1. Toutes les options relatives à la compression et la décompression y sont présentées et afficher comme suite.

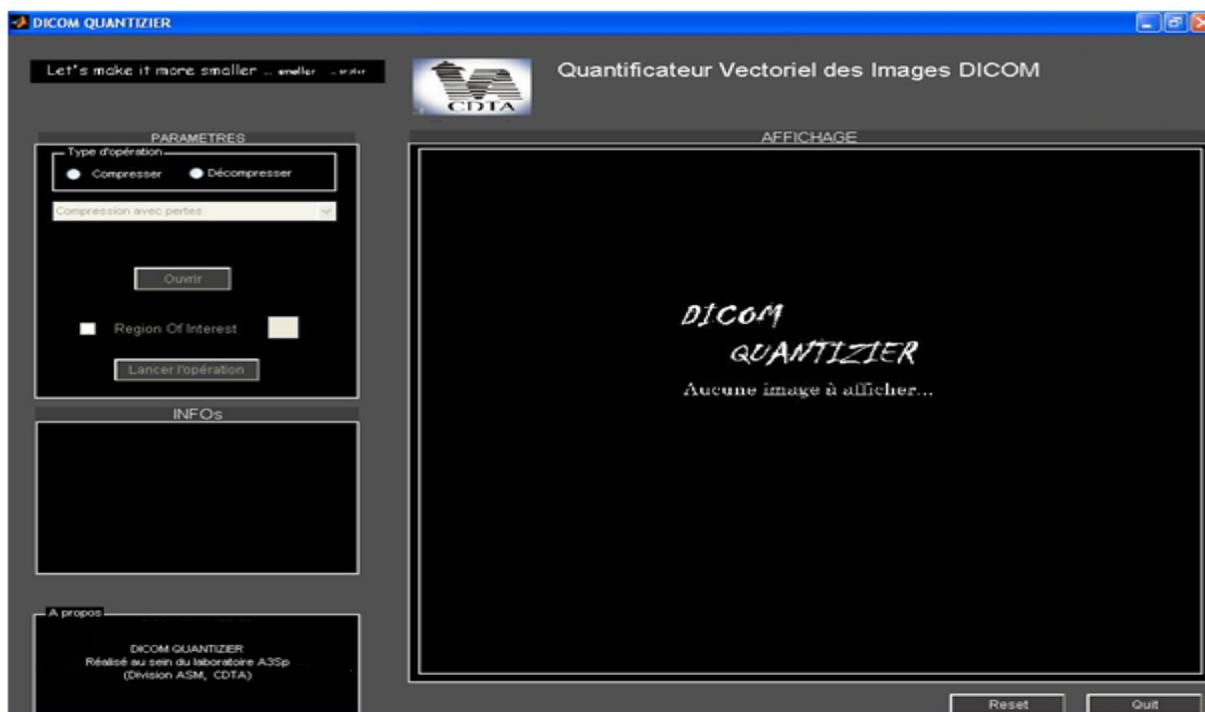


Figure.B.1 : Interface de compression des fichiers DICOM

Dans la première zone active, Type d'opération, on trouve les deux choix : compresser et décompresser. En fonction de ce choix, d'autres zones seront activées ou désactivées. Si le choix « compresser » est validé, on peut choisir, la classe de la compression à savoir : avec pertes ou sans pertes. Si la compression avec pertes est choisie, l'option « Région d'intérêt » sera accessible, et le choix du nombre des régions d'intérêt sont sélectionnés figure.B.2. Une fois chargé, le nom de fichier ainsi que sa taille sont affichés. L'image entourée des informations textuelles est affichée au début d'une compression et à la fin d'une décompression.

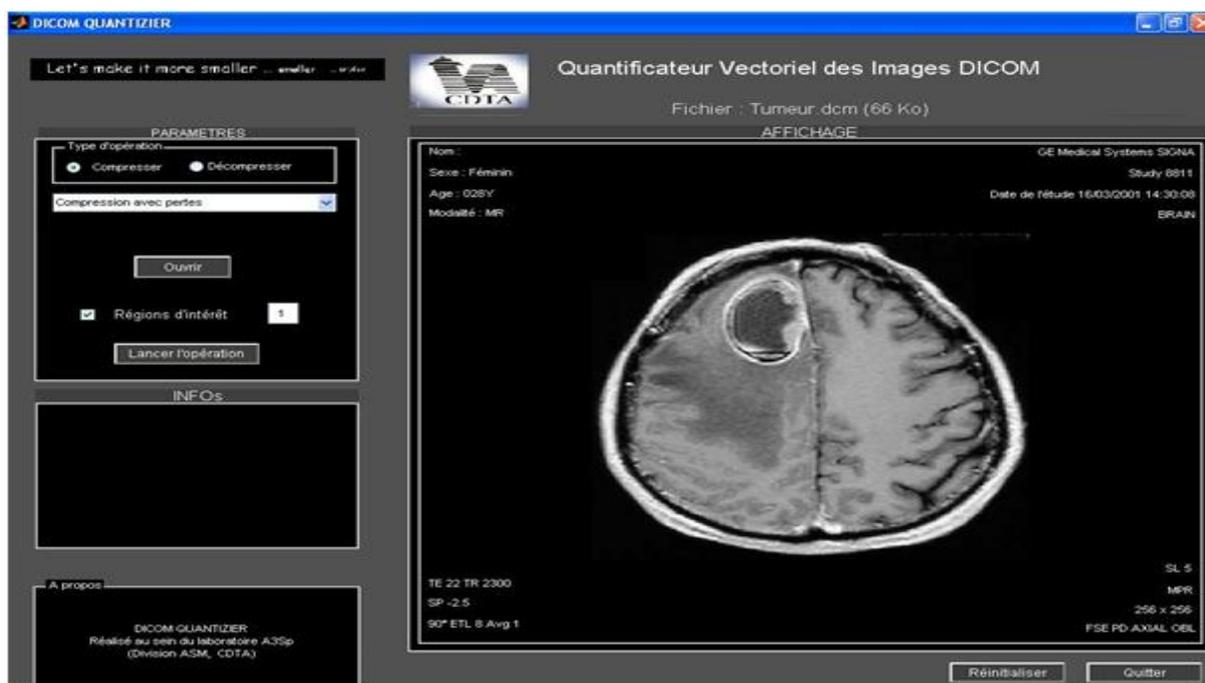


Figure B.2 : L'image et les informations seront affichées

Le nombre des régions d'intérêt doit être précisé. Dans la figure B.3, nous avons pris comme exemple une région (1).

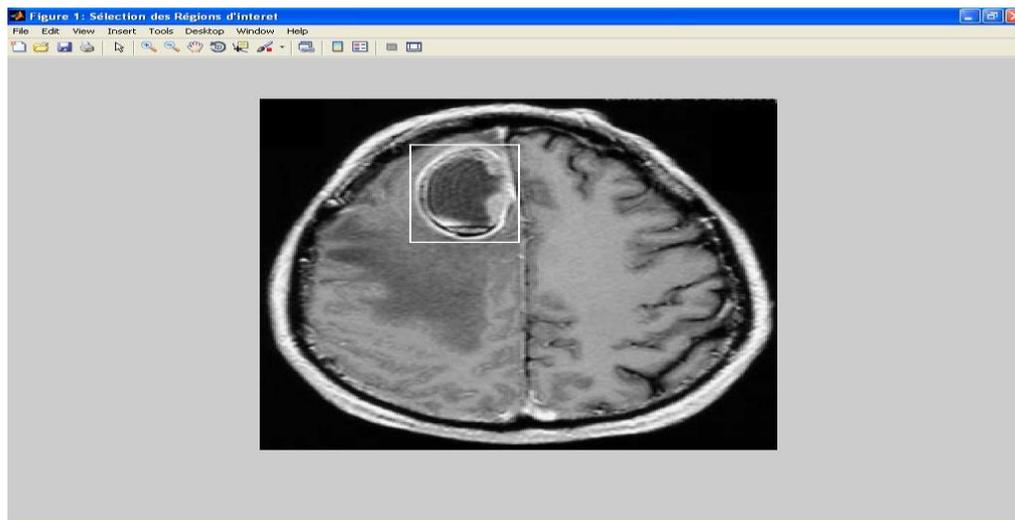


Figure.B.3 : Choix du nombre des régions d'intérêt sont sélectionnés

L'opération peut prendre plus d'une minute, en fonction de la taille et de la complexité de l'image. Un fichier .µdcm est généré. C'est le fichier compressé. Finalement, les résultats de l'opération sont affichés :

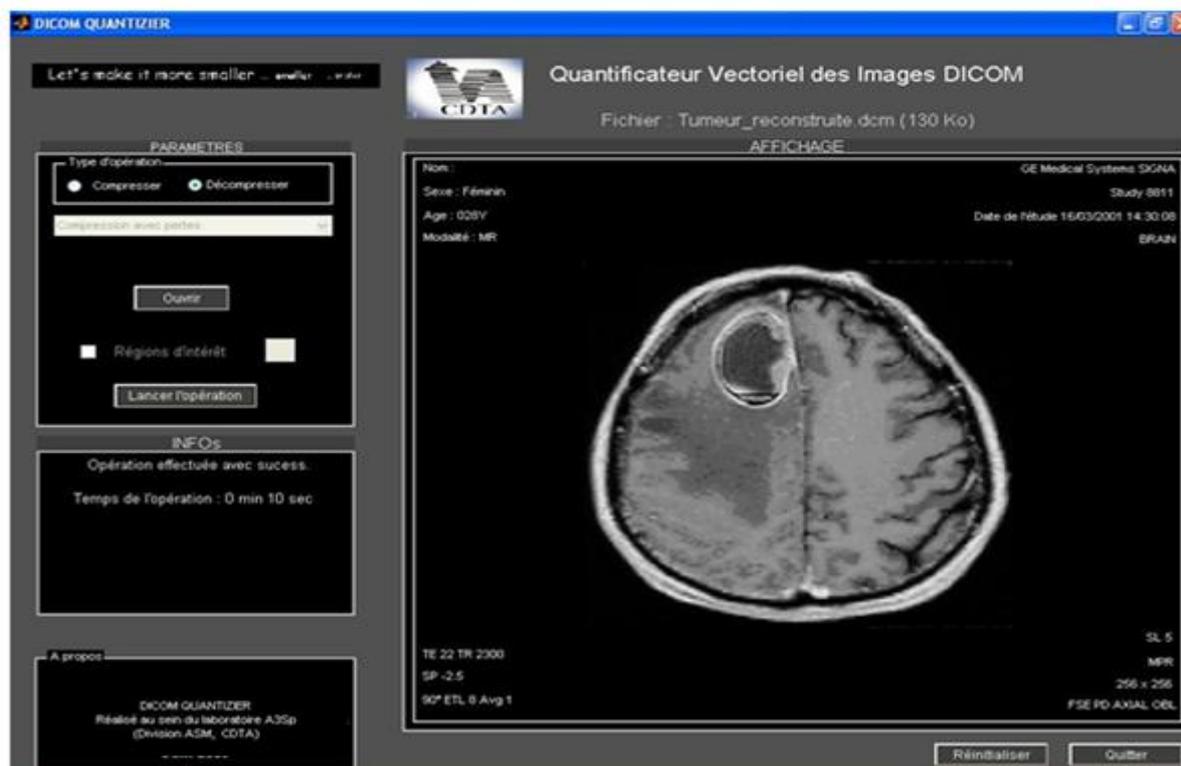
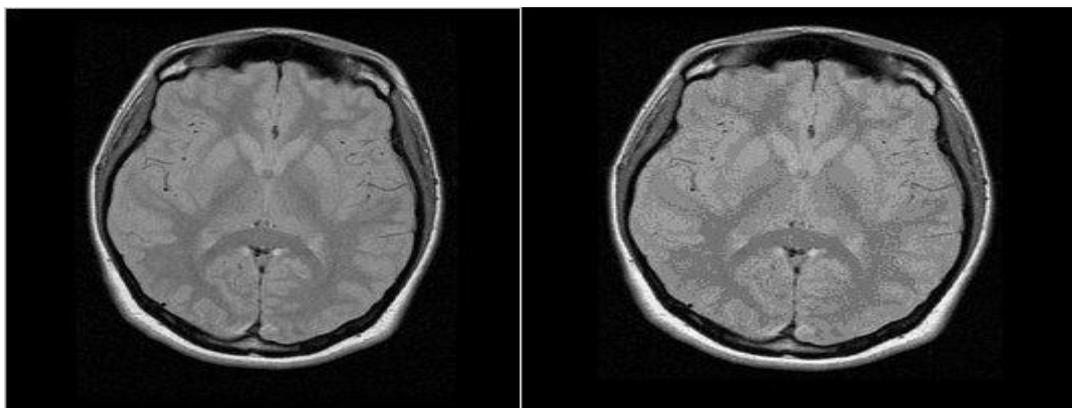


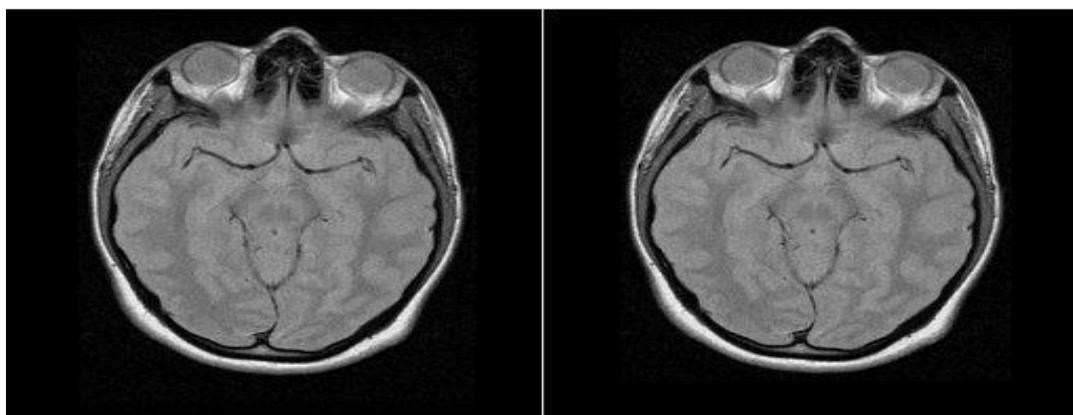
Figure B.4 : Résultats de compression

B.3. Les exemples d'images Dicom; pour le calcul des performance du Tableau 5.5 du chapitre 5



ImageIRM du cerveau originale

images reconstruite



ImageIRM du cerveau originale

images reconstruite

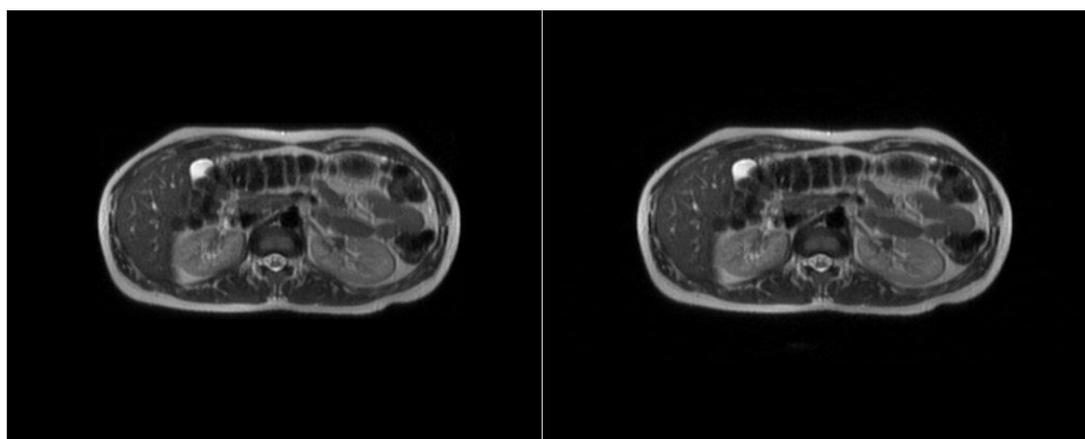


Image IRM de l'abdomen

images reconstruite

APPENDICE C : Les FPGA's (Field Programmable Gate Array)

C.1. Introduction :

Les FPGA (Field Programmable Gate Arrays ou réseaux logiques programmables) sont des composants VLSI entièrement reconfigurables ce qui permet de les reprogrammer à volonté afin d'accélérer notablement certaines phases de calculs.

L'avantage de ce genre de circuits est sa grande souplesse, ce qui permet de le réutiliser à volonté dans des algorithmes différents en un temps très court. Le progrès de ces technologies permet de concevoir des composants toujours plus rapides et à plus haute Intégration, ce qui permet de programmer des applications importantes figure C.3.

Les circuits FPGA sont constitués d'une matrice de blocs logiques programmables entourée de blocs d'entrée sortie programmable. L'ensemble est relié par un réseau d'interconnexions programmable. Les FPGA sont bien distincts des autres familles de circuits Programmables tout en offrant le plus haut niveau d'intégration logique.

C.2. Description d'un FPGA

Un FPGA est constitué d'une matrice carrée de cellules configurables CLB (Configurables Logique Blocs) permettant de réaliser des fonctions combinatoires et des fonctions séquentielles. Tout autour de ces blocs logiques configurables, nous trouvons des blocs d'entrées /sorties IOB (Input Output Blocs) dont le rôle est de gérer les entrées-sorties réalisant l'interface avec les modules extérieurs. La figure C.1 présente l'architecture générale de ces circuits.

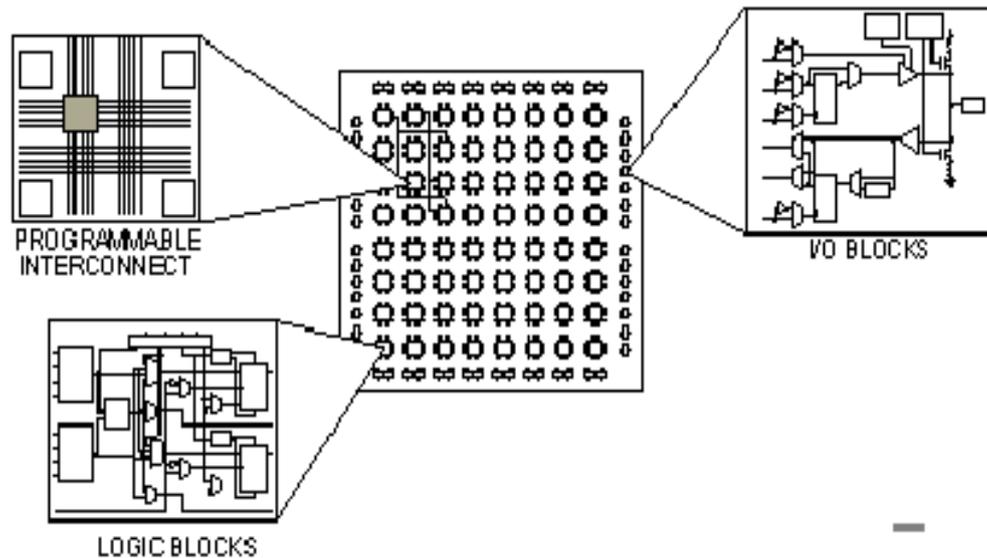


Figure C.1 : Architecture générale d'un FPGA

C.3. Les familles FPGA

Grâce aux évolutions de la technologie microélectronique les FPGA, deviennent de plus en plus performants avec des capacités sans cesse augmentées. Longtemps réalisés autour de blocs logique configurables à base de LUT (Look Up Table), les FPGAs peuvent aujourd'hui comporter de larges mémoires RAM configurables des opérateurs arithmétiques complexes (comme les block select RAM et les block multiplieur du VIRTEX-II de XILINX) et des cœurs de microprocesseurs (tel que le VIRTEX-II pro).

Les circuits FPGA de XILINIX [107] sont caractérisés par une nomenclature spécifique qui définit les performances de chaque famille. Cette nomenclature est la suivante :

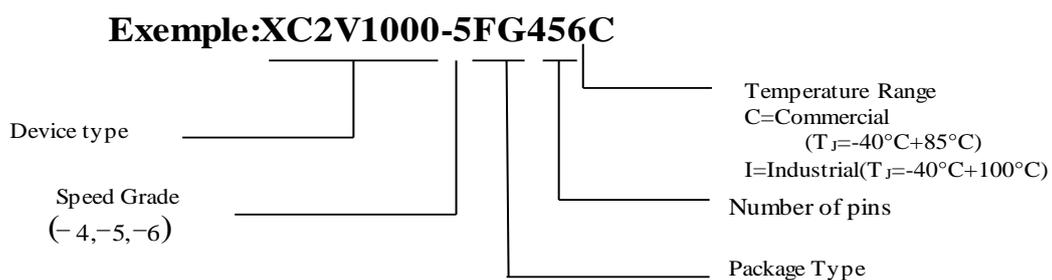


Figure C.2 : Nomenclature d'un FPGA de XILINIX

- ✚ V : "Device type" représente le type de famille, dans ce cas est : VIRTEX-II.
- ✚ -5 : "Speed Grade" représente la vitesse du composant selon la technologie utilisée dans la fabrication du circuit.

- ✚ 456 : "Number of pins "représente le nombre de pins donnée un FPGA.
- ✚ X : représente la compagnie de fabrication XILINX.
- ✚ G : "Package type" représente le type de boîtier.
- ✚ C : "Temperature Range" représente la gamme de température tolérée.

C.4. Les performances des carte FPGAs

Les performances des cartes FPGA évaluent ; une architecture peut être mise en œuvre de différentes façons en fonction de l'outil utilisé et de la plateforme FPGA.

Les plus récentes catégories d'outils ceux qui assurent uniquement le transfert de code C, C++, System C vers le VHDL ou RTL. Ces outils sont prévus pour le portage d'algorithme en FPGA avec pour objectif de diminuer le temps de développement en évitant la conception de bas niveau propre aux développements en VHDL. Cela représente un avantage pour la plupart des développements en général. C'est le cas, par exemple, pour la réutilisation de fonctions de traitement de signal (filtrage, corrélation, FFT...) déjà développées en langage C.

Enfin une dernière méthode de parallélisation sur FPGA consiste à utiliser l'outil générateur de blocs IPs, disponible avec l'outil de développement FPGA. Par exemple, le "Core Generator" de Xilinx Vivado permet de générer rapidement des fonctions de traitement de signal, des opérations arithmétiques et certains algorithmes d'encodage (Reed Solomon par exemple). Cette méthode requiert une bonne expertise de l'outil de développement FPGA. Là encore il reste à mettre en œuvre les autres blocs permettant d'accéder aux ressources physiques de la carte FPGA.

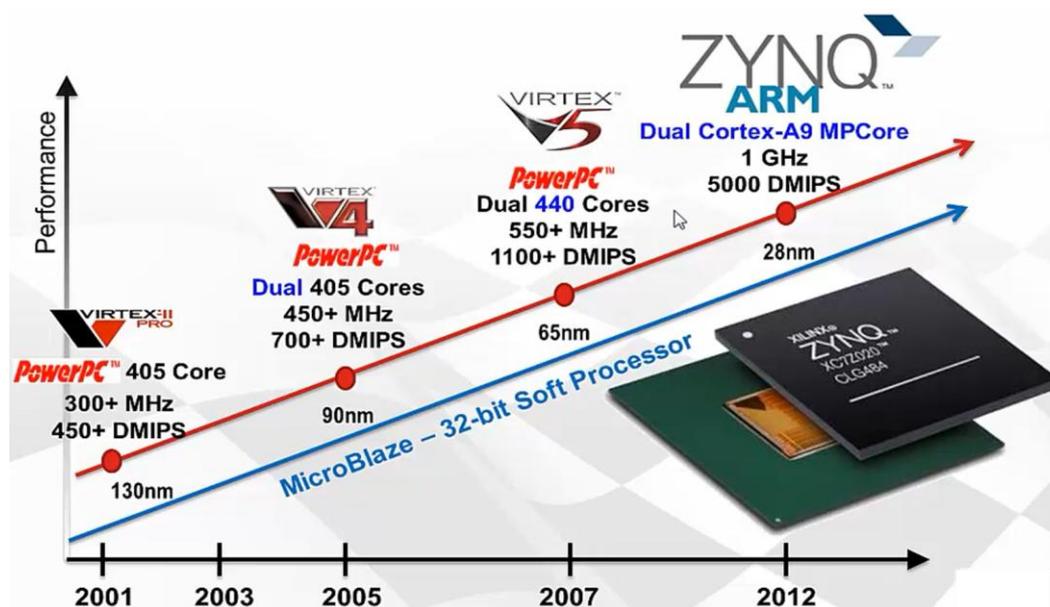


Figure C.3 : Performances des FPGA

C.5. La famille VIRTEX-II

La famille VIRTEX-II à vue le jour grâce à la technologie d'intégration CMOS de 8 couches de métal ($0.15\mu m/0.12\mu m$). Comparées aux familles FPGA, le VIRTEX-II est optimisé pour une haute densité et performance avec une grande capacité (jusqu'à 10 millions de portes). Actuellement elle est utilisée dans plusieurs applications tels que : la télécommunication, les réseaux, la vidéo et les applications DSP (Digital Signal Processing).

C.5.1. Architecture interne de VIRTEX-II

Comme tout circuit FPGA, le VIRTEX-II est composé de blocs entrées /sorties IOB, de blocs logiques internes configurables, des ressources d'interconnexion et des lignes d'interconnexions.

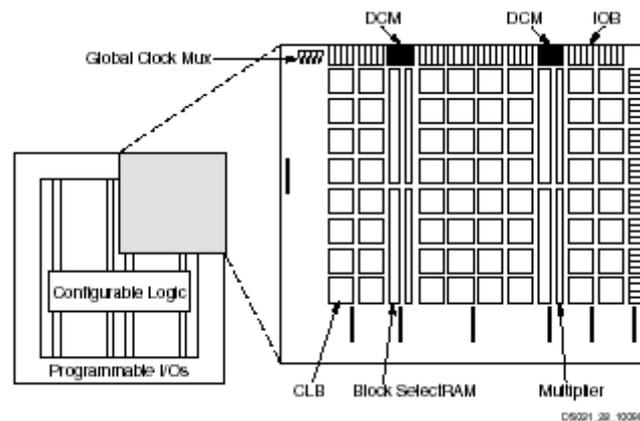


Figure C.4: Architecture interne d'un VIRTEX-II [108]

C.5.2 Le bloc logique interne configurable

Le bloc logique interne configurable inclut quatre éléments majeurs. [109]

- a) Multiplier bloc ($18\text{bits} \times 18\text{bits}$).
- b) Bloc select RAM 18 bits.
- c) DCM (**D**igital **C**lock **M**anager).
- d) Bloc logique Configurable (CLB).

C.5.3. Les éléments logiques

Dans les Virtex l'élément logique de grain le plus fin est appelé un LC (Logic Cell) , il est composé d'une LUT (Look Up Table) un registre (pour synchroniser si nécessaire la sortie) est d'une chaîne de propagation rapide de la retenue (de ce fait on peut réaliser un full adder 1 bit par LC). Au grain logique supérieur l'élément est appelé un SLICE celui ci est

composé de deux LC, voir appendice C, Puis les SLICE sont regroupés par deux pour former un CLB (Configurable Logic Blocks).

C.5.4. Bloc logique configurable (CLB)

Le CLB était l'élément déterminant des performances du FPGA. Mais pour le VIRTEX-II, ou on ne parle plus de CLB mais de slice, chaque CLB est composé de quatre slices (morceaux) et deux buffers à trois états.

Tous les slices sont identiques et contiennent :

- ✚ 2 générateurs de fonction (LUT F et LUT G).
- ✚ 2 éléments de stockage (bascule D).
- ✚ Des portes logiques arithmétique.
- ✚ Des multiplexeurs.
- ✚ Une chaîne rapide de logique de carry (retenue)
- ✚ Une chaîne cascadée des portes OR horizontale.

C.5.5. Bloc d'entrées/sorties (IOB)

Les blocs d'entrée/sortie fournissent l'interface entre les broches externes de module et la logique interne. Ils sont présents sur toute la périphérie du circuit FPGA.

C.5.6. Techniques de programmation des FPGA

Les circuits FPGA ne possèdent pas de programme résident. A chaque mise sous tension, il est nécessaire de les configurer. Leur configuration permet d'établir des interconnexions entre les CLB et les IOB. La configuration du circuit est mémorisée sur la couche réseau SRAM à partir d'une ROM externe. En effet, un dispositif interne permet à chaque mise sous tension de charger la SRAM interne à partir de la ROM. Ainsi on conçoit aisément qu'un même circuit puisse être exploité successivement avec des ROM différentes puisque sa programmation interne n'est jamais définitive.

Le format des données du fichier de configuration est produit automatiquement par le logiciel de développement sous forme d'un ensemble de bits organisés en champs de données.

C.5.7. Les étapes d'un développement en VHDL (Very High Speed Integrated Circuits Hardware Description Language)

C.5.7.1. Approche de conception

La conception, la description et la simulation des composants et systèmes en langage VHDL présentent deux aspects distincts pour un modèle.

- ✚ Une vue externe du circuit ; appelée « Spécification »
- ✚ Une vue interne du circuit : appelée « Architecture »

Chacune de ces deux parties sera une unité de conception et ce sont ces unités qui constituent la bibliothèque VHDL. Une bibliothèque peut compter des centaines d'unités de conception séparées qui permettent de modifier facilement le fonctionnement interne.

Le VHDL est un langage très modulaire, on peut ne pas écrire de longues descriptions, mais des unités plus petites et hiérarchisées, certaines parties de ces descriptions peuvent être compilées séparément ; Elles sont suffisantes en elles-mêmes pour être comprises. Ce sont les unités de conception. On peut dire que :

- ✚ Le circuit est considéré comme une entité (modèle)
- ✚ Cette entité peut être composée par d'autres entités.
- ✚ Chaque entité est composée de deux parties, interface et corps. Elle permet d'avoir plusieurs corps pour une même interface.

L'outil de simulation MODELSIM utilisé est l'un des logiciels de CAO les plus répandus, sur les plates formes de type PC pour la mise en œuvre des circuits programmables FPGA et qui réalise la simulation.

La conception se passe principalement en trois phases :

- ✚ La saisie du circuit.
- ✚ L'implémentation.
- ✚ La configuration du composant.

Auxquelles il faut ajouter les phases de vérification :

- ✚ La simulation fonctionnelle.
- ✚ La simulation temporelle.

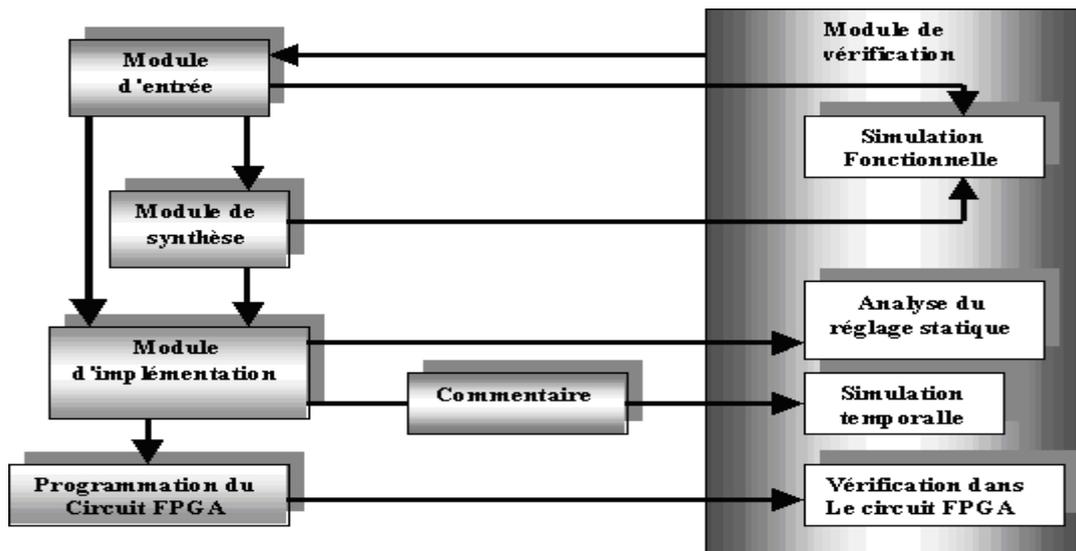


Figure C.5 phases de conception sur FPGA

C.5.7.2. Module d'entrée "design entry"

Ce module permet la description d'une architecture ou d'un design soit en code VHDL, utilisation d'un éditeur de schéma ou bien par les diagrammes d'état.

C.5.7.3. Module de synthèse "Design Synthesis"

Ce module transforme la description VHDL en portes logiques (description proche des ressources matérielles) et procède à l'optimisation : les signaux inutilisés sont retirés, les expressions booléennes sont simplifiées, les signaux équivalents sont détectés.

C.5.7.4. Module d'implémentation de "Design Implementation"

Il réalise le mapping, c'est la projection des équations sur les différents blocs de circuit. Ainsi, il réalise le placement/ routage qui consiste à attribuer les blocs du circuit à chaque équation délivrée par la projection et à définir les connexions.

L'algorithme de placement place physiquement les différentes cellules et définit les chemins d'interconnexions dessinés entre les cellules afin de faciliter le routage.

C.5.7.5. Module de vérification « Design Verification »

Avec ce module, nous pouvons analyser le temps d'exécution (static timing analysis), et réaliser la simulation. Cette dernière déroule en parallèle chaque bloc pour vérifier que le design est opérationnel.

C.6. La carte Zedboard

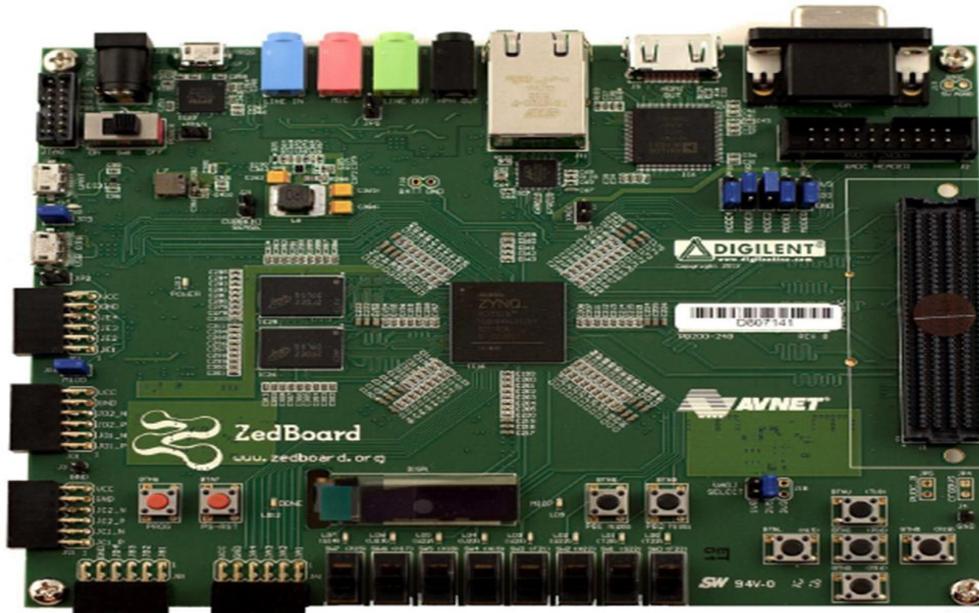


Figure C.1 : La carte Zedboard.

La carte Zedboard est une carte de développement conçue par Digilent. Les systèmes on chip (SOC) ZYNQ de la société xilinx intègre :

- Un système de traitement PS (processing system) basé sur un processeur double cœur ARM cortex A9 capable d'accueillir un système d'exploitation comme LINUX . On appelle PS la partie processeur et les périphériques associés cela inclus : les deux cœurs ARM cortex A9, les bus AMBA et AXI, le DMA, les GPIOs, I2C, UART, CAN, SPI, le contrôleur des mémoires QuadSPI, NAND et NOR et le contrôleur de mémoire vive.

- Un système de programmation logique PL (programmable logic) avec un FPGA de la série XILINX-7. La PL, contient les éléments logiques de base, la RAM, des DSP et les entrées sorties standard [110].

➤ Présentation de la partie PS du SoC Zynq

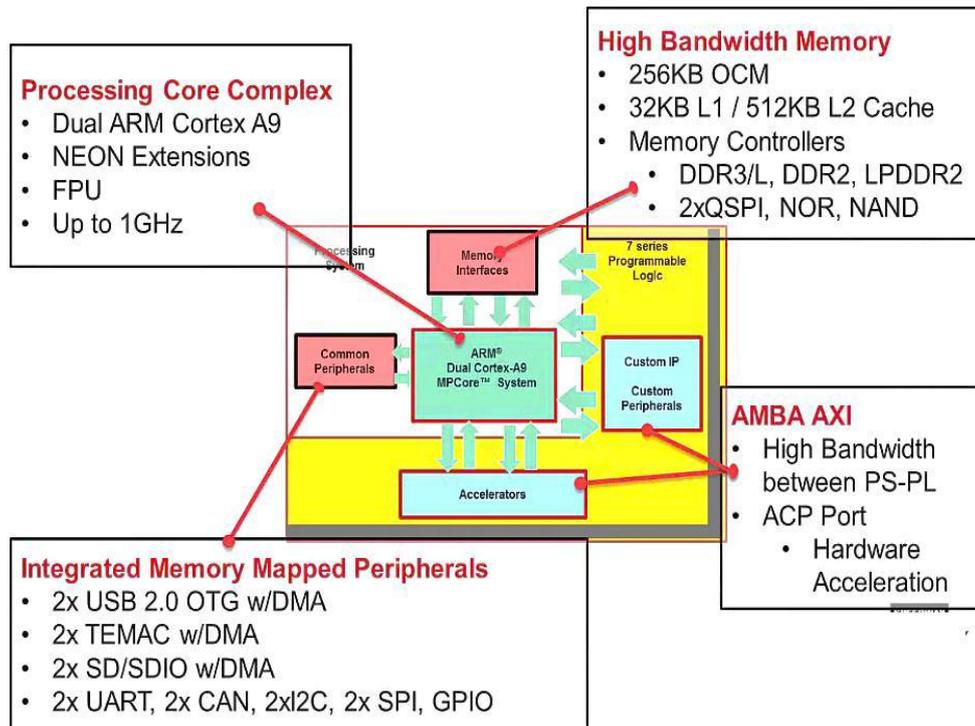


Figure C.7: Présentation des processeurs du SoC Zynq.

C.6.1 Flot de conception entre la partie PS et PL

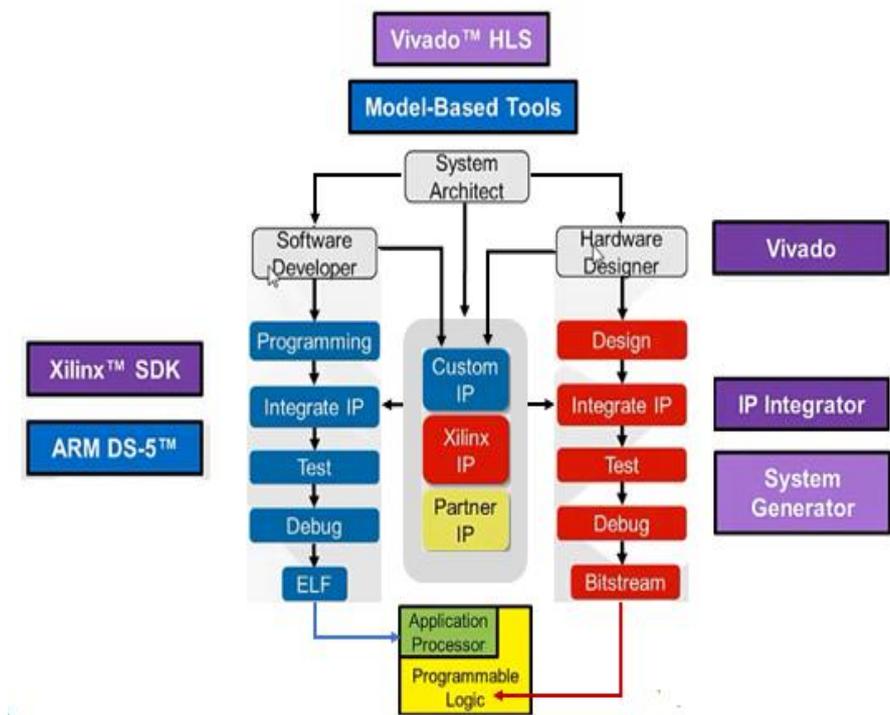


Figure C.8 : Flot de conception entre la partie PS et PL [107].

Nous pouvons voir sur la figure C.8 les différents périphériques présents sur la Zedboard.

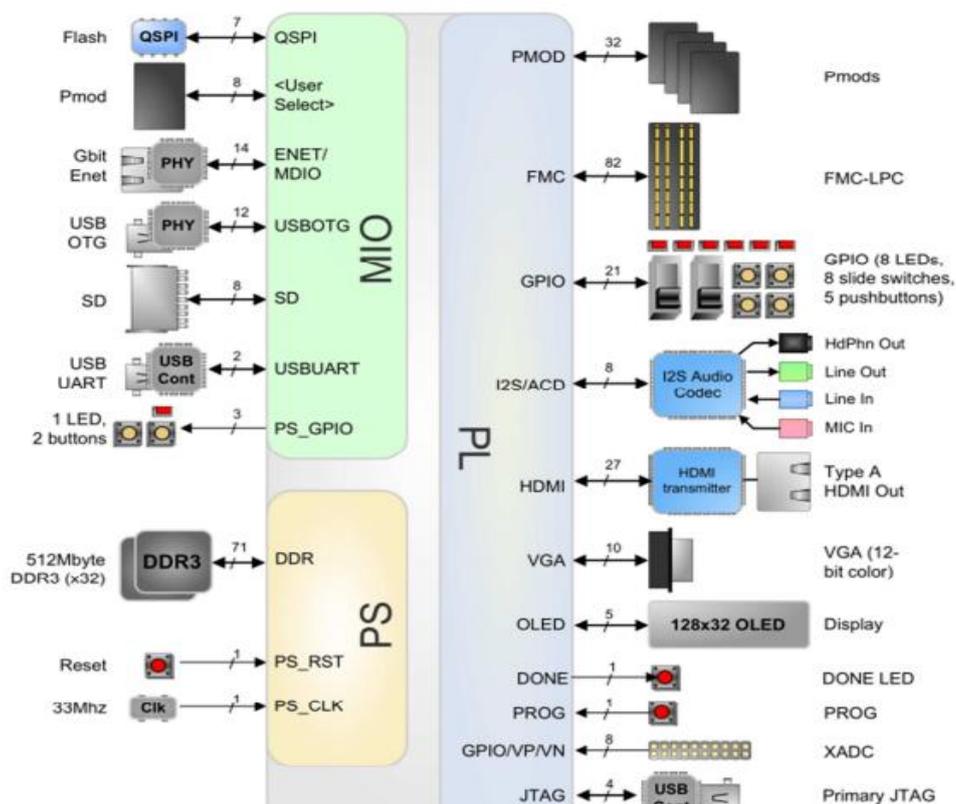


Figure C.9 : Les Périphériques de la carte [110].

Au niveau connexion, la Zedboard est assez complète et permet l'interfacage de nombreux éléments :

- HDMI (Audio et vidéo)
- VGA (Vidéo)
- Ethernet (10/100/1000 Mbps)
- Port de débogage ARM Debug Access Port (DAP)
- Port USB 2.0 OTG (Device, Host et OTG)
- Port de programmation USB-JTAG
- Port USB-UART
- Connecteur FMC.

Les composants IHM sont également assez nombreux :

- 9 LEDs
- 8 interruptrices glissières
- 7 boutons poussoirs + 2 boutons Reset

Les mémoires intégrées, les entrées et sorties vidéo et audio, l'interface USB double rôle et Ethernet, ainsi que le slot SD simplifient au maximum la phase de conception sans matériel supplémentaire. En outre, six connecteurs Pmod facilitent toutes les conceptions [110].

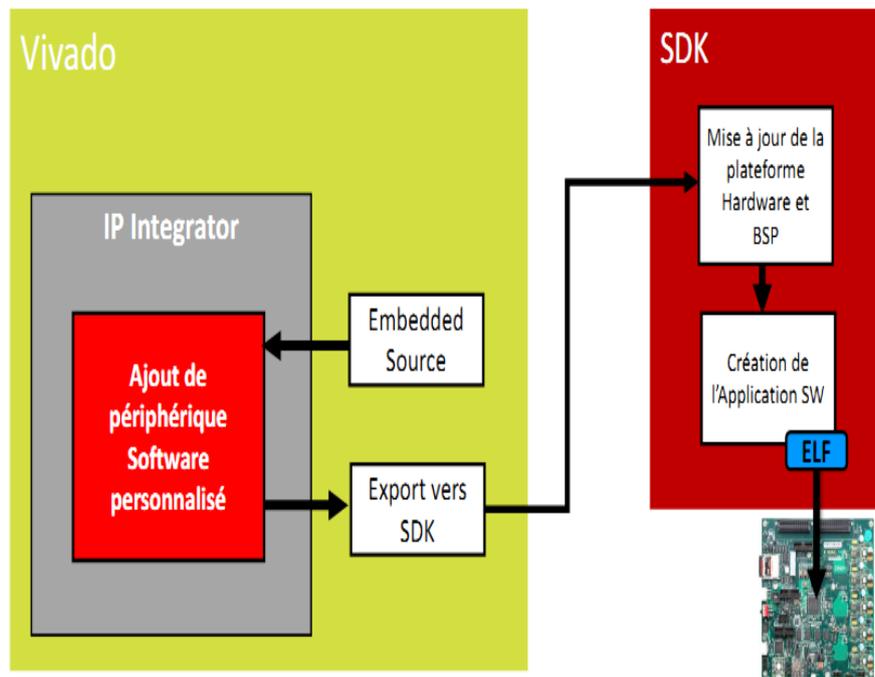


Figure C.10 : exemple d'ajout d'un périphérique PL *custom*

Ils existent différents modèles de ZYNQ dont les caractéristiques sont résumées ci-dessous:

Zynq-7000 AP SoC Devices		Z-7010	Z-7015	Z-7020	Z-7030	Z-7045	Z-7100
Processing System	Processor Core	Dual ARM® Cortex™-A9 MPCore™					
	Processor Extensions	NEON™ & Single / Double Precision Floating Point					
	Max Frequency	866 MHz					
	Memory	L1 Cache 32KB I / D, L2 Cache 512KB, on-chip Memory 256KB					
	External Memory Support	DDR3, DDR3L, DDR2, LPDDR2, 2x QSPI, NAND, NOR					
	Peripherals	2x USB 2.0 (OTG), 2x Tri-mode Gigabit Ethernet, 2x SD/SDIO, 2x UART, 2x CAN 2.0B, 2x I2C, 2x SPI, 4x 32b GPIO					
Programmable Logic	Approximate ASIC Gates	~430K (28k LC)	~1.1M (74k LC)	~1.3M (85k LC)	~1.9M (125k LC)	~5.2M (350k LC)	~6.6M (444k LC)
	Block RAM	240KB	380KB	560KB	1,060KB	2,180KB	3,020KB
	Peak DSP Performance (Symmetric FIR)	100 GMACS	200GMACS	276 GMACS	593 GMACS	1334 GMACS	2662 GMACS
	PCI Express® (Root Complex or Endpoint)	-	Gen2 x4	-	Gen2 x4	Gen2 x8	
	Agile Mixed Signal (XADC)	2x 12bit 1Msp A/D Converter					
I/O	Processor System IO	100					
	Multi Standards 3.3V IO	100	150	200	100	212	250
	Multi Standards High Performance 1.8V IO	-	-	-	150	150	150
	Multi Gigabit Transceivers	-	4	-	4	16	16

Figure C.11 : Caractéristique des différents modèles de la ZYNQ.

La carte est compatible avec la nouvelle solution haute performance Vivado Design Suite de Xilinx, ainsi que le jeu d'outils ISE.

REFERENCES

1. Chateil J-F, Hazebroucq V, « Dossier médical informatisé ,archivage des images : législation et recommandation » Bulletin de la société Française de radiologie, vol.21,p 11-13, avril 2005.
2. A.W. Wong, R.K. Taira, and H.K. Huang, “Implementation of a digital archive system for a radiology department”, in Proc. SPIE Conf. on Medical Imaging VI: PACS Design and Evaluation 1645, pp. 182-190, 1992.
3. B. J. Erickson, “Irreversible compression of medical images”, Journal of Digital Imaging, Vol. 15, No. 1, March 2002, pp. 5-14.
4. Yann-Gaudeau. «: Contributions en compression d’images médicales 3D et d’image naturelles 2D. » thèse université Henri Poincaré Nancy décembre 2006.
5. A.Naït-ali et C.Cavaro-Menard, “ Compression des images et des signaux médicaux”, Lavoisier, Hermes Sciences publications 2007.
6. F. Alim-Ferhat-Taleb : « Implémentation sur FPGA d’algorithme en ligne d’analyse spectrale et de filtrage numérique », thèse magistère université de Blida septembre 2005.
7. H. Bessalah, F. Alim, M. A. Bencherif, S. Seddiki, “Toward an Embedded Image Wavelet Transform Implementation Approach”, Shakerverlag; Transactions on Systems, Signals & Devices Vol. 2, No. 3, TSSD 1861-5252/ c 2006.
8. T. Heskes, “Self-Organizing Maps, Vector Quantization, and Mixture Modelling”, IEEE transaction on neural networks, Vol. 12, No. 6, November 2001.
9. W. Kurdthongmee, “A novel Kohonen SOM-based image compression architecture suitable for moderate density FPGAs”, Elsevier image and vision computing, 2008.
10. Muthuramalingam, S. Himavathis, E. Srinivasan, “neural network implementation using FPGA: issues and application”, International journal of information technology, Vol. 4, No.2, ISSN 1305-2403, 2007.
11. David Salomon, “Data Compression Third Edition”, Springer-Verlag London Limited 2007
12. O.Hammami, “Etudes d’optimisation algorithmique de JPEG2000”, ministère de l’industrie dans le cadre du réseau national de la recherche en télécommunication (RNRT), février 2004.
13. D. Huffman, "A method for the construction of minimum redundancy codes", in proceedings of IRE, pages 1098–1101.1948.

14. D. Duttweiler and C. Chamzas, "Probability estimation in arithmetic and Adaptive-Huffman entropy coders", IEEE transactions on image processing, Vol. 4, No. 6, pp 237–246, June 1995.
15. P. Howard and J. Vitter, "Practical Implementations of Arithmetic Coding", Appears in Image and text Compression, James A. Storer, ed., Kluwer Academic Publishers, Norwell,MA, pp 85-112,1992.
16. J. Marconi M. Rodrigues, "Transfert Sécurisé D'images par Combinaison de Techniques de Compression, Cryptage et Marquage", thèse doctorat de l'université Montpellier II Mention Informatique, 31 Octobre 2006.
17. A. N,Natarajan T,Rao K.R, " Discret cosine transform", IEEE Signal Processing Magazine, vol.14n°2,p.24-41,1974.
18. M. Barlaud, "Wavelets in image communication, Advances in Image Communication, Elsevier, 1994.
19. I. Daubechies, "Ten Lectures on Wavelets, CBMS-NSF Regional Conference Series in Applied Mathematics, Vol. 61, Philadelphia, USA, PA: SIAM, 1992.
20. F. Friedrich H. Führ, L. Demaret, "Beyond wavelets: New image representation Paradigms",in Survey article, preprint version, 2005.
21. M. Barlaud, C. Labit, "Compression et codage des images et des vidéos, Traité IC2, série Traitement du Signal et de l'Image, Hermes Sciences Publications, 2002.
22. J.F. Musy,E .Bacry,A.Arneodo," multifractal formalism for fractal signals:the structure function approach versus the wavelet transform modulus-maxima method. Physical review E,47(2) ;875.1993.
23. F. Davoine, M. Antonini, J.M. Chassery and M. Barlaud, "Fractal image compression based on Delaunay triangulation and vector quantization", IEEE Transactions on Image Processing, vol. 5, 2, p. 338-346, February 1996.
24. F. Davoine, J.M. Chassery, A. Baskurt and F. Dugelay, "Codage par Transformation Fractale, chapitre 7, " Compression et codage des images et des videos, Traité IC2, M. Sciences publications, 2002.
25. E. Pennec and S. Mallat, "Image compression with geometrical wavelets", IEEE International Conference on Image Processing, ICIP2000, Vancouver,BC, Canada, Vol. 1, pp. 661-664, 2000.
26. E.Le Pennec and S. Mallat, Geometrical Image Compression with Bandelets, Society for Industrial and Applied Mathematics, Vol. 4, No. 3, pp. 992–1039;2005
27. E. L. Pennec and S. Mallat, "Sparse geometric image representation with bandelets," IEEE Trans. Image Proc., vol. 14, no. 4, pp. 423–438, April 2005.

28. N. M. Nasrabadi and R. A. King, "Image coding using vector quantization": review, IEEE Transactions on Communications, pp. 957-971, 1988
29. M. Barlaud, P. Sole, T. Gaidon, M. Antonini, and P. Mathieu, "Pyramidal lattice vector quantization for multiscale image coding", IEEE Transactions on Image Processing, vol. 3, pp. 367-381, July 1994.
30. Jain A.K, « Image data compression, Proceeding of IEEE, vol 69, n°3 p349-389,1981.
31. Gray R.M, "Vector quantization" IEEE ASSP Magazine, vol.1, p4-29,1984.
32. A. Gersho et Robert M. Gray. "Vector quantization and signal compression". Kluwer Academic,1992.
33. J.H Conway and N.J.A. Sloane, "Fast quantizing and decoding algorithms for lattice quantizers and codes", IEEE Transactions on Information Theory, vol. 28, 2, pp. 227-232, March 1982.
34. Robinson J.A," Efficient general purpose image compression with binary tree predictive coding "IEEE Transaction on image processing, vol 6, n° 4, p 601 608 ,1997.
35. Said A, Pearlman W.A, "Image multi resolution representation for lossless and lossy compression ", IEEE Transaction on image processing, vol 5, n° 9, p 1303-0310. 1996.
36. Tischer P.E, WorleyR.T, Maeder A.J, Goodwin M "Context based lossless image compression ", The computer journal, vol .36 n°1 p 68-77, 1993.
37. Wu X, Memon N.D "context based adaptive lossless image compression " IEEE transaction on information theory , vol .23, p 337-343, 1997.
38. Huang L, Bijaoui A, "An efficient image compression algorithm without distorsion“, pattern Recognition Letters, vol 12, p 69-72, 1991.
39. N. Akrouf, R. Prost, et R. Goutte. "Image compression by vector quantization: a review focused on codebook generation ". Image and vision computing, 12(10):627–637, December 1994.
40. T-Chuen Lu "A Survey of VQ Codebook Generation" Journal of Information Hiding and Multimedia Signal Processing; ISSN 2073-4212; Volume 1, Number 3, July 2010.
41. C. Foucher, "Algorithmes neuronaux et non neuronaux de construction de dictionnaire pour la quantification vectorielle en traitement d'images", Thèse en traitement du signal ; Université de Rennes, Décembre 2002.
42. Y.Linde,A.Buzo,and R.M.Gray, "An Algorithm for Vector Quantizer Design", IEEE Transactions on Communications,January 1980

43. T. Kohonen, "The self-organizing map", Proc. of the IEEE,78(9), pp. 1464-1480. September 1990.
44. M. Loève, "Fonctions Aléatoires de Second Ordre", Processus stochastiques et Mouvement Brownien, P. Lévy, Ed. Hermann, Paris, 1948.
45. V. Bristanak, P. Yip, K. R.Rao, "Discrete cosine and sine transforms", Elseiver, Amsterdam, 2007.
46. Y.Meyer, "Méthodes Temps-Frequence et Temps-Echelle en Traitement du Signal et de l'Image", INRIA, 1991.
47. P.Flandrin, N. Martin, "Analyse Temps-Frequence et Temps-Echelle", Traitement du Signal et Image, Vol. 9, No.1, 1992.
48. I. Daubechies, "Othonormal bases of Compactly Supported Wavelet", com. pure Appl.Math.Vol.41, pp. 909-996, November 1988.
49. S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, pp. 674-693, July 1989.
50. S. Mallat, 'Une exploration des signaux en ondelettes, Editions de l'Ecole Polytechnique, 2000.
51. M. Barrat, O. Lepetit, " Calcul Rapide de la Transformée en Ondelettes", Traitement du Signal et Image. Vol. 8. No.1 pp.43-49, 1991.
52. Said A, Pearlman W.A, "Image multi resolution representation for lossless and lossy compression ", IEEE Transaction on image processing, vol 5, n° 9, p 1303-0310. 1996.
53. A. Cziho, "Quantification vectorielle et compression d'image. Application à l'imagerie médicale", Thèse de doctorat, Université de Rennes1, Mai 1999.
54. José Marconi M. Rodrigues, "Transfert Sécurisé D'images par Combinaison de Techniques de Compression, Cryptage et Marquage", thèse doctorat de l'université Montpellier II; Mention Informatique, 31 Octobre 2006.
55. Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity", IEEE Transactions on Image Processing, pp. 600- 612, Vol. 13, no. 4, Apr. 2004.
56. Z. Wang and Eero P. Simoncelli, "Translation insensitive image similarity in complex wavelet domain", IEEE Inter. Conf. Acoustic, Speech & Signal Processing Volume II, Pages 573-576, March 2005.

57. Ramakrishnan B, Sriraam N, "Internet transmission of DICOM images with effective low bandwidth utilization", Digital Signal Processing, 2006.
58. J.Giron, F.Joffre. « Bases physiques et évolution de l'imagerie radiologique, Abrégés d'imagerie Radiologique, 1993.
59. P.Grangeat, "La tomographie médicale: imagerie morphologique et imagerie fonctionnelle, Traité IC2, Série traitement du signal et de l'image, Hermès, Paris, 2002.
60. J.T.Bushberg, J.A Seibert and al, "The essential Physics of Medical Imaging, 2nd Edition, Lippincott Williams and Wilkins, 2001.
61. B.Kastler, D.Vetter, Z.Pattay, P.Germain, « Comprendre L'IRM », 5e Edition, Collection imagerie médicale, diagnostic, Masson, Paris, 2003.
62. S.R.Cerry, J.A.Sorenson, M.E, Phelps. « Physics in Nuclear Medicine, 3rd Edition, Saunders Elsevier Science, Philadelphia, 2003.
63. J.F, Chateil, V.Hazebroucq. " Dossier médical informatisé (DMP), archivage des images : législation et recommandations » Bulletin de la société française de radiologie, 2005.
64. Kivijarvi J, Ojala T, Kaukoranta T, Kuba A, Nyul L, Nevalainen O, " A comparison of lossless compression methods for medical images" Computerized medical imaging and graphics vol 22, p323-339, aout 1998.
65. Clunie D.A, "Lossless compression of grayscale medical images-Effectiveness of traditional and state of the art approaches", Proceedings of SPIE .Medical imaging 2000, vol.3980, p.74-84, 2000.
66. Adamson C, Lossless compression of magnetic resonance imaging data, thesis Monash University, Melbourne, Australie, 2002.
67. Y. Gaudeau. Contributions en compression d'images médicales 3D et d'images naturelles 2D. PhD thesis, Université Henri Poincaré, Nancy 1, 2006.
68. S-G Miaou and Shih-Tse Chen. Automatic quality control for wavelet-based compression of volumetric medical images using distortion-constrained adaptive vector quantization. Medical Imaging, IEEE Transactions on, 23(11):1417-1429, Nov. 2004.
69. S.-C.B. Lo, Huai Li, and M.T. Freedman. "Optimization of wavelet decomposition for image compression and feature preservation". Medical Imaging, IEEE Transactions 1141-1151, Sept. 2003.
70. Yung-GiWu and Shen-Chuan Tai. Medical image compression by discrete cosine transform spectral similarity strategy. Information Technology in Biomedicine, IEEE Transactions on, 5(3) :236-243, Sept. 2001.
71. D. Koff, P. Bak, P. Brownrigg, D. Hosseinzadeh, A. Khademi, A. Kiss, L. Lepanto, T. Michalak, H. Shulman, and A. Volkening. Pan-canadian, " evaluation of irreversible

compression ratios ("lossy" compression) for development of national guidelines", *Journal of Digital Imaging*, 2008.

72. L'Association canadienne des radiologistes CAR. "Normes de la CAR en matière de compression irréversible pour l'imagerie numérique diagnostique en radiologie", 2008.

73. C. Delgorge, C. Rosenberger, G. Poisson, and P. Vieyres. "Towards a new tool for the evaluation of the quality of ultrasound compressed images". *Medical Imaging, IEEE Transactions on*, 25(11) :1502–1509, Nov. 2006.

74. Xie Kai, Yang Jie, Zhu Yue Min, and Li Xiao Liang. "Hvs-based medical image compression". *European Journal of Radiology*, 55(1) :139–145, July 2005.

75. B.J. Erickson, "Irreversible Compression of Medical Images", *Journal of Digital Imaging*, vol. 15, 1, pp. 5-14, March 2002.

76. A.Naït-ali et C.Cavaro-Menard, "Compression des images et des signaux médicaux", Lavoisier, Hermes Sciences publications 2007.

77. A. Naït-Ali and Christine Cavaro-Ménard, editors. *Compression of Biomedical Images and Signals*. ISTE / WILEY, 2008.

78. E. Riskin, R. Gray and al. "Variable rate vector quantization for medical image compression", *IEEE Transaction on Medical Imaging*, vol 9, p.290-298, 1990.

79. N. Mohsenian, H. Shahri, N. Nasrabadi, "Scalar vector quantization of medical images", *IEEE Transaction on Medical Imaging*, vol 12, p.727-739, 1993.

80. J. Han, H. Kim, "A differential index assignment scheme for tree-structured vector quantization", *IEEE Transaction on Medical Imaging*, vol 18, p 442-447, 1999.

81. Y. Wu, "GA-based DCT quantisation table design procedure for medical images", *IEEE proceeding, image signal processing*, vol 151, p353-359, 2004.

82. S. Grgic, M. Grgic, B. zoko-cihlar, "performance analysis of image compression using wavelets", *IEEE Trans, industrial electronics*, Vol. 48, No.3, June 2001.

83. Z. Xiong, K. Ramchandran, M. T. Orchard and Y. Zhang, "a comparative study of DCT and Wavelet based image coding", *IEEE Trans. Circuits and systems for video technology*, Vol.9, No. 5, Aug. 1999.

84. A.N. Akansu, W. A. Serdijn, I. W. Selesnick, "Emerging applications of wavelets: A review", *Elsevier Physical Communication*, Vol. 3, No. 1, pp 1-18, Mars 2010.

85. Rapport de recherche « Codage conjoint source canal appliqué à la transmission d'images, robustesse d'une chaîne de transmission », laboratoire SIC, université de Poitiers ; octobre 2005.

86. C. Delgorge. « Proposition et Evaluation de techniques de compression d'images ultrasonores dans le cadre d'une télé-échographie robotisée. thèse université d'Orléans décembre 2005.
87. Matlab, "Wavelet Toolbox: Wavelets: A New Tool for Signal Analysis: Five Easy Steps to a Continuous Wavelet Transform", Mathworks.
88. P.Ravier, « Cours d'ondelettes », Espeo-université d'Orléans 2000.
89. S.Mallat, « A theory for multiresolution signal decomposition: The wavelet representation", IEEE transactions on pattern Analysis and Machine Intelligence, Vol II, N°07 July 1989.
90. A. Lewis, G. Knowles "Image Compression Using the 2-D Wavelet", IEEE Transactions on Image Processing, Vol. 1 No. 2. pp. 244-250. 1992.
91. H.Bessalah, « Application of the half line mode of computation to digital signal processing»,the proceedings of the 7 th international conference on signal processing application et technology. Boston, Massachusetts, USA October 1996.
92. M. D. Ercegovac, "On-line Arithmetic for Recurrence Problems", SPIE vol. 1556. Signal.Processing (1991).
93. A.Avizienis, «Signed-Digit number representations for fast parallel arithmetic», Transactions on Electronic Computers, p:389–400, September 1961.
94. H. Bessalah, «High half line VLSI computation for images compression »the sixth international conference on microelectronics. Marmara research Center Istanbul Turkey 1994.
95. H.Bessalah, F.Alim, S.Seddiki « Implementation of the Hough transform by the on line mode »Vipromcom 02; 4th EURASIP IEEE region 8 international symposium on video/image processing and multimedia communication Zadar ,Croatia.2002.
96. H. Bessalah, F. Alim, M. A. Bencherif, S. Seddiki, "Toward an Embedded Image Wavelet Transform Implementation Approach", SHAKERVERLAG; Transactions on Systems, Signals & Devices Vol. 2, No. 3, TSSD 1861-5252/ c 2006.
97. H. Bessalah, F.Alim Ferhat-Taleb, S.Seddiki, O.Kerdjidj, "On Line Wavelets Transform on a Xilinx, FPGA Circuit to Medical Images Compression", International Conference on Digital Image Processing (ICDIP), Thailand March 2009.
98. F. Alim Ferhat-Taleb, H. Bessalah, S. Seddiki", WT-SOM network implementation on FPGA for the medical images compression", Proceedings of the 5th international conference on Soft computing as trans disciplinary science and technology CSTST '08: ISBN: 978-1-60558-046-3, Paris ,2008.
99. F.Alim.Ferhat, H.Bessalah S.Seddiki, O.Kerdjidj H.Salhi, M.Issad Yakhlef "Wavelet Transform and Neural Network combined method for medical images Compression and

implementation on FPGA ". The Mediterranean Journal of Computers and Networks, Vol. 5, No. 1, pp 34-41, 2009.

100. <http://www.medeisa.net/>

101. F.Alim.Ferhat-Taleb;H.Bessalah;S.Seddiki;O.Kerdjidj, "Optimal Implementation on FPGA'S based on wavelet transform and self organizing map for medical images compression ", the Journal of Electric Engineering : ISSN :1582-4594 , Edition : 1;Volume 13 /2013.

102. Hong Wang, Ling Lu, Da-shun Que, Xun Luo," Image compression based on wavelet transform and vector quantization", Proc. IEEE Conf.on machine learning and cybernetics,Beijing, Ieeexplore 0-7803- 7508-41021. November 2002.

103. T. Denk, K. K. Parhi, V. Cherkassky, "combining neural networks and the wavelet transform for image compression", IEEE ISBN: 0-7803-0946-4, 1993.

104. M. Pormann., H. Kalte, U. Witkowski, J. C. Niemann, U. Ruckert, "A dynamically reconfigurable hardware accelerator for self-organizing feature maps", Proc. 5th World Multi-Conference, Vol. 3, Orlando, Florida, USA, pp. 242-247. July 2001.

105. J. Grando, M. A. Vega, R. Perez, J. M. Sanchez, J. A. Gomez, "Using FPGAs to Implement Artificial Neural Networks",IEEE 1-4244-0395-2/06/2006.

106. C.Diou, « Contribution à l'intégration sur silicium de la transformée en ondelettes application au traitement d'images », Thèse de doctorat université de Montpellier II Décembre. 2002.

107. <http://www.xilinx.com>

108. Xilinx, "Logic Core IP Multiplier v 11.2", Sept 2009.

109. Xilinx, "Block Memory Generator v 2.8 Logic Core", Sept 2008.

110. ZedBoard(Zynq™ Evaluation and Development) Hardware User's Guide,Version 1.1, 01-08-2012.

111. Wim Vanderbauwhede Khaled Benkrid, High-Performance Computing Using FPGAs, Edition © 2013 by Springer Science Business Media

112. David Money Harris and Sarah L. Harris, Digital Design and Computer Architecture, Second Edition © 2012 by Elsevier Inc.