

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



ECOLE NATIONALE POLYTECHNIQUE
Département d'Electronique

Projet de fin d'études
En vue de l'obtention du diplôme d'ingénieur d'état en Electronique

Thème

Quantification vectorielle des images DICOM

Présenté par :

Mr. BENAHMED DAHO Omar

Mr. EL BECHIR Salim

Dirigé par :

Mme. F.ALIM (CDTA)

Mr. S.SEDDIKI (CDTA)

Pr. D.BERKANI (ENP)

Organisme d'accueil : Centre de Développement des Technologies Avancées
Equipe : A3SP



Promotion: 2009
ENP, 10 Avenue Hassen Badi, El-Harrach, Alger.

ملخص

لقد قمنا في مشروعنا باختزال خوارزميتين لضغط الملفات الناتجة عن معيار التصوير الطبي DICOM. فأول خوارزمية تقريبا بدون خسارة وهي قائمة على طريقة RLE. أما الثانية هي بخسارة، أين أجرينا علاج أولي بواسطة المحولة بالتمويجات حيث أدرجت النتائج المتحصل عليها في الشبكة العصبية Kohonen من أجل إقامة قواميس تستعمل للتكميم. أجريت عملية الاختزال بواسطة Matlab.

الكلمات المفتاحية : التصوير الطبي – DICOM – المحولة بالتمويجات – علم الضغط – RLE – التشفير الحسابي – التكميم – Kohonen.

Résumé

Dans notre projet, nous avons réalisé une implémentation de deux algorithmes de compression de fichiers issus de la norme en imagerie médicale DICOM. Le premier est un algorithme quasiment sans pertes basé sur la méthode RLE. Le second est un algorithme avec pertes dans lequel nous avons effectué un prétraitement à l'aide de la transformée en ondelettes dont les résultats ont été injectés dans un réseau neuronal de Kohonen pour la création de dictionnaires servant à des quantifications scalaire et vectorielle. L'implémentation a été réalisée avec Matlab.

Mots clés : Imagerie médicale – DICOM – Transformée en ondelettes – Compression – RLE – Codage arithmétique – Quantification – Kohonen.

Abstract

In our project, we have carried out an implementation of two compression algorithms that dealt with the DICOM medical imagery standard. The first one is a nearly-lossless algorithm based on the RLE method. The second is a lossy algorithm in which we have conducted a pre-processing using the wavelet transform whose results were injected into a Kohonen neural network for the creation of codebooks used for scalar and vector quantization. The implementation was carried out using Matlab.

Keywords: Medical imaging – DICOM – Wavelet transform – Compression – RLE – Arithmetic coding – Quantization – Kohonen.

Remerciements

Nous tenons tout d'abord à remercier DIEU, tout puissant, de nous avoir donné la force de réaliser nos projets tout au long de ces années.

Nous remercions spécialement Mme F.ALIM et Mr S.SEDDIKI, Chercheurs au Centre de Développement des Technologies Avancées, de nous avoir offert la possibilité de réaliser ce projet au sein de leur équipe et de nous avoir dirigés durant notre stage. Nous remercions aussi Mr O.KERDJIDJ, membre de l'équipe A3SP du CDTA pour l'aide qu'il nous a apporté pendant la réalisation de ce projet.

Nos remerciements vont aussi à Pr D.BERKANI, Professeur à l'Ecole Nationale Polytechnique, de nous avoir donné les bases essentielles pour pouvoir réaliser ce projet et pour les conseils qu'il nous a donnés en tant que co-promoteur.

Nous tenons à remercier Pr R.AKSAS, Professeur à l'Ecole Nationale Polytechnique d'avoir accepté de présider le jury.

Nous remercions Dr R.ZERGUI, d'avoir accepté d'examiner notre travail.

Nous remercions tous les enseignants de l'Ecole Nationale Polytechnique d'Alger, spécialement ceux des départements des Sciences Fondamentales et d'Electronique pour le savoir qu'il nous ont permis d'acquérir et pour l'esprit d'analyse qu'ils nous ont permis de développer.

Nous tenons aussi à remercier toute personne dévouée au service de l'Ecole Nationale Polytechnique.

Enfin, nous remercions toute personne ayant contribué, de près ou de loin, à l'accomplissement de ce projet.

Dédicaces

Je dédie ce travail à ma très chère mère qui a toujours été derrière moi pour me guider dans la bonne direction, à mon père pour ses précieux conseils, à ma sœur et mon frère, mon beau frère et ma belle sœur qui ont toujours été là pour me soutenir durant mon parcours.

Je dédie, aussi, ce travail à tous les membres de ma famille, grands et petits, pour leur soutien.

Je dédie ce travail à tous mes amis qui ont toujours été là pour moi : Khalil, Karim, Khaled, Yanis, Charaf, Anis, Farouk, Omar, Rabie, Krimo, Yasmine et Maïly...

Enfin, Une Dédicace Spéciale à ma petite nièce adorée Ferial.

Salim

À mes chers parents, puisse dieu me les garder.

À ma chère sœur et mes frères.

À la mémoire de mon défunt oncle « Ammi DAHO ».

À la mémoire de mon défunt cousin « Kadouri ».

À tous ceux qui pensent à moi.

Omar

SOMMAIRE

Liste des figures	
Liste des tableaux	
Index	
Introduction générale.....	1

Chapitre I

Généralités sur la compression d'images

I.1 Introduction.....	3
I.2 La compression.....	4
I.2.1 Les techniques sans distorsions.....	4
I.2.1.1 Codage de Huffman.....	5
I.2.1.2 Codage arithmétique.....	6
I.2.1.3 Codage par plage (RLE).....	8
I.2.2 Les techniques avec distorsions.....	9
I.2.3 Critères d'évaluation.....	9
I.3 Compression des images médicales.....	12
I.4 Principe générale d'une méthode de compression.....	12
I.4.1 La transformation (La transformée en Ondelettes).....	13
I.4.2 La quantification.....	17
I.4.2.1 La quantification scalaire.....	17
I.4.2.2 La quantification vectorielle.....	18
I.4.2.3 Le dictionnaire.....	19
I.4.2.3.1 Construction du dictionnaire.....	19
I.4.2.3.1.1 Les algorithmes non-neuronaux.....	19
I.4.2.3.1.2 Les algorithmes neuronaux.....	21
I.4.3 Le codage.....	26
I.5 Conclusion.....	27

Chapitre II

La norme DICOM

II.1 Introduction.....	28
II.2 Vue générale de la norme DICOM.....	29
II.3 Buts de la norme DICOM.....	31
II.4 Historique.....	32
II.4.1 Avant DICOM.....	32
II.4.2 Les standard ASR-NEMA.....	32
II.4.3 Le standard DICOM.....	33
II.5 Comité DICOM.....	34
II.6 Propriétés des fichiers DICOM.....	36
II.6.1 L'orientation Objet.....	36
II.6.2 Les <i>UIDs</i>	37
II.6.3 Les Objets de définition des informations.....	37
II.6.4 Codage des attributs.....	38
II.6.4.1 Codage implicite.....	38
II.6.4.2 Codage explicite.....	38
II.6.5 Les SOP Class.....	39

II.6.6 Les services	40
II.7 Modèle d'information DICOM.....	42
II.8 Structure des fichiers DICOM.....	44
II.9 Particularités d'une image DICOM	45
II.10 Une norme orientée réseau.....	46
II.11 La communication entre les machines.....	48
II.12 Conclusion	50

Chapitre III

Méthodes proposées

III.1 Introduction.....	51
III.2 La méthode sans pertes.....	52
III.2.1 La compression.....	52
III.2.2 La décompression.....	52
III.3 La méthode avec pertes.....	56
III.3.1 La compression.....	57
III.3.2 La décompression.....	62
III.4 Conclusion.....	65

Chapitre IV

Présentation et interprétation des résultats

IV.1 Introduction.....	66
IV.2 Compression sans pertes.....	67
IV.3 Quantification scalaire.....	69
IV.4 Quantification vectorielle.....	72
IV.5 Compression avec pertes.....	75
IV.6 L'interface graphique DICOM QUNTIZIER.....	77
IV.7 Exemple de déroulement d'une opération.....	79
IV.1 La compression.....	79
IV.1 La compression.....	84
IV.8 Conclusion	87

Conclusion générale.....	88
--------------------------	----

Annexe : Dictionnaire DICOM

Bibliographie

Liste des figures

- Figure I.1. Compromis dans le mode de codage sans perte
- Figure I.2. Arbre binaire de Huffman
- Figure I.3. Génération de l'étiquette pour la séquence "acaab"
- Figure I.4. Exemples de Codage par plage RLE
- Figure I.5. Compromis dans le mode de codage avec perte
- Figure I.6. Comparaison des mesures de similarité des images avec différentes distorsions
- Figure I.7. Principe général d'une chaîne de compression des images
- Figure I.8. Transformée de Fourier
- Figure I.9. Transformée de Fourier à court terme
- Figure I.10. Transformée en ondelettes
- Figure I.11. Décompositions successives de la transformée en ondelettes
- Figure I.12. Les quatre sous-bandes de l'image décomposée
- Figure I.13. Banc de filtres pour la décomposition et la reconstruction en ondelettes
- Figure I.14. Principe de la quantification scalaire
- Figure I.15. Principe de la quantification vectorielle
- Figure I.16. Carte de Kohonen
- Figure I.17. Algorithme d'apprentissage de Kohonen
- Figure I.18. Deux exemples de dictionnaires
- Figure I.19. Robustesse aux erreurs de transmission de la carte de Kohonen
-
- Figure II.1. Codage des attributs dans DICOM
- Figure II.2. Modèle d'information DICOM
- Figure II.3. Couches réseau gérées par DICOM
- Figure II.4. Exemple d'une communication entre deux dispositifs
-
- Figure III.1. Schéma général de la compression Sans perte
- Figure III.2. Elimination de la zone sombre
- Figure III.3. Schéma général de la décompression Sans perte
- Figure III.4. Ajout de la zone sombre
- Figure III.5. Schéma général de la compression Avec perte
- Figure III.6 Transformation en ondelettes

Figure III.7 Le Principe de la quantification vectorielle

Figure III.8 Le parcours d'une image quantifiée en spiral

Figure III.9 Schéma général de la décompression Avec perte

Figure III.10 Réorganisation d'une image (vecteur) codée

Figure IV.1. Résultats de la compression sans pertes

Figure IV.2. Dictionnaire utilisé pour la quantification Scalaire

Figure IV.3. Résultats de la quantification scalaire

Figure IV.4. Dictionnaire utilisé pour la quantification vectorielle

Figure IV.5 Résultats de la quantification vectorielle

Figure IV.6. Résultats de la compression avec pertes

Figure IV.7. L'interface graphique DICOM QUNTIZIER

Liste des tableaux

Tableau II.1. Liste des organisations participant au comité DICOM

Tableau II.2. Liste des groupes de travail au sein du comité DICOM

Tableau II.3. Liste des organisations participant au comité DICOM

Tableau III.1. Coefficients des filtres Le Gall 5/3 de décomposition

Tableau III.2. Coefficients des filtres Le Gall 5/3 de reconstruction

Tableau IV.1. Résultats de la compression sans pertes

Tableau IV.2. Résultats de la quantification scalaire

Tableau IV.3. Résultats de la quantification vectorielle

Tableau IV.4. Résultats de la compression avec pertes

Index

A

- ACR : American College of Radiology.
- AE : Application Entity.
- ANSI : American National Standards Institute.

B

- BD : Bit Depth.

C

- CEN : Comité Européen de Normalisation.
- CMJN : Cyan Magenta Jaune Noir.

D

- DCT : Discrete Cosine Transform.
- DICOM : Digital Imaging and COmmunication in Medicine.
- DIMSE : DICOM Message Service Element.
- DPCM : Differential Pulse Code Modulation.
- DWT : Discrete Wavelet Transform.

G

- GLA : Generalized Lloyd Algorithm.
- GUI : Graphical User Interface.

H

- HL7 : Health Level Seven.

I

- IOD : Information Object Definition.

J

- JPEG : Joint Photographic Experts Group.
- JPIP : JPEG2000 Interactive Protocol.

L

- LBG : Linde-Buzo-Gray.

M

- MIPS : Million of Instructions Per Second.
- MOPS : Million of Operations Per Second.
- MSE : Mean Square Error.

N

NEMA : National Electrical Manufacturers Association.

NIU : Network Interface Unit.

O

OSI : Open Systems Interconnected.

P

PACS : Picture Archiving and Communication System.

PSNR : Peak Signal to Noise Ratio.

R

Rc : Rapport de compression.

RLE : Run Length Encoding.

ROI : Region Of Interest.

RVB : Rouge Vert Bleu.

S

SCP : Service Class Provider.

SCU : Service Class User.

SNR : Signal to Noise Ratio.

SOM : Self Organizing Map.

SOP : Service Object Pair.

SQ : Scalar Quantization.

SSIM : Structural SIMilarity.

STFT : Short Time Fourier Transform.

T

Tc : Taux de compression.

TCP/IP : Transmission Control Protocol/Internet Protocol.

TLS : Transport Layer Security.

U

UID : Unique IDentifier.

V

VQ : Vector Quantization.

W

WADO : Web Access to DICOM pertinent Objects.

WG : Work Group.

INTRODUCTION GENERALE

Depuis l'invention du premier microprocesseur en 1971, les technologies numériques ont connu une très importante évolution au fil des années. Elles occupent aujourd'hui une place prépondérante dans notre société.

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. On est bien dans l'ère du numérique.

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 permet aujourd'hui d'avoir des représentations de plus en plus fines des organes, il en résulte des diagnostics de plus en plus pertinents. Cependant, le volume d'information généré est très important encombrant les systèmes de stockage et les réseaux de communication des établissements de la santé. L'augmentation continue des capacités de stockage et des bandes passantes permet d'apporter une solution à ce problème, elle reste cependant non optimale. La compression d'images médicales devient alors un outil inévitable.

Aujourd'hui, plusieurs techniques de compression sont utilisées en imagerie médicale, elles sont en général sans pertes. Elles sont basées sur la réduction de la redondance et permettent de restituer la qualité des images originales, mais les rapports de compression qu'elles apportent sont insuffisants. Dans un souci d'optimisation de l'utilisation des bandes passantes et des espaces de stockage, quelques méthodes avec pertes, exploitant les défauts du système visuel humain, commencent à être utilisées, avec une certaine réserve de la part des médecins ; une mauvaise reconstruction d'une image peut mener à de mauvais diagnostics.

Dans ce travail, nous présentons deux méthodes de compression des images médicales issues de la norme DICOM : la première est quasiment sans pertes (*notée sans pertes dans la suite de ce document*) basée sur l'algorithme RLE, la seconde est avec pertes. Dans cette dernière, nous avons effectué un prétraitement à l'aide de la transformée en ondelettes afin de séparer les basses fréquences des hautes fréquences. Nous avons ensuite fait une quantification avec des dictionnaires construits à l'aide de la carte auto-organisatrice de Kohonen. Notre chaîne de compression se termine par un codage arithmétique.

L'élaboration de ces deux méthodes a été réalisée au sein de l'équipe A3SP de la division ASM du CDTA.

Les chapitres abordés dans ce document sont les suivants :

Le premier est une introduction à la compression en général et à la compression des images médicales en particulier. Nous y aborderons les différentes techniques utilisées et définirons les critères d'évaluation objectifs.

Le second est consacré à la norme DICOM. Nous la définiront, expliquerons ses spécificités, les raisons qui ont mené à son élaboration et ses apports dans les systèmes d'information de la santé.

Dans le troisième, nous expliquerons les algorithmes que nous avons mis au point ainsi que leurs spécificités.

Dans le dernier, nous exposeront les résultats obtenus lors des principaux tests effectués et justifiant nos choix pour les méthodes finales adoptées. Nous donnerons aussi un aperçu de l'interface graphique que nous avons élaboré.

Chapitre I

GENERALITES
SUR LA COMPRESSION
D'IMAGES

I.1 Introduction

Actuellement, l'image numérique est omniprésente. Du plus simple téléphone mobile au plus complexe ordinateur, tous les systèmes électroniques, utilisant des interfaces utilisateur graphiques (*Graphical User Interface : GUI*) [1] ou servant à créer et/ou stocker des images sous forme numérique, manipulent de grandes quantités d'information. Ces opérations deviennent encombrantes avec la présence des contraintes de temps réel, de simplicité, de stockage et de transmission.

Dans le domaine médical, un bon diagnostic ne sera possible qu'avec un suivi rigoureux et efficace des patients imposant de garder les données médicales issues du suivi (informations précieuses à la prise de décision : diagnostic partiel, analyse, scanner...) après chaque consultation. Ceci pose de sérieux problèmes pour les services d'imagerie des hôpitaux. Les études ont prouvé que le service de radiologie d'un grand hôpital peut produire plus de 20 Térabits de données d'image par année [2].

Pour remédier à ces problèmes, la compression des images numériques est alors une solution incontournable.

Une caractéristique très importante de la compression d'images est qu'elle peut être avec pertes. Après tout, une image, est visualisée par des humains donc on peut se permettre de négliger quelques informations imperceptibles par le système visuel humain (HVS).

Éliminer la redondance présente dans l'image et adapter les techniques de compression avec le système visuel humain est l'idée de base derrière toutes les méthodes de compression avec pertes qui existent à l'heure actuelle.

Néanmoins, on trouve dans quelques domaines exigeant à la fois l'efficacité et la qualité, des techniques hybrides c.-à-d. qui utilisent pour la compression d'une image, une méthode sans pertes ou avec une faible dégradation pour quelques régions dites d'intérêt et une autre avec pertes pour le reste. Ceci est très utilisé dans la compression des images médicales avec régions d'intérêt (par exemple : une image IRM cérébrale avec une tumeur).

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 décompressée. Nous présentons par la suite, les méthodes les plus utilisées pour la compression avec et sans pertes.

I.2 La compression

C'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.

Toutes les techniques de compression d'images essaient en général de tirer partie des redondances d'origine statistique, spatiale ou fréquentielle.

On différencie généralement les méthodes de compression selon la distorsion ou la perte d'informations qui est engendrée, comme suit :

- Les méthodes avec pertes « avec distorsion ».
- Les méthodes sans pertes « sans distorsion ».

I.2.1 Les techniques sans distorsion

La compression sans pertes (codage réversible) permet de retrouver, après reconstruction, la valeur exacte du signal comprimé. En fait, la même information est réécrite d'une manière plus concise. Le processus de codage sans pertes crée des "mots-codes" à partir d'un dictionnaire statique ou d'un dictionnaire construit dynamiquement. 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 présentés dans [3]. 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. [4]

Dans ce cas, les compromis liés à ce mode de compression sont selon trois axes illustrés par la figure I.1 : [5]

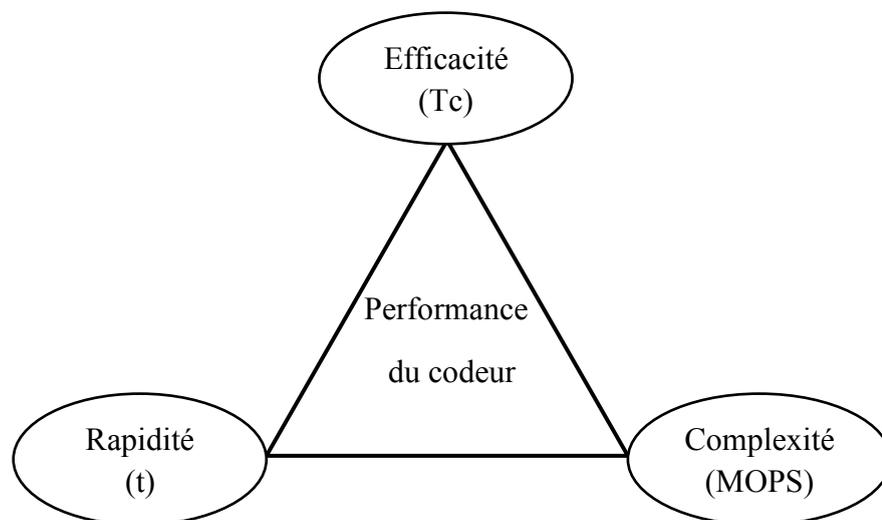


Figure I.1 : Compromis dans le mode de codage sans perte [5]

- 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 (*Million 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.

Dans ce qui suit, nous allons présenter les trois méthodes de compression sans pertes les plus utilisées. Plus de méthodes sont détaillées dans [1].

I.2.1.1 Codage de Huffman

Le codage de Huffman consiste à coder les symboles par une représentation de bits à longueur variable. Les symboles ayant une probabilité d'apparition plus 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 code d'un symbole ne doit pas être le préfixe d'un autre code. Cette propriété est admise afin que la reconnaissance soit possible. Pour représenter le codage de Huffman, on utilise l'arbre binaire.

Soit un message a codé "ABBBBAAC". La fréquence d'apparition ainsi que le code Huffman correspondant est donné dans le tableau suivant et représentés par la figure I.2 :

Symbole	Fréquence d'apparition	Code Huffman
a	3	01
b	4	1
c	1	00

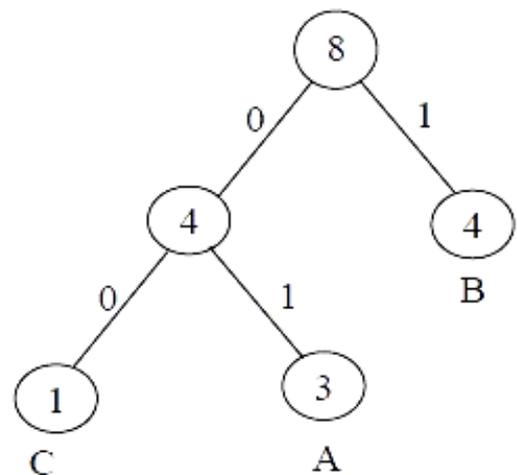


Figure I.2 : Arbre binaire de Huffman [1]

I.2.1.2 Codage arithmétique

Le codage arithmétique 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. Il tend vers la limite inférieure théorique de Shannon. Cependant il est gourmand en ressources et nécessite de connaître à priori l'intégralité du signal avant de pouvoir procéder au codage.

Les différentes étapes de l'algorithme de codage sont :

- L'initialisation : Nous affectons à chaque symbole une plage d'intervalle dont la longueur est égale à sa probabilité d'apparition fournie par le modèle. Les bornes externes de l'intervalle initial sont zéro et un.

- Le traitement du message: Nous initialisons un intervalle de travail en prenant comme bornes 0 et 1. Le premier symbole est représenté par la plage qui lui est affectée à l'étape 1.

Chaque symbole suivant restreint davantage l'intervalle et il est représenté par sa plage relative dans la plage précédente. Ainsi le flot de données est traduit par un nombre contenu dans la dernière plage calculée.

- On rajoute un symbole spécial pour déterminer la fin du message où l'on donne la longueur du flot avec le message codé pour permettre au décodeur de déterminer la fin du message.

Par exemple, supposons qu'on veut coder une partie "acaab" d'une longue séquence avec une probabilité d'apparition indiquée dans le Tableau suivant :

Symbole	Probabilité	Intervalle
a	0.7	$[0, 0.7]$
b	0.1	$[0.7, 0.8]$
c	0.2	$[0.8, 1]$

L'intervalle initial $[0, 1]$ va être divisé en trois sous-intervalles suivant les probabilités des symboles de la séquence. Ce qui donne les sous-intervalles suivants:

$[0, 0.7]$, $[0.7, 0.8]$ et $[0.8, 1.0]$.

Dans cet exemple, le 1^{er} symbole est 'a', l'étiquette appartient donc à l'intervalle $[0, 0.7]$.

Après que le 1^{er} symbole soit codé, les limites inférieures et supérieures de l'intervalle sont respectivement 0 et 0.7 pour le symbole suivant. L'intervalle $[0, 0.7]$ va être divisé en trois sous-intervalles : $[0, 0.49]$, $[0.49, 0.56]$ et $[0.56, 0.7]$ correspondant respectivement aux symboles "a", "b" et "c".

Le 2^{ème} symbole est "c", de probabilité 0.2. Ainsi, le nouveau sous-intervalle sera $[0.56, 0.7]$.

Ce dernier va être divisé en trois sous-intervalles : $[0.56, 0.658]$, $[0.658, 0.672]$ et $[0.672, 0.7]$.

Le 3^{ème} symbole est "a". Le nouveau sous-intervalle sera en conséquence $[0.56, 0.658]$. Le sous-intervalle $[0.56, 0.658]$ va être partagé à son tour en trois sous-intervalles : $[0.56, 0.6286]$, $[0.6286, 0.6286]$ et $[0.6286, 0.658]$.

Le 4^{ème} symbole est "a", donc le nouvel intervalle est $[0.56, 0.6286]$, on le divise en 3 sous-intervalles : $[0.56, 0.60802]$, $[0.60802, 0.61488]$, $[0.61488, 0.6286]$.

Le 5^{ème} et dernier symbole est "b", le dernier intervalle est donc $[0.60802, 0.61488]$.

Un nombre contenu dans l'intervalle final comme 0.60972 code sans ambiguïté le message "acaab". La figure I.3 est une représentation de ce processus.

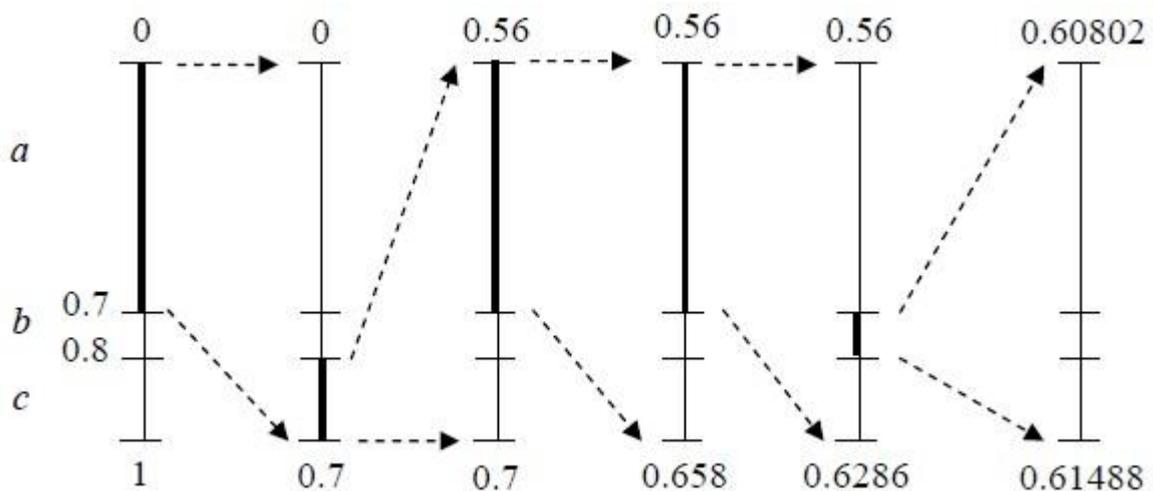


Figure I.3 : Génération de l'étiquette pour la séquence "acaab" [1]

La table contenant les probabilités d'occurrence des différents symboles doit être transmise. Pour le décodage, on prend le symbole correspondant à l'intervalle auquel appartient notre code, on lui soustrait la limite basse, et on divise par la longueur de l'intervalle pour avoir le code suivant.

Dans notre exemple, on reçoit le nombre 0.60972 qui appartient à l'intervalle [0, 0.7], le premier symbole est donc "a", on soustrait ensuite la limite basse (0) et on divise par la longueur de l'intervalle (0.7), on obtient : 0.87103 qui appartient à [0.8, 1], le second symbole est donc "c", on soustrait ensuite la limite basse (0.8) et on divise par la longueur de l'intervalle (0.2), on obtient : 0.35514 qui appartient à l'intervalle [0, 0.7], le troisième symbole est donc "a". On continue ainsi jusqu'à obtenir notre séquence complète.

La longueur d'une séquence à coder avec le codeur arithmétique est limitée par la résolution de la machine qui assure le traitement.

I.2.1.3 Le Codage par plage (*Run length Encoding : RLE*)

Le codage par plage (RLE) ne relève pas d'une théorie mathématique très complexe. Il s'agit 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 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. Un exemple de traitement RLE est donné à la figure I.4.

10 10 10 10 10	est codé 05 10
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 I.4 : 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.

Si n octets successifs sont dans un même état, il est aisé de transmettre l'octet répété et le nombre de répétitions. On pourra ainsi coder sur 2 octets les n octets composant le signal initial.

I.2.2 Les techniques avec distorsions

Les approches irréversibles sont des méthodes qui tirent parti d'une corrélation (ou redondance) présente dans l'image, en même temps elles exploitent le fait que l'œil n'est pas sensible à certaines dégradations. Par exemple, le nombre de niveaux de gris que l'œil peut distinguer est largement inférieur à 256, il n'est même pas de 64. La compression, probablement la plus simple, des images monochromes codées sur 8 bits par pixel serait donc tout simplement d'ignorer les deux derniers bits de chaque pixel.

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 I.5) : il est utilisé pour caractériser le signal à la sortie du décodeur [5]. 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).

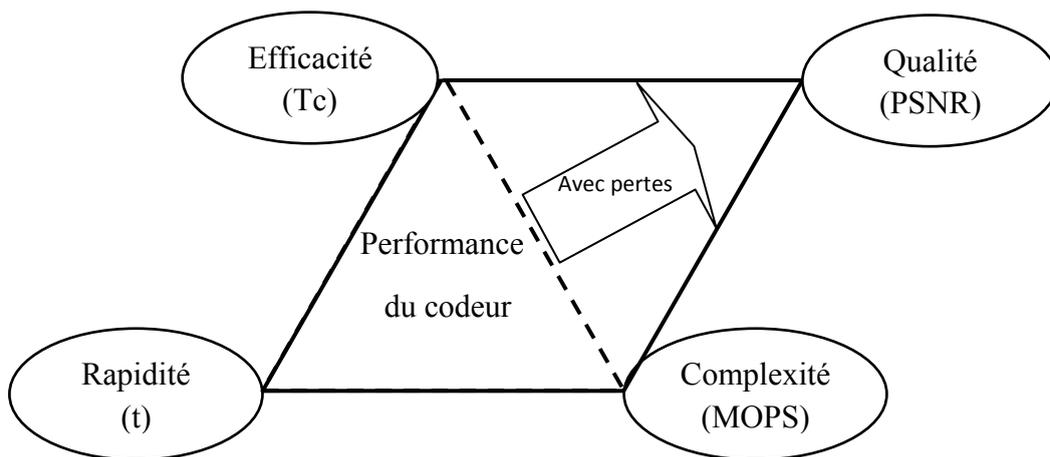


Figure I.5 : Compromis dans le mode de codage avec perte [5]

I.2.3 Critères d'évaluation [3]

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.

- **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 :

$$Rc = \frac{\text{Espace mémoire occupé par les données non compressées}}{\text{Espace mémoire occupé par les données compressées}} \quad (I.1)$$

On définit alors le taux de compression par :

$$Tc = 1 - \frac{1}{Rc} \quad (I.2)$$

- **L'évaluation de la distorsion**

On utilise deux méthodes, une subjective l'autre objective :

1- 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.

2- Pour la méthode objective, on utilise souvent le calcul de l'erreur quadratique moyenne (*Mean Square Error : MSE*), elle est définie par la moyenne des écarts au carré entre les pixels de l'image originale (X_i) et l'image reconstruite (x_i) comme suit [3]:

$$MSE = \frac{1}{R} \sum_{i=1}^R (X_i - x_i)^2 \quad (I.3)$$

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

On peut alors définir le rapport crête signal sur bruit (*Peak Signal to Noise Ratio : PSNR*) qui mesure la fidélité de la compression puisqu'il relie le MSE à la dynamique de l'image [3]:

$$PSNR = 10 \times \log_{10} \left(\frac{D^2}{MSE} \right) dB \quad (I.4)$$

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

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

- **Similarité structurelle [7]**

Basé sur la distorsion structurelle, le SSIM (Structural SIMilarity) est utilisé 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. Ceci donne de très bons résultats (qualité

directement proportionnelle à la valeur de SSIM, figure I.6).

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 [7]:

$$SSIM(x, y) = \frac{(2 \mu_x \mu_y + C_1)(2 cov_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (I.5)$$

C_1 et C_2 sont des constantes :

$$C_1 = (K_1 L)^2 \text{ et } C_2 = (K_2 L)^2$$

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 instable. Ce problème a été résolu en ajoutant ces deux petites constantes, calculées en posant $K_1 = 0.01$ et $K_2 = 0.03$.

Dans cette figure, on constate que le SSIM est plus efficace pour la mesure de la qualité perceptuelle des images.

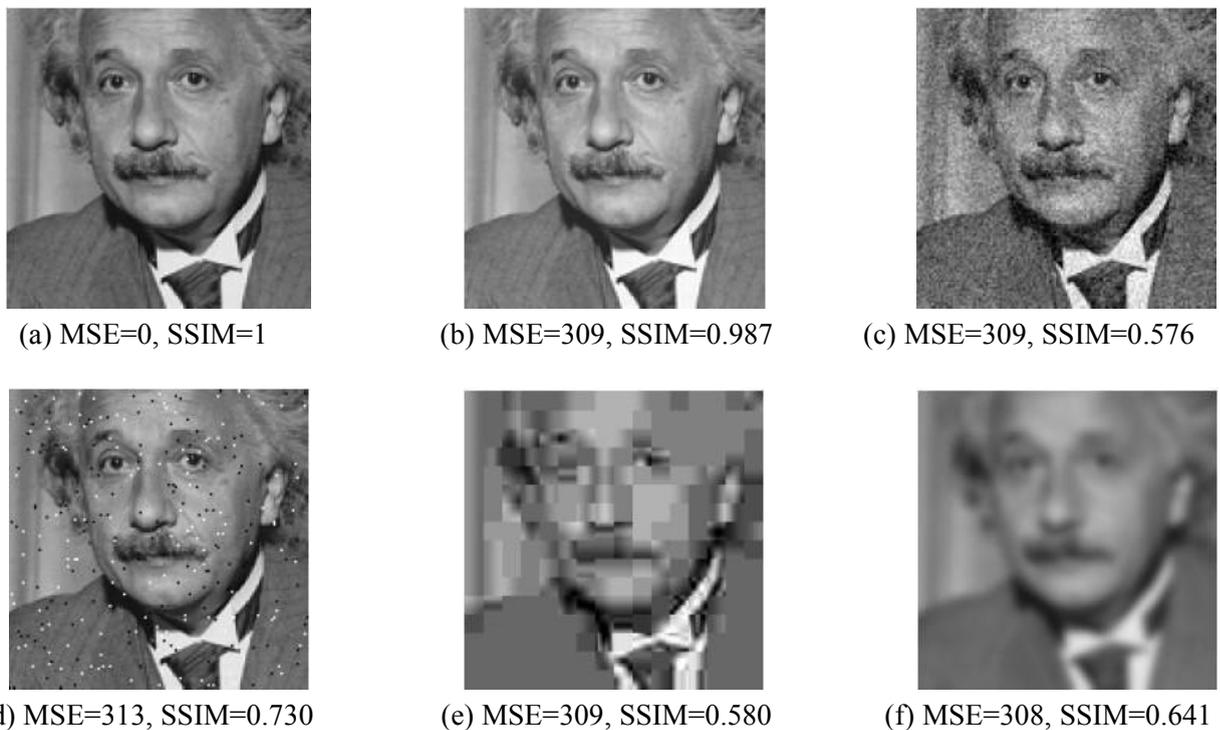


Figure I.6 : Comparaison des mesures de similarité des images avec différentes distorsions [8]

- (a) image référence (supposée avoir une qualité parfaite) ; (b) luminance moyenne décalée ;
(c) bruit Gaussien ; (d) bruit impulsif ; (e) compression JPEG ; (f) image floue

Le SSIM satisfait les conditions suivantes :

- La symétrie : $SSIM(x, y) = SSIM(y, x)$
- $SSIM(x, y) \leq 1$.
- Unicité du maximum : $SSIM(x, y) = 1$ si et seulement si $x = y$.

Le SSIM est calculé en utilisant une fenêtre courante qui parcourt l'ensemble de l'image bloc-par-bloc. Cette opération nous donne la carte SSIM (une matrice de dimension inférieure) qui représente la carte qualité de l'image distordue, en supposant que l'image originale a une qualité parfaite. Finalement, le SSIM moyen, moyenne de la carte SSIM, est utilisé pour évaluer la qualité globale de l'image.

Ces mesures sont globales, en aucun cas elles ne permettent de localiser une erreur dans l'image ou de donner une information sur la qualité perçue par les observateurs humains.

Elles donnent une appréciation de la qualité d'ensemble de l'image et sont utilisées essentiellement pour comparer les méthodes ou les taux de compression. Elles entrent également dans les processus d'optimisation des algorithmes de compression pour lesquels on cherche à minimiser les dégradations totales. Par ailleurs, contrairement aux images naturelles ou à la vidéo, très peu d'études ont été réalisées pour proposer d'autres critères objectifs d'évaluation de la qualité spécifiques à l'imagerie médicale [6].

I.3 Compression des images médicales

De plus en plus, les médecins ont besoin de visualiser plusieurs fois les clichés et de les communiquer à des confrères pour partager leurs avis et leurs diagnostics. Ainsi, le stockage, la visualisation et la communication sont les trois motivations qui ont amené le 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 type de fichiers.

Les techniques d'acquisition s'améliorent et offrent une résolution toujours plus fine, augmentant d'autant plus l'espace nécessaire pour l'archivage ou le temps pour la transmission des images. Une compression efficace de ces images est alors nécessaire pour réduire le coût et le temps.

I.4 Principe général d'une méthode de compression

Le schéma général souvent utilisé pour décrire le fonctionnement des algorithmes de compression est celui présenté dans la figure I.7.

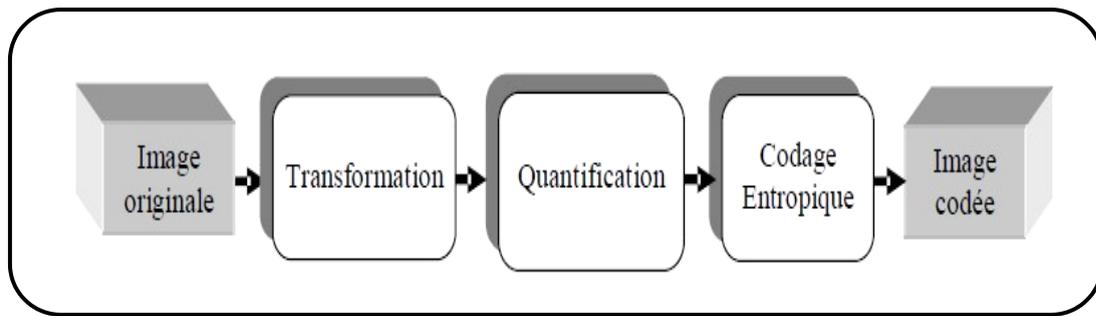


Figure I.7 : Principe général d'une chaîne de compression des images

La chaîne de retour, qui est la reconstruction (ou la décompression), est l'application inverse de ces blocs. Chacun d'eux est expliqué par la suite.

I.4.1 La transformation

Dans un premier temps, afin de mieux compacter l'information, la source est transformée en groupes de coefficients réorganisés de manière à pouvoir éliminer les redondances (séparer les basses fréquences des hautes fréquences).

En plus, effectuer une quantification directement sur l'image n'est pas très efficace car on n'exploite pas la propriété de corrélation entre les pixels voisins. La transformée linéaire T ainsi utilisée vise à décorréler les composantes des vecteurs (valeurs des pixels de l'image). Pour cela, elle doit être impérativement orthogonale, et pour plus de simplicité orthonormée (donc $T^{-1} = T^t$) [9].

Cette indépendance créée grâce à l'orthogonalité de la transformation implique que le système vu depuis le codeur sera considéré comme une source sans mémoire (d'ordre 0), qui est le modèle le plus simple à coder.

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 (*Discrete Cosine Transform : DCT*), la Transformée en Ondelette Discrète (*Discrete Wavelet Transform : DWT*) et la décomposition fractale. Quelque soit la transformation, la variance (l'énergie) des composantes transformées est concentrée sur les basses fréquences.

Dans notre travail, nous avons utilisé la DWT qui est devenue un outil très efficace dans le domaine de traitement numérique du signal et qui a été utilisée pour la compression des images fixes et des vidéos. La DWT est aussi utilisée dans d'autres applications du traitement d'image telle que la détection de contour, la détection d'objet, l'élimination du bruit présent dans l'image (*denoising*) et la fusion des images [10].

Dans ce qui suit, nous allons brièvement présenter la DWT pour mieux comprendre notre algorithme et les choix de ses classificateurs et justifier le choix de la DWT.

La transformée en ondelettes [10]

La transformée en ondelettes est un outil mathématique capable de fournir une représentation multi-résolution d'un signal en le projetant sur des bases de fonctions, donnant ainsi des approximations de moins en moins fines du signal d'entrée.

Peut-être la plus célèbre transformation dans le domaine du traitement de signal, la transformée de Fourier est un autre outil mathématique qui nous permet de passer du domaine temporel au domaine fréquentiel en projetant le signal sur des sinusoides de différentes fréquences (figure I.8).



Figure I.8 : Transformée de Fourier [10]

Néanmoins, en faisant cela, toute information sur le temps sera perdue et il sera impossible de dire quand un événement particulier a eu lieu. Cet inconvénient n'est pas très important quand le signal est stationnaire. Par contre, la transformée de Fourier sera incapable de détecter les nombreuses caractéristiques non-stationnaires, comme le début et la fin du signal, qui sont souvent les parties les plus importantes du signal.

Dans un effort pour corriger cette insuffisance, la transformée de Fourier a été adaptée pour analyser seulement une petite section du signal à un temps donné. Cette adaptation connue sous le nom de la transformée de Fourier à court terme (*Short-Time Fourier Transform* : *STFT*) fait apparaître l'ordonnancement de portions du signal (figure I.9).



Figure I.9 : Transformée de Fourier à court terme [10]

La STFT représente une sorte de compromis entre le temps et la fréquence, elle donne des informations sur les deux mais avec une précision limitée qui est déterminée à partir de la

largeur de la fenêtre. L'inconvénient de cette méthode est que : une fois la largeur de la fenêtre temps choisie elle l'est pour toutes les fréquences alors que beaucoup de signaux nécessitent une approche plus flexible où on peut faire varier la largeur de la fenêtre pour déterminer avec plus de précision : le temps et/ou la fréquence.

La transformée en ondelette est logiquement la prochaine étape dans cette suite : une technique de fenêtrage avec largeur de fenêtre variable, des intervalles temporels long pour plus de précision sur les basses fréquences et courts pour les hautes fréquences (figure I.10).



Figure I.10 : Transformée en ondelettes [10]

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 I.11).

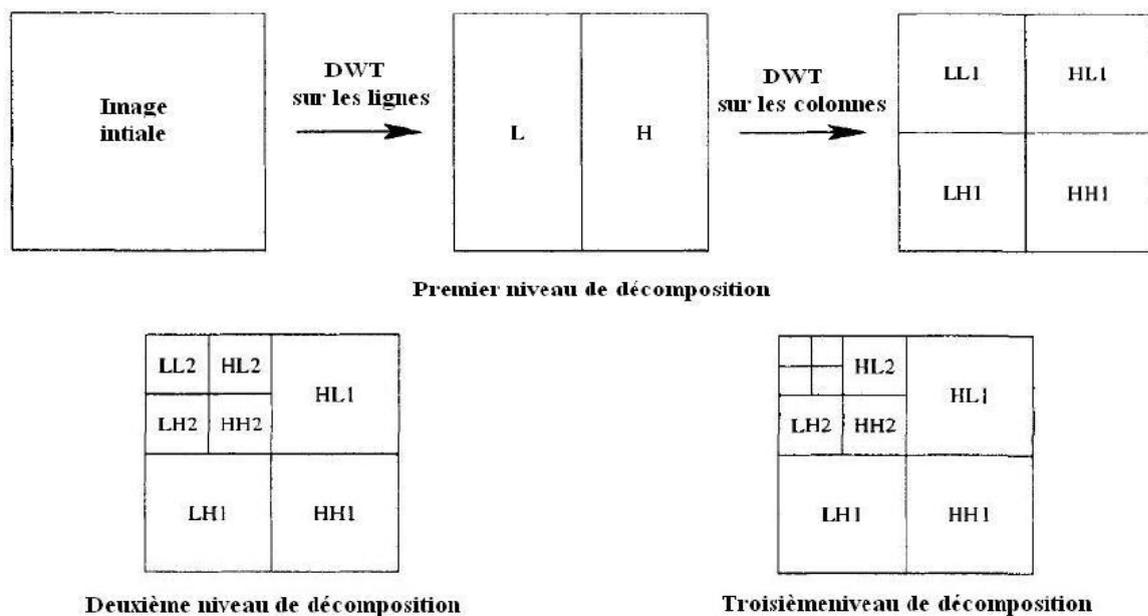


Figure I.11 : Décompositions successives de la transformée en ondelettes [10]

La bande LL est une approximation de l'image initiale et peut être considérée comme sa version sous-échantillonnée 2 : 1 (à la fois horizontalement et verticalement). Les trois autres bandes LH, HL et HH (horizontale, verticale et diagonale) contiennent les informations hautes fréquences (figure I.12).

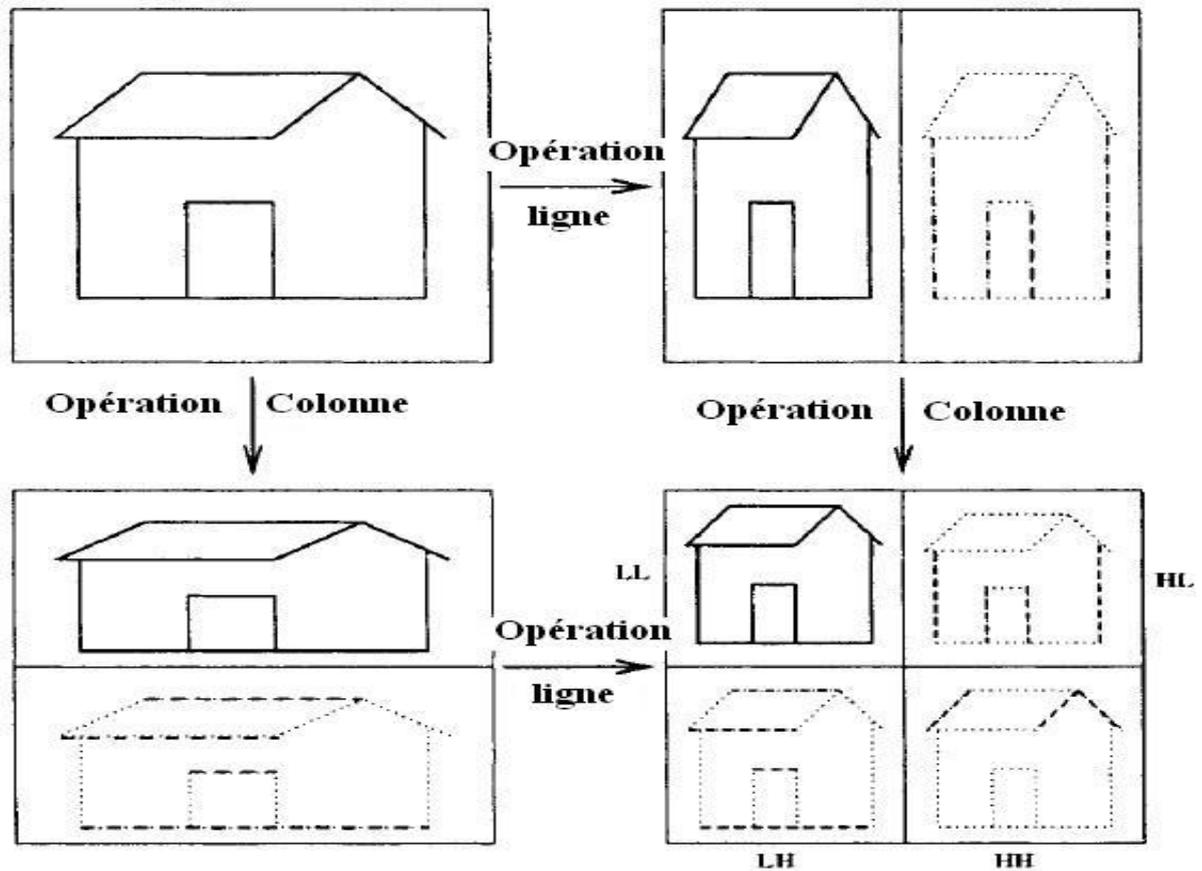


Figure I.12 : Les quatre sous-bandes de l'image décomposée [10]

Le passage entre deux approximations successives se fait par un filtrage passe bas (L) et un filtrage passe haut (H), suivi d'une décimation d'un facteur de 2 (notée $\downarrow 2$).

La reconstruction est l'étape inverse de décomposition avec une interpolation par des zéros. On insère un zéro tous les deux échantillons successifs (notée $\uparrow 2$), puis on applique un filtrage passe bas de reconstruction (L^{-1}) et un filtrage passe haut de reconstruction (H^{-1}). Pour appliquer cette analyse sur une image on applique les traitements si dessus d'abord sur les lignes puis sur les colonnes ou inversement comme on va le montrer sur la figure I.13.

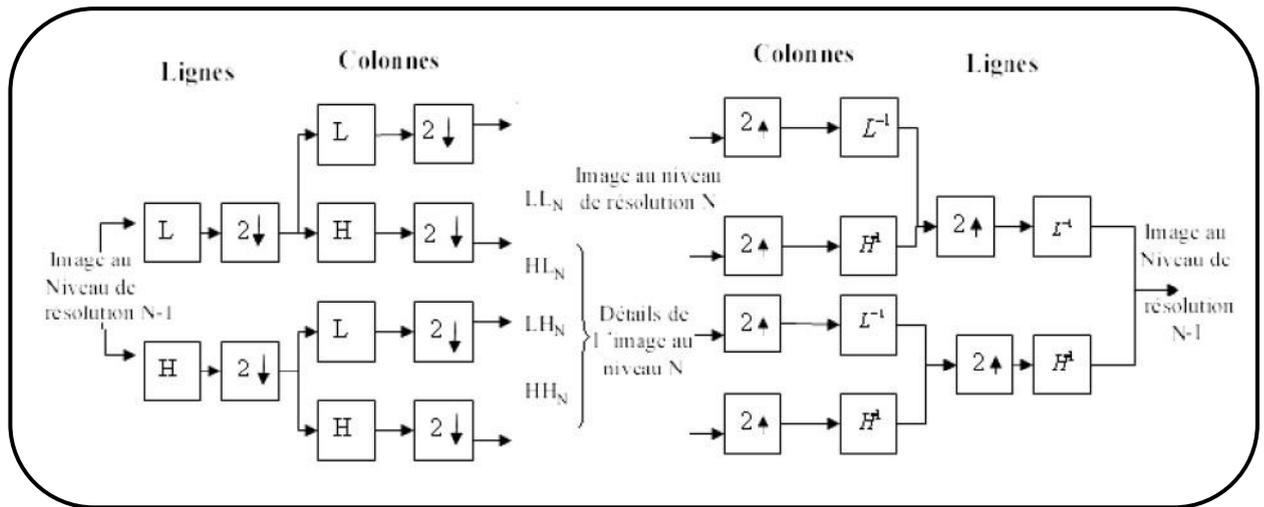


Figure I.13 : Banc de filtres pour la décomposition et la reconstruction en ondelettes [19]

I.4.2 La quantification [11]

Par définition, l'opération de quantification permet une transcription des données depuis un espace de taille "infini" constitué par exemple de l'ensemble des nombres flottants, ou de taille très importante, vers un espace contenant un nombre limité de coefficients. La quantification fait partie de plusieurs méthodes de compression d'image. L'objectif est de réduire la taille des coefficients de façon que cette réduction n'apporte pas de dégradations visuelles à l'image.

Deux principales familles de quantification existent; elles travaillent soit coefficient par coefficient, soit par groupe de coefficients. Dans le premier cas on parle de quantification scalaire, et dans le second, où on alloue conjointement un vecteur représentant à cet ensemble de coefficients, on parle alors de quantification vectorielle. Le choix de ces représentants vectoriels ou scalaires et leur répartition a donné lieu à de nombreux algorithmes de quantification.

I.4.2.1 Quantification Scalaire

La quantification scalaire (*Scalar Quantization* : *SQ*) est réalisée indépendamment pour chaque élément. D'une manière générale, on peut la définir comme étant l'association de chaque valeur réelle x , à une autre valeur q qui appartient à un ensemble fini de valeurs comme indiqué dans la figure I.14. La valeur q peut être exprimée en fonction de la troncature utilisée soit par : l'arrondi supérieur, l'arrondi inférieur, ou l'arrondi le plus proche.

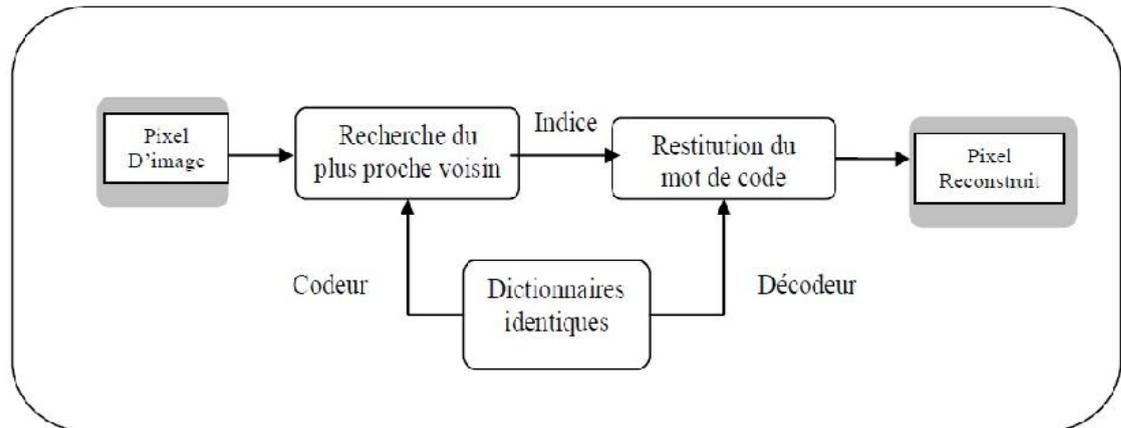


Figure I.14 : Principe de la quantification scalaire

I.4.2.2 Quantification Vectorielle

La quantification vectorielle (*Vector Quantization : VQ*) a été développée par Gersho et Gray et elle fait aujourd'hui l'objet de nombreuses publications dans le domaine de la compression numérique. Le principe de la quantification vectorielle est issu du travail 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 $X_i = (x_i, i=1 \dots k)$ un vecteur $Y_j = (y_j, j=1 \dots k) = Q(X_i)$; ce vecteur Y_j qui est affecté suivant une distance minimale est choisi parmi un dictionnaire (*Codebook*) de taille finie. La quantification vectorielle produit de meilleurs résultats que la quantification scalaire en terme de taux de compression, néanmoins elle nécessite un codage complexe et de grandes capacités de mémoire en plus d'être très gourmande en temps de calcul.

La figure suivante montre le schéma général d'un quantificateur vectoriel, pour un bloc de dimension prédéfinie.

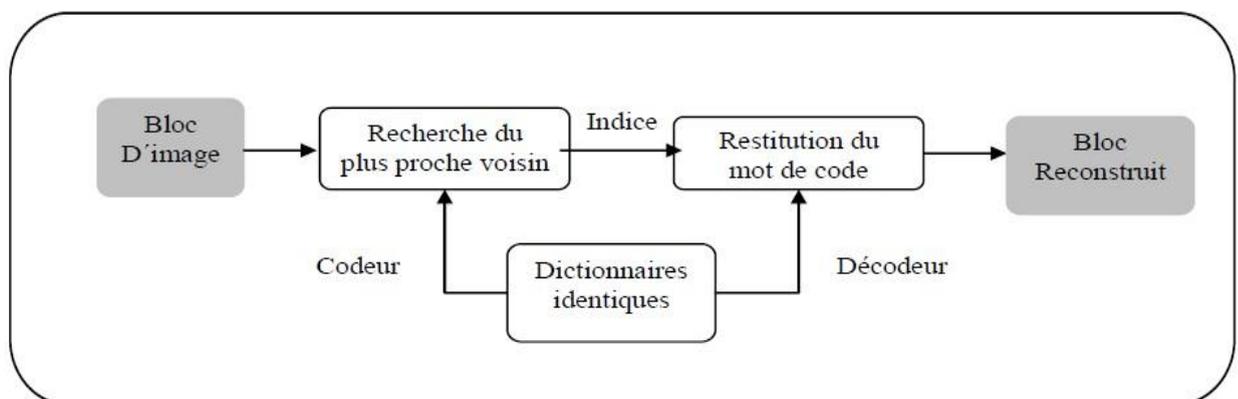


Figure I.15 : Principe de la quantification vectorielle

I.4.2.3 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. Les mots codes de l'espace de sortie doivent avoir la même distribution statistique que les échantillons de l'espace d'entrée.

I.4.2.3.1 Construction du dictionnaire [9]

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. Les algorithmes de création du dictionnaire travaillent à partir d'un dictionnaire initial qu'ils modifient à chaque itération afin de minimiser la distorsion moyenne.

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

- 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 décrivons d'abord brièvement un ensemble d'algorithmes. Nous détaillerons ensuite l'algorithme que nous avons utilisé dans notre application : La carte auto-organisatrice (*Self Organizing Map* : *SOM*) de Kohonen.

I.4.2.3.1.1 Les algorithmes non-neuronaux

I.4.2.3.1.1.1 Algorithme de Lloyd Généralisé (*Generalized Lloyd Algorithm* : *GLA*)

C'est l'algorithme de référence 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 et 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 (représentant moyen).

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.

On continue ainsi pour un certain nombre d'itérations prédéfini ou jusqu'à ce que la différence entre deux dictionnaires successifs soit inférieure à un certain seuil, en général 0.01 ou 0.001.

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.

I.4.2.3.1.1.2 Algorithme de Linde-Buzo-Gray (LBG) [3]

Cet algorithme 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 ensuite 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 ensuite 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.

I.4.2.3.1.1.3 Agrégation constructive progressive [9]

Cet algorithme 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.

I.4.2.3.1.1.4 Quantification Vectorielle adaptative par ré-apprentissage [9]

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 et Yano, seuls deux vecteurs codes sont remplacés à chaque changement.

I.4.2.3.1.2 Les algorithmes neuronaux [9]

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.

En général, on utilise des réseaux de la seconde catégorie pour la construction des dictionnaires.

I.4.2.3.1.2.1 Compétition simple [9]

On initialise d'abord un dictionnaire. Ensuite on tire un vecteur de la base d'apprentissage et on recherche le mot code correspondant. On le met ensuite 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.

I.4.2.3.1.2.2 Compétition sensible à la fréquence [9]

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.

I.4.2.3.1.2.3 Carte auto-organisatrice (Carte de Kohonen) [3]

Mis au point par Teuvo Kohonen, 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.

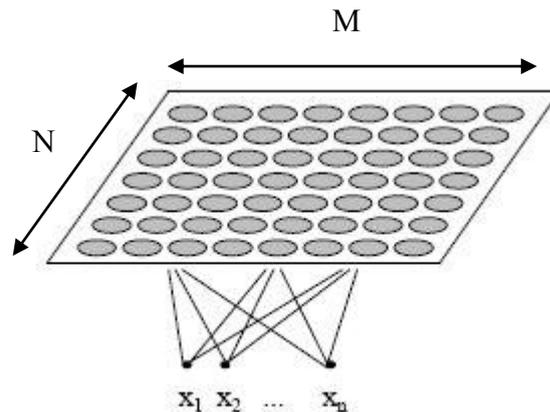


Figure I.16 : Carte de Kohonen [3]

I.4.2.3.1.2.3.1 Algorithme d'apprentissage [3]

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 I.6) entre les neurones et les vecteurs d'apprentissage. La distance euclidienne est définie par la relation suivante [3]:

$$D = \sqrt{\sum_{i=1}^n (W_i - X_i)^2} \quad (I.6)$$

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 (I.7).
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 [3]:

$$W_i(t+1) = W_i(t) + \beta_{i,k}(t) (X_i - W_i(t)) \quad (I.7)$$

$\beta_{i,k}(t)$ 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 [3]:

$$\beta_{i,k}(t) = \beta_0(t) \exp\left(-\frac{|r_i - r_k|}{\sigma^2(t)}\right) \quad (I.8)$$

Où $\beta_0(t)$ et $\sigma(t)$ sont des fonctions décroissantes appropriées, r_i représente les coordonnées du i -ème neurone et r_k représente les coordonnées du k -ième 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.

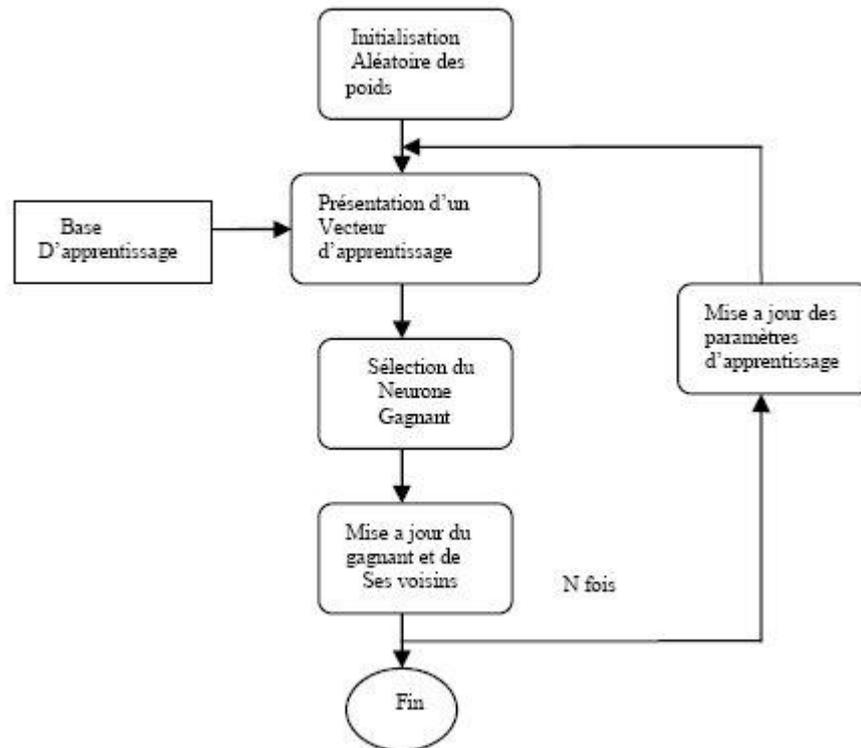


Figure I.17 : Algorithme d'apprentissage de Kohonen [3]

I.4.2.3.1.2.3.2 Propriétés de la carte auto-organisatrice [3]

- 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.

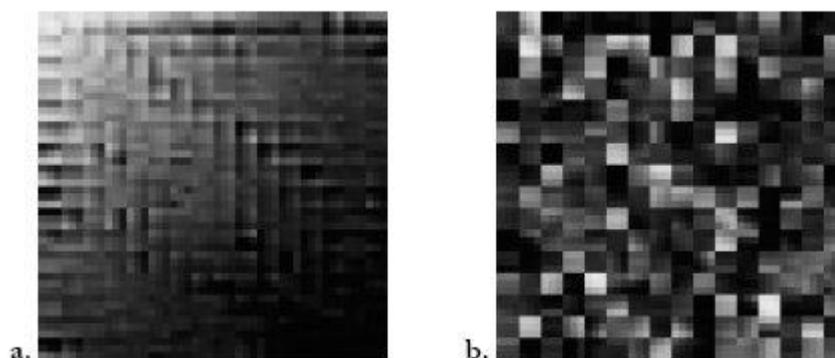


Figure I.18 : Deux exemples de dictionnaires [3]

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. Si ce n'est pas le cas, l'algorithme risque d'osciller et les résultats obtenus risquent d'être altérés.

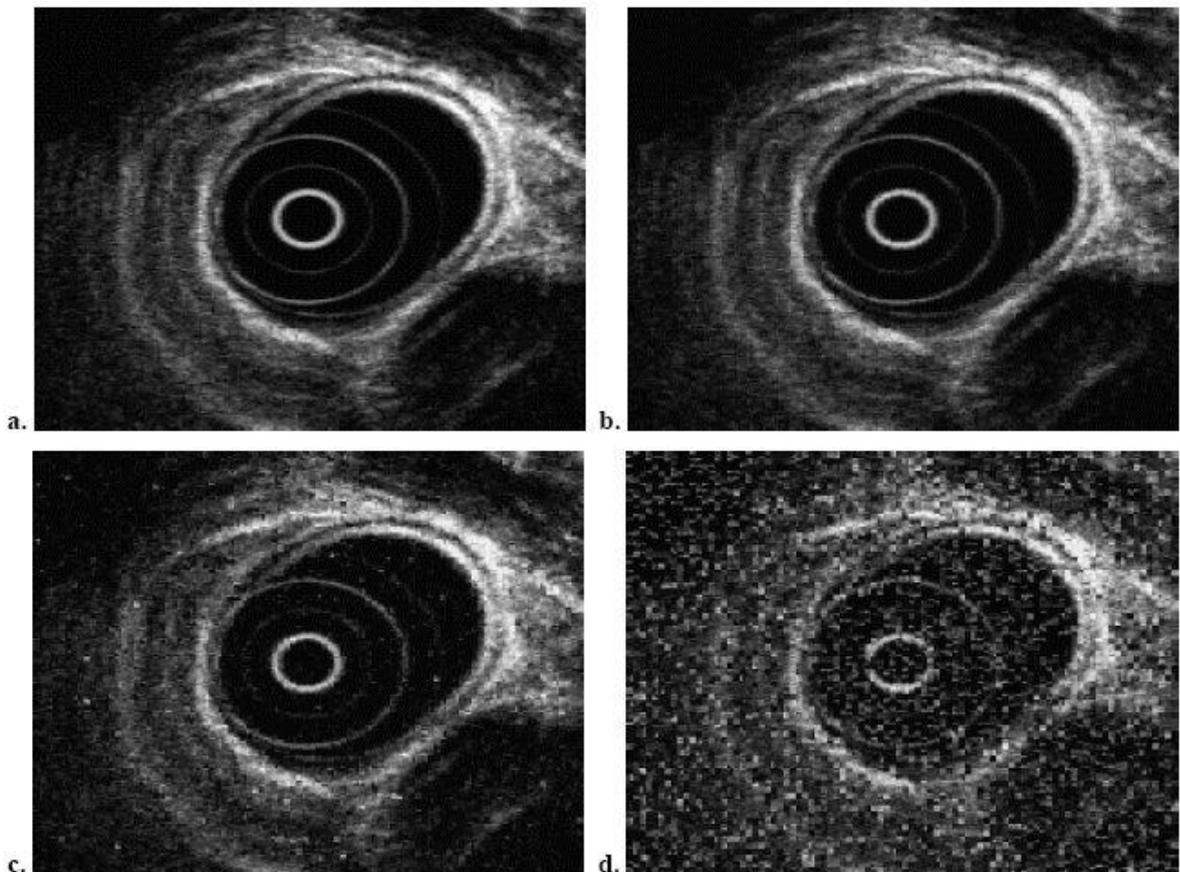


Figure I.19 : Robustesse aux erreurs de transmission de la carte de Kohonen [3]

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.

I.4.2.3.1.2.3.3 Conclusion sur la carte auto-organisatrice

Les réseaux de neurones présentent une bonne alternative aux algorithmes basés sur des modèles mathématiques plus complexes. La qualité des quantificateurs dépendant principalement de leurs dictionnaires, le choix d'une bonne technique pour sa construction est primordial.

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 à travers des canaux bruités.

L'inconvénient principal de cette méthode est qu'elle prend un temps considérable pour le calcul.

I.4.3 Le codage

C'est la dernière étape, qui consiste à attribuer des codes binaires aux coefficients quantifiés selon leurs fréquences d'apparition (propriétés statistiques) en choisissant une loi de compression en fonction de la densité de probabilité des messages à coder.

L'étape de codage entropique est généralement effectuée sur des symboles qui ont été rendus décorrélés grâce à une transformation orthogonale.

Les codeurs les plus utilisés sont le codage Huffman et le codage arithmétique (notamment dans la norme JPEG 2000).

Généralement, l'étape de codage est précédée par une étape de prétraitement (qui pourrait être une façon de lire l'image quantifiée et/ou un codage).

Par exemple, pour la norme JPEG, le bloc avant codage est constitué d'une lecture Zigzag de l'image quantifiée après la DCT (en raison de sa forme particulière) et aussi d'un codage RLE basé sur la valeur 0, qui permet d'augmenter encore plus le taux de compression.

I.5 Conclusion

Les images présentent plusieurs types de redondances exploitables lors de la compression. En plus, le système visuel humain peut apercevoir la forme générale et plusieurs détails d'une image, mais il lui est impossible d'évaluer précisément la couleur de chaque pixel. Donc, on peut procéder à une compression avec un très bon taux de compression sans que la dégradation soit perceptible par l'œil humain, et ceci en éliminant toute information jugée inutile.

La compression présente plusieurs avantages [10]. Le principal est la réduction de l'espace mémoire requis pour le stockage de l'information et le temps lors sa transmission pour une bande passante donnée, et par conséquent le coût de communication pour la transmission des données de grand volume. C'est grâce à cela qu'on peut recevoir des appels (et même des visioconférences) de l'autre moitié du globe à travers le même ancien canal de communication, avec une meilleure qualité et à un prix réduit.

En contre partie, elle présente des inconvénients [10] qui dépendent du domaine de l'application. Par exemple, pour les grandes bases de données où le changement n'est pas très fréquent, on est obligé de décompresser toute la base de données pour n'atteindre à la fin, que la donnée désirée, et après modification, la base de donnée est à nouveau entièrement compressée. La compression réduit en générale la fiabilité des données codées. Dans certains cas, une erreur sur seulement un bit engendrera une mauvaise interprétation des bits subséquents, produisant une donnée incorrecte, ce qui est intolérable pour certains applications (la télémédecine) qui utilisent un canal de transmission bruité.

Chapitre II

LA NORME DICOM

II.1 Introduction

Le domaine de l'imagerie médicale a beaucoup évolué depuis ses débuts au courant des années 1970 [12]. On a vu au cours des années l'émergence de plusieurs modalités (rayons X, ultrasons, résonnance magnétique...) utilisées dans divers services de la médecine.

Cette évolution a inévitablement attiré les manufacturiers d'équipements électriques et électroniques à s'intéresser aux équipements médicaux car ils voyaient là un marché prometteur pour l'avenir. Chaque constructeur offrait donc une large gamme d'équipements. Un problème est malheureusement survenu avec la vulgarisation des équipements : la communication entre équipements de différentes marques était impossible. Ceci entraînait donc une complexité des systèmes d'information des établissements médicaux. C'est là que l'urgence d'une normalisation des équipements et des données médicales est apparue.

Différents comités au sein des organisations de normalisation ont donc vu le jour dans le but de simplifier les systèmes d'information de la santé. Parmi ces comités, on citera [13]:

- Le Comité Technique 251 (appelé *Health Informatics*) au sein du CEN (Comité Européen de Normalisation) en 1991.
- Le Comité Technique 215 (appelé également *Health Informatics*) au sein de l'ISO en 1998.
- Le Comité ACR-NEMA en 1983 devenu DICOM en 1991.
- Le Comité HL7 (*Health Level seven*) au sein de l'ANSI (*American National Standards Institute*) en 1987.

Parmi les différentes normes créées par ces différents organismes, la plus aboutie et la plus prépondérante est la norme DICOM.

II.2 Vue générale de la norme DICOM [12]

DICOM, l'abréviation de « Digital Imaging and COmmunication in Medicine », 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.

Actuellement, la norme DICOM contient 16 parties :

- **Introduction et vision globale (*Introduction and overview*)** : Donne les grandes lignes du standard en expliquant les raisons qui ont mené à sa création et les buts qu'il a permis d'atteindre.
- **Conformité (*Conformance*)** : Définit les principes d'une implémentation conforme du standard. C'est suivant cette partie que les déclarations de conformité sont rédigées.
- **Définition des objets d'informations (*Information Object Definitions : IODs*)** : Spécifie l'ensemble des *IODs* permettant de donner une représentation des éléments du monde réel et applicables pour la communication numérique d'informations médicales.
- **Définition des services (*Service Class Specifications*)** : Spécifie l'ensemble des services permettant de donner une représentation des activités du monde réel et applicables pour la communication numérique d'informations médicales.
- **Encodage (*Data Structured and Encoding*)** : Facilite l'échange des données en définissant une syntaxe de codage précise pour la représentation de l'information.
- **Dictionnaire des données (*Data Dictionary*)** : Contient tous les codes des balises définissant les éléments DICOM ainsi que les identifiants uniques (*Unique Identifiers : UIDs*)
- **Echange des messages (*Message Exchange*)** : Définit la structure des messages DICOM (*DICOM Message Service Element : DIMSE*) échangés entre les services communicants DICOM (point-à-point ou à travers un réseau). Un DIMSE définit un élément de communication incluant un service et un protocole, il est utilisé par les entités DICOM (*Application Entity : AE*) pour le transfert des informations

médicales. Le protocole DIMSE définit les règles nécessaires pour la construction de messages conformes.

- **Support réseau pour l'échange des messages (*Network Communication Support for Message Exchange*)** : Le modèle de communication DICOM à travers un réseau est basé sur le modèle OSI (*Open Systems Interconnected*, paragraphe II.10) de l'ISO. Cette partie définit les règles de communication au niveau applicatif entre les entités DICOM à travers un réseau. Les aspects de la communication à de plus bas niveaux mentionnés et explicités dans cette partie ne sont pas propres au standard. DICOM ne définit que la communication au niveau applicatif (paragraphe II.10).
- **Support de stockage et format des fichiers pour l'échange de donnée (*Media Storage and File Format for Data Interchange*)** : Spécifie un modèle général pour le stockage des informations médicales sur une large gamme de supports amovibles pour permettre l'échange des données à travers ces derniers.
- **Profils des applications spécifiques aux supports de stockage (*Media Storage Application Profiles*)** : Spécifie un sous-ensemble d'applications conformes DICOM utilisables pour le stockage sur les supports amovibles.
- **Formats de médias et supports physiques pour l'échange de données (*Media Formats and Physical Media for Data Interchange*)** : Spécifie une structure décrivant les relations entre le modèle de la partie 10 et des supports physiques spécifiques et des formats spécifiques. Elle définit aussi les caractéristiques des supports physiques et des formats associés.
- **Visualisation en niveaux de gris (*Grayscale Standard Display Function*)** : Définit une fonction normalisée pour l'affichage des images en niveaux de gris.
- **Profils de Sécurité (*Security and System Management Profiles*)** : Spécifie la manière dont doit être implémenté et géré un protocole de sécurité, déjà existant (*Transport Layer Security (TLS)* par exemple) pour être conforme à la norme. DICOM ne définit pas de procédés de sécurisation spécifiques à la norme.
- **Ressources terminologiques (*Content Mapping Resource*)** : Définit l'ensemble des abréviations utilisées tout au long du document.
- **Informations explicatives (*Explanatory Information*)** : Permet de mieux comprendre la signification des termes utilisés dans le document.

- **Accès web (*Web Access to DICOM Persistent Objects (WADO)*)** : Définit un service basé sur le web facilitant la distribution et la consultation d'informations médicales en utilisant les protocoles web (à partir d'une page HTML par exemple).

II.3 Buts de la norme DICOM [12]

La norme DICOM facilite l'interopérabilité entre différents dispositifs revendiquant une conformité en :

- Définissant la sémantique des commandes de communications et des données associées, de sorte à fournir un certain nombre de règles à suivre par les constructeurs. En fait, pour que deux ou plusieurs dispositifs puissent communiquer en liaison point-à-point ou à travers un réseau, il doit y avoir des normes sur la manière dont ils sont censés réagir aux différentes commandes et aux données qui y sont associées, et non seulement aux informations qui sont simplement transférées de l'un à l'autre.
- Définissant la sémantique des fichiers et des services qui y sont associés pour permettre une communication à travers les différents supports amovibles (systèmes non connectés : *off-line*).
- Définissant de manière très explicite les conditions de conformité pour son implémentation. De plus, une déclaration de conformité doit obligatoirement être émise avec chaque nouveau dispositif respectant la norme, document qui doit contenir assez d'informations permettant de définir les fonctions d'interopérabilité avec d'autres dispositifs.
- Facilitant les opérations dans un environnement connecté en réseau.
- Etant structurée de manière à faciliter l'introduction de nouveaux services, ainsi il lui est facile de s'adapter aux futures applications d'imagerie médicale.
- En se basant sur des normes internationales déjà existantes (le modèle OSI par exemple, paragraphe II.10).

II.4 Historique :

II.4.1 Avant DICOM :

Avant l'apparition du standard DICOM, il existait plusieurs problèmes liés à l'archivage des images et à la communication entre les machines des différents constructeurs. En effet, chaque constructeur possédait ses formats propriétaire, qui n'était donc lisible que par les machines de ce même constructeur, il se posait donc le problème d'interopérabilité entre les machines, cela rendait impossible la communication entre les médecins, les hôpitaux et même les services d'un même hôpital si les équipements étaient de marques différentes.

L'autre problème qui se posait était que les différentes informations liées à l'image médicale telles les informations relatives au patient (nom, âge, sexe, antécédents...), le nom du médecin traitant, les rapports d'examen... étaient stockées séparément de l'image. Cette manière hétérogène de stocker les informations menait inévitablement à des erreurs d'archivage, donc à de la perte d'information utile. La probabilité qu'une erreur pareille ne se produise est assez importante vu le nombre important d'images présentes dans chaque hôpital, voir chaque service.

Dans plusieurs cas, le médecin traitant consulte ses collègues pour émettre son diagnostic, or il n'y a que son nom qui figure dans le rapport qu'il fait, pourtant il prend en considération l'avis des autres médecins. En cas de mauvais diagnostic, le médecin traitant est tenu comme le seul responsable alors qu'il a pu être, accidentellement, induit en erreur par un autre collègue.

II.4.2 Les standards ACR-NEMA [12]

Avec le développement des technologies numériques en général et en imagerie médicale particulier à partir des années 1970, l'*American College of Radiology (ACR)* et la *National Electrical Manufacturers Association (NEMA)* ont remarqué la nécessité grandissante de créer une méthode standard facilitant le transfert des images médicales et leurs informations entre machines de différents constructeurs qui, jusque là, possédait des formats propriétaires.

L'ACR et la NEMA ont formé en 1983 un comité conjoint pour créer un standard dans le but de :

- Promouvoir la communication numérique des images et de leurs informations entre les machines quel que soit le constructeur.
- Faciliter le développement et l'essor des systèmes d'archivage et de communication des images (*Picture Archiving and Communication System : PACS*) pouvant dialoguer avec d'autres systèmes de traitement de l'information.

Permettre la création de bases de données de diagnostics, pouvant être consultées par le plus grand nombre de médecins possible, sur une surface la plus étendue possible.

En 1985, une première version du standard, fut publiée, « *ACR-NEMA Standards Publication No. 300-1985* ». Elle fut désignée version 1.0 et, comme toute première version, contenait des erreurs. Deux révisions la suivirent, la première en Octobre 1986, et la seconde en Janvier 1988.

Au courant de l'année 1988, une seconde version du standard fut publiée, « *ACR-NEMA Standards Publication No. 300-1985* ». Elle incluait la première version ainsi que ses deux révisions publiées. En plus d'autres révisions apportées, elle contenait de nouveaux outils pour :

- Faciliter le support des dispositifs d'affichage
- Définir un nouveau schéma hiérarchique pour la présentation des informations relatives à l'image
- Ajouter des données supplémentaires à l'image afin d'accroître sa spécificité.

Ces publications définissaient aussi une configuration matérielle et logicielle minimale pour permettre son exploitation correctement par les constructeurs.

II.4.3 Le standard DICOM [12]

Ce standard, aussi développé initialement par le comité *ACR-NEMA* et sorti en 1991, englobe un certain nombre d'améliorations considérables par rapport aux versions antérieures. Ces améliorations notables sont :

- Il est applicable dans un environnement connecté en réseau. Les standards *ACR-NEMA* n'étaient applicables que lors de connexions point-à-point. Pour pouvoir les utiliser en réseau, il fallait une unité d'interface réseau (*Network Interface Unit : NIU*). Le standard DICOM quant à lui est capable d'opérer dans un réseau en utilisant, par exemple, le standard TCP/IP.

- Il est applicable utilisant différents supports amovibles (CDs, DVDs...). Les standards *ACR-NEMA* ne définissaient pas de format de fichier, de choix de support physique ou de système de fichiers logique. DICOM supporte quant à lui l'utilisation des standards de l'industrie, physiques tels les CD-R, ou bien les systèmes de fichiers logiques, tel le *PC File System FAT 32*.
- Il spécifie comment les dispositifs revendiquant la conformité à la norme réagissent aux commandes et aux données échangées. Les standards *ACR-NEMA* étaient limités au transfert des données, mais DICOM spécifie, par le concept des classes de service (*Service Classes*), la sémantique des commandes et des données associées.
- Il spécifie des niveaux de conformité. Les standards *ACR-NEMA* avaient spécifié un niveau minimum de conformité. DICOM décrit explicitement comment un constructeur doit structurer une déclaration de conformité (*Conformance Statement*) pour choisir des options spécifiques.
- Il est structuré en plusieurs parties sur un document. Ceci facilite l'évolution du standard dans un environnement en pleine évolution en simplifiant l'ajout de nouvelles fonctionnalités. Les directives de l'ISO définissant la manière dont doit être structuré un document en plusieurs parties ont été suivies lors de la construction de la norme DICOM.
- Il introduit des informations explicites pour les images, graphes, formes d'ondes, rapports, impressions...
- Il spécifie une technique établie pour l'identification unique de chaque information. Ceci facilite la définition sans ambiguïté des relations entre les différentes informations.

C'est cette dernière version du Standard qui a été maintenue, avec quelques modifications au fil des années pour s'adapter aux nouvelles fonctionnalités et capacités des dispositifs.

II.5 Comité DICOM [13]

Initialement, le comité DICOM n'était composé que de l'ACR et de la NEMA. Cependant, au cours des années, il s'est élargi avec la participation d'autres organisations du monde de l'industrie et du milieu académique à son développement. Cette collaboration a permis, d'une part, d'élargir l'éventail des applications DICOM dans le milieu médical et a

ouvert la voie à d'autres modalités de l'imagerie médicale (résonance magnétique, ultrasons, milieu dentaire...), et d'autre part, de sortir des frontières des Etats-Unis.

Le tableau II.1 énumère les acteurs collaborant à l'amélioration du standard.

Acteurs industriels	Acteurs académiques ou institutionnels
AGFA U.S. Healthcare	American Academy of Ophthalmology
Boston Scientific	American College of Cardiology
Camtronics Medical systems	American College of Radiology
Carl Zeiss Meditec	American College of Veterinary Radiology
DeJarnette Research Systems	American Dental Association
Eastman Kodak	College of American Pathologists
ETIAM	Deutsche Röntgengesellschaft
FujiFilm Medical Systems U.S.A.	European Society of Cardiology
GE Healthcare	Healthcare Information and Management
Heartlab	Systems Society
Hologic	Medical Image Standards Association of Taiwan
IBM Life Sciences	Societa Italiana di Radiologia Medica
Konica Minolta Medical Corporation	Société Française de Radiologie
Matrix View	Society for Computer Applications in Radiology
McKesson Medical Imaging Company	Canadian Institute for Health Informatics
MEDIS	Center for Devices & Radiological Health
Merge eMed	Japan Industries Association of Radiological
Philips Medical Systems	Systems (JIRA)
RadPharm	Korean PACS Standard Committee
R2 Technology, Inc.	National Cancer Institute
Sectra Imtec AB	National Electrical Manufacturers Association
Siemens Medical Solutions USA, Inc.	
Sony Europe	
Toshiba America Medical Systems	

Tableau II.1 : Liste des organisations participant au comité DICOM [12]

De grands constructeurs de l'imagerie médicale contribuent activement au développement du standard. Il existe aujourd'hui 26 groupes de travail au sein du comité DICOM réunissant environ 750 experts techniques ou médicaux. La liste de ces groupes figure dans le tableau II.2 :

Groupes de travail du comité DICOM	
WG-01 : Cardiac and Vascular Information	WG-14 : Security
WG-02 : Projection Radiography and Angiography	WG-15 : Digital Mammography and CAD
WG-03 : Nuclear Medicine	WG-16 : Magnetic Resonance
WG-04 : Compression	WG-17 : 3D
WG-05 : Exchange Media	WG-18 : Clinical Trials and Education
WG-06 : Base Standard	WG-19 : Dermatologic Standards
WG-07 : Radiotherapy	WG-20 : Integration of Imaging and Information Systems
WG-08 : Structured Reporting	WG-21 : Computed Tomography
WG-09 : Ophtalmology	WG-22 : Dentistry
WG-10 : Strategic Advisory	WG-23 : Application Hosting
WG-11 : Display Function Standard	WG-24 : Surgery
WG-12 : Ultrasound	WG-25 : Veterinary Medicine
WG-13 : Visible light	WG-26 : Pathology

Tableau II.2 : Liste des groupes de travail au sein du comité DICOM [12]

II.6 Propriétés des fichiers DICOM

II.6.1 L'orientation Objet

En premier lieu, précisons que DICOM est orientée objet, pour cela on désigne par « objet information » (*Information Object*) une information dans le fichier (image, nom du patient...) et par classe de service (*Service Class*) appliqué à l'objet toute opération sur cette information.

II.6.2 Les *UIDs* [14]

Avant d'aller plus loin, il est important de définir un paramètre primordial utilisé dans DICOM : l'identifiant unique (*Unique Identifier UID*). C'est un identifiant généré automatiquement par chaque machine conforme DICOM à l'aide d'une technique qui garantit son unicité. L'*UID* identifie une information et, du fait de son unicité, permet de la préserver d'éventuelles erreurs. Les identifiants sont obligatoirement présents dans chaque fichier DICOM. Il ne peut exister deux *UIDs* identiques pour désigner deux informations différentes quelque que soit la machine ou sa localisation. Ainsi l'identifiant d'une série d'images est spécifique à un patient, un examen, une date, un hôpital et une machine. En résumé, l'*UID* est nécessaire pour des raisons médicales et médico-légales, et permet aussi la création et la gestion de bases de données.

Il existe quatre *UIDs* obligatoires dans chaque fichier DICOM :

- ***SOP Class UID*** : Identifiant du service auquel est destinée l'image.
- ***Study instance UID*** : Identifie un examen entier en temps et en lieu.
- ***Series Instance UID*** : Identifie une série d'images dans l'examen.
- ***SOP Instance UID ou Image UID*** : Identifie l'image associée au fichier.

II.6.3 Les Objets de définition des informations [14]

Le principe des *IODs* (*Information Object Definition*) est un modèle de données abstrait orienté objet utilisé pour spécifier des informations sur des éléments du monde réel. Un *IOD* fournit aux entités applicatives (*AE*) communicantes une vue d'ensemble sur les informations échangées.

Un *IOD* ne représente pas une information spécifique d'un élément du monde réel mais plutôt une classe d'éléments possédant les mêmes propriétés. Un *IOD* utilisé pour représenter une seule classe d'éléments réels est appelé Objet Information Normalisé (*Normalized Information Object*). Un *IOD* incluant plusieurs éléments du monde réel en relation entre eux est appelé Objet Information Composé (*Composite Information Object*). Un *IOD* contient plusieurs attributs (*attributes*). Ces attributs décrivent les propriétés de l'information sur l'élément du monde-réel représenté.

II.6.4 Codage des attributs [12]

Les attributs sont encodés, comme toutes les données textuelles dans DICOM, de deux manières :

II.6.4.1 Codage implicite

Dans le codage implicite, l'information est codée sur 3 champs comme suit :

- Champ 1 : Balise contenant un code de 32 bits (16 bits (groupe), par exemple le groupe 0010(h) est le groupe d'identification du patient + 16 bits (élément)) définissant le type d'information contenue dans le troisième champ.
- Champ 2 : Code de 16 ou 32 bits donnant la longueur en octets du troisième champ.
- Champ 3 : L'information utile codée en ASCII dans le cas d'une chaîne de caractères, sa longueur est bien sur variable.

Exemple :

Information relative au nom du patient : Salim EL BECHIR

On va représenter les valeurs en hexadécimal :

(0010,0010) | 0000000F | 53 61 6C 69 6D 20 45 4C 20 42 45 43 48 49 52

II.6.4.2 Codage explicite

Dans le codage explicite, l'information est codée sur 4 champs comme suit :

- Champ 1 : Balise contenant un code de 32 bits (16 bits (groupe) + 16 bits (élément)) définissant le type d'information contenue dans le troisième champ.
- Champ 2 : Code sur 16 ou 32 bits contenant une étiquette de 2 lettres représentant le type d'information (PN pour Patient's Name, DA pour patient's birth date...etc), ces codes sont aussi définis par le standard.
- Champ 3 : Code de 16 ou 32 bits donnant la longueur en octets du quatrième champ.
- Champ 4 : L'information utile codée en ASCII dans le cas d'une chaîne de caractères, sa longueur est bien sur variable.

Exemple :

On va utiliser le même exemple qu'avant, information relative au nom du patient : Salim EL BECHIR

On va représenter les valeurs en hexadécimal :

(0010,0010) | 50 4E | 0000000F | 53 61 6C 69 6D 20 45 4C 20 42 45 43 48 49 52.

N.B : Lors du transfert d'un fichier DICOM, il doit impérativement être précisé si le codage utilisé est implicite ou explicite pour éviter une interprétation erronée des informations.



Figure II.1 : Codage des attributs dans DICOM

II.6.5 Les SOP Class [15]

Le traitement d'une information DICOM s'effectue en associant un objet avec une classe de service en créant ainsi une paire objet/service (*Service/Object Pair SOP*). Par exemple : le service « imprimer » + l'objet « image » donne la *SOP* « imprimer l'image ».

L'application d'un service donnée à un objet donné constitue une *SOP Class*. Par exemple le service « stocker » + l'objet « une image MR (*Magnetic Resonance*) » donne la *SOP Class* « Stocker une image MR ». Chaque *SOP Class* est définie par son propre *UID*.

En faisant appel à une *SOP Class* sur une machine avec un patient « réel », il y a génération d'une *SOP Instance* (information) qui reçoit son propre *UID*.

Toute machine voulant se conformer à DICOM (ou toute autre *AE* DICOM) doit impérativement pouvoir traiter au moins un type spécifique d'images et effectuer au moins un service pour ainsi pouvoir gérer une *SOP Class*. DICOM définit toutes les paires « objet/classe de service » possibles.

Le processus de communication DICOM comporte l'échange de *SOP Instances* en utilisant des messages DICOM. Ce dernier est la « version communication » des *SOP Class*, il contient les commandes qui utilisent ou fournissent le service spécifié et le code de l'objet information à traiter.

Par ailleurs, la *SOP Class* doit spécifier si le service DICOM est employé en tant qu'utilisateur (*Service Class User : SCU*) ou comme fournisseur (*Service Class Provider : SCP*). Par exemple un scanner **utilise** le service d'impression **fourni** par un reprographe, le scanner est alors doté d'une *SCU* pour le reprographe, de son côté le reprographe est doté d'une *SCP* pour le scanner.

II.6.6 Les services (Service Class) [12]

Les services sont les opérations qu'il est possible d'effectuer sur les objets information. DICOM utilise les services suivants :

- **Vérification (*Verification*)** : Permet de vérifier si deux machines communiquent correctement.
- **Stockage (*Storage*)** : Facilite le transfert et la sauvegarde d'informations entre deux dispositifs. Permet aussi à une entité DICOM d'envoyer et/ou de recevoir des images, formes d'ondes, rapports...
- **Stockage sur Support amovible (*Media Storage*)** : Spécifie l'échange entre deux dispositifs par l'intermédiaire de supports amovibles.
- **Interroger/Retrouver (*Query/Retrieve*)** : Implémente des commandes de type : FIND, GET et MOVE. FIND permet de demander une liste d'images alors que GET et MOVE permettent d'entamer un transfert qui sera effectué via la classe *Storage Service Class* citée ci-dessus.
- **Notification du contenu d'un examen (*Study Contents Notifications*)** : Utilisée pour notifier l'arrivée d'une nouvelle image ou série d'images. Elle peut être utilisée pour initier un transfert ou vérifier si le transfert d'une image ou d'une série d'images est complet.
- **Gestion d'impression (*Print Management*)** : Permet la connexion avec un reprographe, elle sert aussi à donner les spécifications de l'image (couleur, niveaux de gris, taille...)
- **Gestion des examens (*Study Management*)** : Permet la gestion des examens (création des rendez-vous, suivi des patients...)
- **Gestion des résultats (*Result Management*)** : Permet la gestion des résultats des examens (rapports des médecins, commentaires...).
- **Engagement de stockage (*Storage Commitment*)** : Lors de l'utilisation de la *Storage SOP Class* pour un transfert, il n'y a aucun engagement entre les deux

entités utilisant ce transfert pour garder la *SOP Instance* transmise (elle peut être supprimée après l'envoi ou à la réception après son utilisation une seule fois). La *Storage Commitment SOP Class* permet à l'émetteur de demander au récepteur de s'engager à sauvegarder la *SOP Instance* transmise pour un certain laps de temps ou de manière permanente, comme pour un système d'archivage par exemple.

- **Gestion des listes de travail (*Basic Worklist Management*)** : Facilite l'accès aux listes de travail (*worklists*) et permet leur gestion. Une liste de travail contient les informations relatives à un ensemble de tâches à accomplir, spécifiant les détails de chaque tâche ainsi que son ordre de priorité d'exécution. Un exemple simple d'une liste de travail est celle présente dans un service d'imagerie et contenant les examens programmés.
- **Stockage d'une copie logicielle de l'état d'affichage (*Softcopy Presentation State Storage*)** : Ajoute une fonctionnalité supplémentaire à la *Storage SOP Class* consistant en la capacité de transporter un état d'affichage (*Presentation State*) souhaité ou d'enregistrer un état existant. Un état d'affichage est défini par différents paramètres tels niveaux de gris/couleurs, fenêtrage... Cette *SOP Class* permet de spécifier les paramètres suivants :
 - L'espace de sortie des niveaux de gris.
 - L'espace de sortie des couleurs.
 - La zone de l'image à afficher, à inverser ou à tourner.
 - Les données textuelles à afficher avec l'image.
 - Afficher deux ensembles d'images sur une seule fenêtre.
- **Stockage des rapports structurés (*Structured Reporting Storage*)** : Extension de la *Storage Class*, elle donne plus de possibilités au comportement du récepteur.
- **Journal d'évènements (*Application Event Logging*)** : Facilite le transfert, sur un réseau, des enregistrements des journaux d'évènements de manière à ce qu'ils soient enregistrés sur le même dispositif central ; un serveur centralisé par exemple.
- **Interrogation des informations pertinentes sur le patient (*Relevant Patient Information Query*)** : Facilite l'accès aux informations pertinentes relatives au patient telles qu'elles ont été enregistrées aux différents examens.
- **Notification de la disponibilité d'une information (*Instance Availability Notification*)** : Permet à une *AE DICOM* de notifier une autre de la présence et de la disponibilité d'une *SOP Instance*.

- **Gestion de la création de supports (*Media Creation Management*)** : Définit un mécanisme qu'un émetteur peut utiliser pour donner une instruction à un dispositif de créer un support d'échange contenant un ensemble de *SOP Instances* qui ont déjà été transférées au dispositif de création du media par le biais de la *Storage service Class*.
- **Stockage du protocole d'attachement (*Hanging Protocol Storage*)** : Permet l'envoi d'une *Hanging Protocol SOP Instance* d'une *AE* à une autre. Le protocole d'attachement est utilisé pour l'attachement des fichiers DICOM en pièce jointes à des e-mails.
- **Interrogation/Recherche d'un protocole d'attachement (*Hanging Protocol Query/Retrieve*)** : Facilite l'accès aux *Hanging Protocol SOP Instances*.

II.7 Modèle d'information DICOM

DICOM définit une structure bien précise pour la représentation des informations médicales. Cette structure définit les relations entre les services et les objets, elle est organisée en hiérarchie avec au sommet un service et à la base les informations « brutes ».

N.B : Dans cet organigramme et suivant la norme DICOM et les directives ISO, les rectangles désignent les entités et les losanges désignent les relations les reliant.

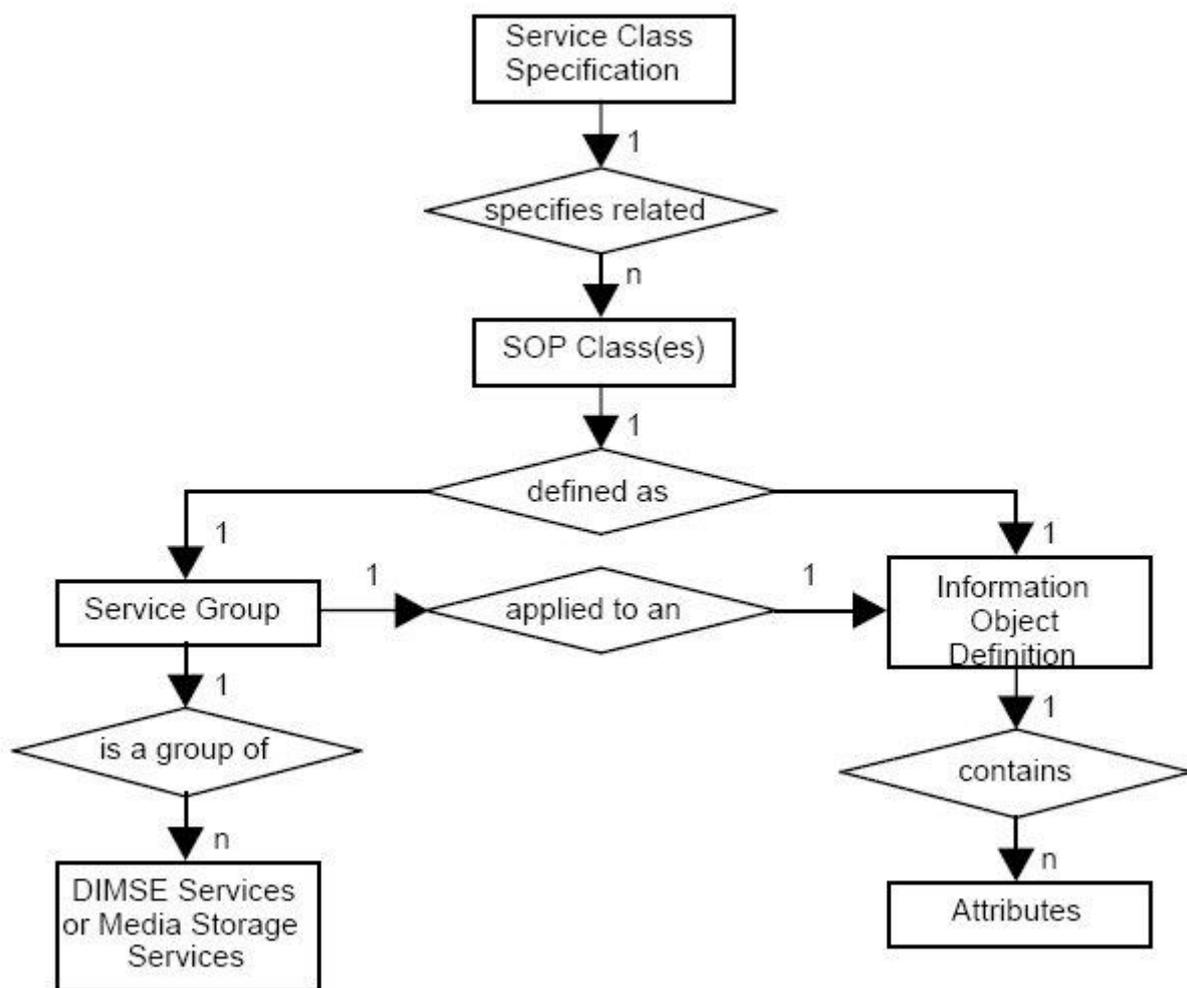


Figure II.2 : Modèle d'information DICOM (*DICOM Information Model*) [12]

Explication du modèle :

Au sommet, une classe de services possède ses propres spécifications, elle est en relation avec plusieurs *SOP Class* du fait de la présence de plusieurs *IODs* utilisant cette classe. Chaque *SOP Class* définit un groupe de services appliqués à un *IOD*. Un *IOD* contient plusieurs attributs qui représentent l'information exploitable. De son côté, le groupe de services contient des services de gestion des supports de stockage et/ou des services de communication définis via les *DIMSEs*.

II.8 Structure des fichiers DICOM [12]

La structure des fichiers issus de la norme DICOM remédie très efficacement aux problèmes qui existaient avant son apparition.

La première remarque qu'on fera, qui est la plus évidente, est qu'étant donné le fait que DICOM soit une norme adoptée par les constructeurs, il n'y a plus de souci d'interopérabilité entre les machines.

Autre point plus important, un fichier DICOM contient en plus de l'image médicale, plusieurs données textuelles relatives à l'image, le patient, l'examen... Cette manière homogène de stocker toutes les données en un seul fichier facilite grandement leur stockage et accroît de manière très significative la sauvegarde des informations et diminue la probabilité de perte.

On peut donc considérer le fichier DICOM comme un dossier médical complet.

En général, un examen médical s'effectue en **une** série de **plusieurs** tests, chaque fichier DICOM contient l'identifiant de la série à laquelle il appartient, il est ainsi aisé de stocker et de rassembler les fichiers d'une même série ensemble. Il est à noter, bien sur, que l'identifiant de la série est unique. Il existe aussi un autre identifiant unique relatif à chaque fichier DICOM.

De la sorte, chaque image est indépendante, même si elle est transférée ou renommée, les informations qu'elle contient restent intactes.

Le fait que chaque médecin, donnant son diagnostic sur l'image, laisse sa trace dans le fichier implique qu'il a une part de responsabilité dans le diagnostic final établi par le médecin traitant. En cas de mauvais diagnostic, il existe plus d'informations permettant d'éviter une telle erreur à l'avenir.

Un fichier DICOM est composé comme suit, il y a 128 zéros d'abord, ensuite viennent les caractères « DICM » en ASCII. Après viennent les informations textuelles et les pixels de l'image. Parmi les informations textuelles, on trouve :

- Les coordonnées du patient.
- La date et le type d'examen.
- Les références de la machine utilisée.
- Le nom du médecin ou du technicien qui effectue l'examen.
- Le nom de l'hôpital.
- Les données d'acquisition de l'image.
- L'identifiant de la série à laquelle appartient l'examen.

- Les rapports des diagnostics effectués par le/les médecin(s) traitant(s).

Une fausse idée reçue très répandue dit que dans le fichier DICOM, il y a un entête contenant les informations textuelles et qu'ensuite viennent les pixels de l'image. Le fichier DICOM contient des champs d'informations et la matrice représentative de l'image est un de ces champs, le dernier. L'*IOD* matrice image est identifié par la balise (7FE0, 0010).

II.9 Particularités d'une image DICOM [12]

Une image DICOM peut être représentée en niveaux de gris ou en couleurs en utilisant un des systèmes RVB ou CMJN ou bien en utilisant une palette de couleurs. Un pixel est représenté sur 8 bits, 16 bits ou plus suivant le nombre de nuances à représenter. DICOM permet aussi le réglage du fenêtrage [17]. Le principe du fenêtrage consiste à ne pas lire l'intégralité de la matrice image en ne sélectionnant que certaines informations. Cette particularité est plus qu'appréciable dans le milieu médical où il est important de considérer le maximum de nuances possibles. Par exemple, l'image d'un organe peut nécessiter un échantillonnage de 0 à 4096 alors qu'il n'est possible de visualiser que 256 niveaux à l'écran. Le fenêtrage nous permet donc d'avoir 16 représentations différentes de cette image.

Les pixels d'une image DICOM peuvent être stockés de manière native (sans compression), avec une compression sans pertes ou avec pertes. DICOM prévoit l'utilisation de plusieurs types de compression. La méthode de compression est spécifiée dans la syntaxe de transfert. Chaque compression possède son propre *UID* de la syntaxe de transfert (*Transfer Syntax UID*). Les méthodes de compression suivantes sont actuellement supportées par DICOM [13] :

JPEG : Standard développé par l'ISO. Il définit plusieurs processus de compression sans pertes (*lossless*) et avec pertes (*lossy*). Il utilise la transformée en cosinus discrète (*Discrete Cosine Transform : DCT*) dans le cas de compression avec pertes et est basé sur la méthode *DPCM (Differential Pulse Code Modulation)* dans le cas de compression sans pertes. DICOM est compatible avec quatre des processus JPEG :

- *Baseline, lossy*
- *Extended, lossy*
- *Lossless, non-hierarchical*
- *Lossless, non-hierarchical, first-order-prediction.*

JPEG-LS : Standard développé par l'ISO. Il peut aussi être utilisé en mode avec pertes ou sans pertes. Il spécifie un processus unique fondé sur une méthode prédictive basée sur un modèle statistique. Il est réputé être plus efficace que JPEG. DICOM utilise les deux modes du JPEG-LS, à savoir sans pertes et avec pertes.

JPEG2000 : Standard développé par l'ISO. Travaille en mode sans pertes ou avec pertes. Il est fondé sur la transformée en ondelettes discrète et sur des transformations multi-composantes, applicables notamment aux images en couleur. Il utilise aussi des schémas de prédiction de type *DPCM* et des transformations plus complexes comme la Transformée de *Karhunen-Loeve*. DICOM a retenu quatre des nombreux procédés de JPEG2000. Les deux premières font appel à la première partie du standard JPEG2000 en mode avec pertes et sans pertes. Les deux autres, étendent les possibilités des deux premières en généralisant le codage multi-composantes à toutes les séquences d'images.

N.B : La première partie du standard JPEG2000 n'applique le codage multi-composantes qu'aux images en couleur.

Compression d'image progressive (JPEG2000 Interactive Protocol : JPIP) : Cette possibilité répond au besoin de transmettre les images de façon progressive, en permettant l'affichage des données avec une précision croissante au fur et à mesure de la transmission, afin de permettre à l'utilisateur de voir l'image avant la fin du transfert et de lui permettre de l'interrompre si celui-ci n'est plus nécessaire.

Du point de vue décodage et simplicité des applications de lecture et d'affichage, il est souhaitable de garder la matrice représentative sous forme native. Rappelons que pour qu'une *AE* puisse lire correctement une image compressée à partir d'un fichier DICOM, elle doit être compatible avec la méthode de compression utilisée. Cependant, vu le nombre croissant des images médicales dans les hôpitaux, et donc l'espace important qu'elles utilisent, la compression commence à gagner du terrain et l'intérêt qu'on lui porte commence à grandir.

II.10 Une norme orientée réseau

L'évolution de l'imagerie médicale a vu évoluer, en parallèle, le monde de l'informatique en général et des réseaux en particulier, surtout avec le concept client/serveur qui a mené à l'avènement de réseaux de plus en plus étendus tel Internet. Au sein des institutions médicales, les systèmes d'information se sont aussi développés et ont vu l'installation de réseaux étendus et performants. C'est ainsi que DICOM s'est orientée vers le domaine du réseau. Il a fallu alors standardiser la norme suivant le modèle OSI « Open

Systems Interconnection » de l'ISO. Rappelons que le modèle OSI définit 7 couches différentes qui sont [16] :

- La couche « physique » est chargée de la transmission effective des signaux entre les interlocuteurs. Son service est typiquement limité à l'émission et la réception d'un bit ou d'un train de bit continu (notamment pour les supports synchrones).
- La couche « liaison de données » gère les communications entre 2 machines adjacentes, directement reliées entre elles par un support physique.
- La couche « réseau » gère les communications de proche en proche, généralement entre machines : routage et adressage des paquets.
- La couche « transport » gère les communications de bout en bout entre processus (programmes en cours d'exécution).
- La couche « session » gère la synchronisation des échanges et les « transactions », permet l'ouverture et la fermeture de session.
- La couche « présentation » est chargée du codage des données applicatives, précisément de la conversion entre données manipulées au niveau applicatif et chaînes d'octets effectivement transmises.
- La couche « application » est le point d'accès aux services réseaux, elle n'a pas de service propre spécifique et entrant dans la portée de la norme.

DICOM s'est donc vue enrichie avec des parties définissant des protocoles de communication réseau. Cependant, DICOM ne normalise que la communication, au niveau applicatif, entre machines physiquement connectées. Le protocole ne décrit pas les communications à un niveau physique (connecteurs, câbles...), ni au niveau des protocoles réseau (TCP/IP). La norme DICOM permet donc la communication d'une application à une autre.

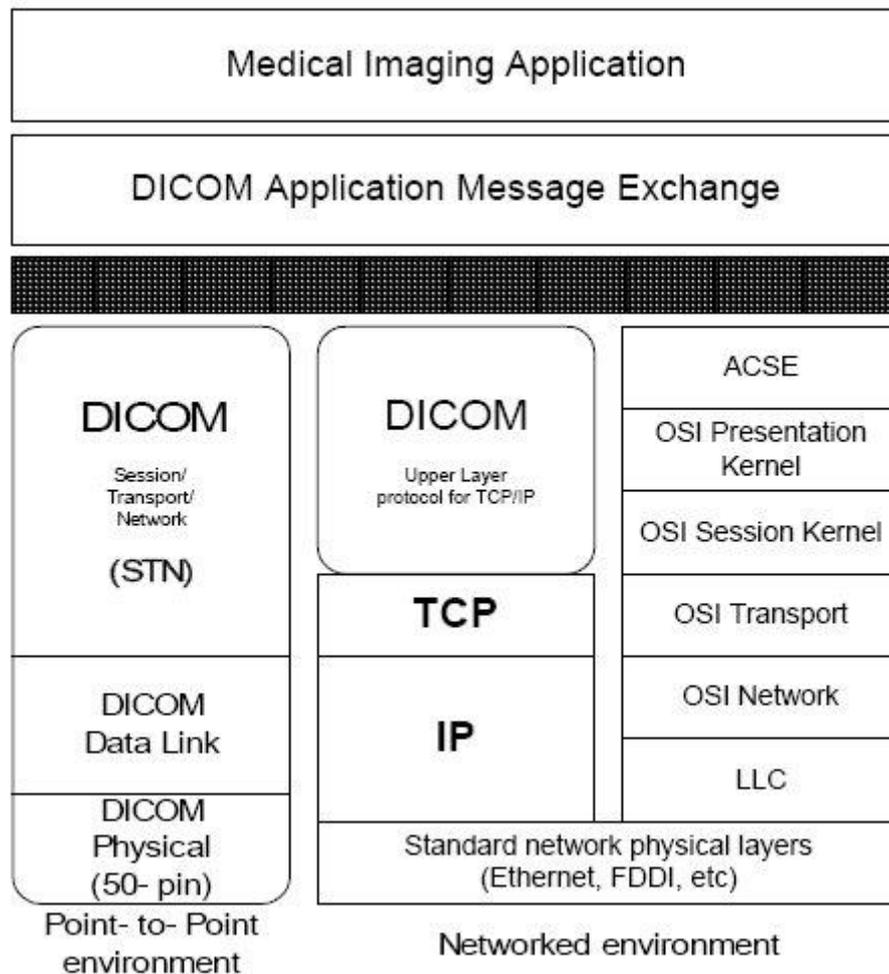


Figure II.3 : Couches réseau gérées par DICOM [17]

II.11 La communication entre les machines [17]

La communication entre machines conformes DICOM s'effectue avec l'échange de messages DICOM (*DIMSEs*). Il existe deux types de *DIMSEs* :

- Les opérations (store, print...).
- Les notifications (rapports d'évènement...).

La communication entre deux dispositifs passe par plusieurs étapes, il y a d'abord établissement de la connexion, ensuite échange des *SOP Class* disponibles et supportées par chaque dispositifs. Ensuite le *SCU* envoie une requête que le *SCP* exécute. Après la fin de l'opération, la connexion est clôturée.

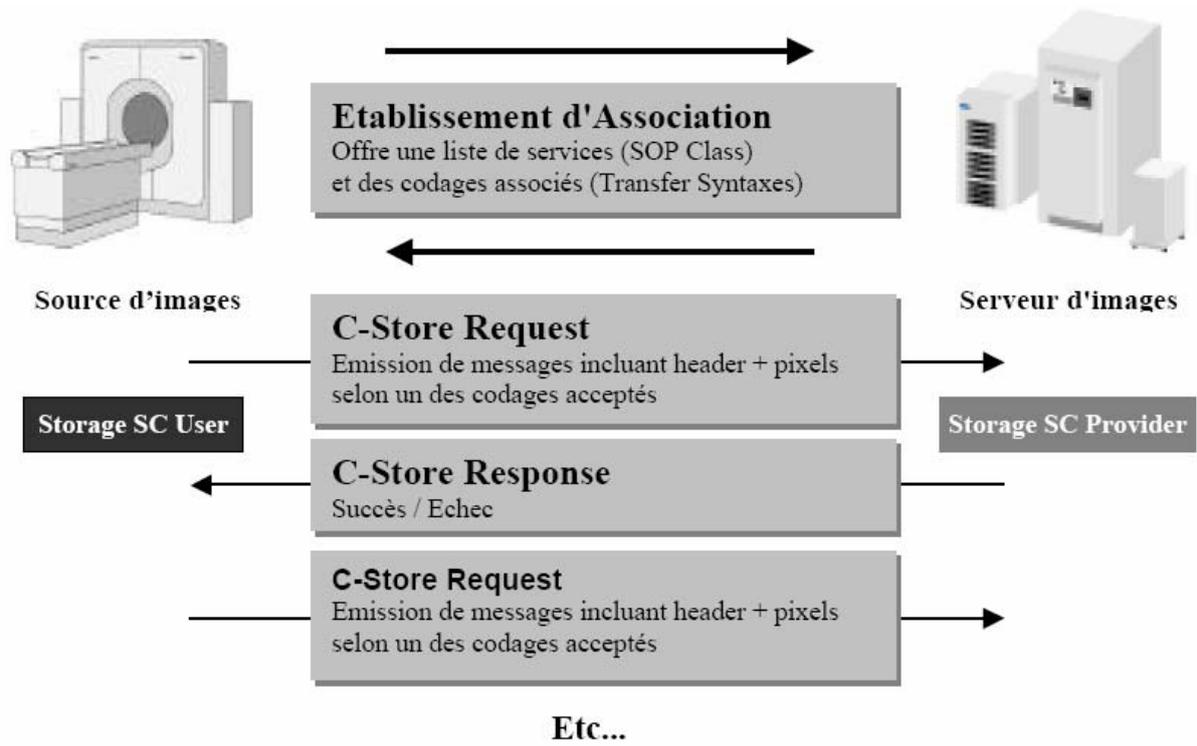


Figure II.4 : Exemple d'une communication entre deux dispositifs
(Opération exécutée : stockage d'images) [17]

II.12 Conclusion :

La norme DICOM a apporté beaucoup d'améliorations au domaine de l'imagerie médicale. Elle a d'un côté simplifié les échanges en supprimant les formats propriétaires, et de l'autre facilité le stockage et l'accès aux différentes informations relatives aux images médicales. Grâce à son procédé d'identification unique robuste, DICOM s'est imposée comme une norme fiable et efficace. De plus, elle a facilité la création de bases de données consultables par les spécialistes de la médecine et a paré à quelques failles médico-légales existantes dans les systèmes qui l'ont précédé.

Autre avantage de cette norme, sa modification est aisée ce qui lui a permis d'être toujours à la page et de suivre l'évolution des technologies de l'information et de l'imagerie médicale.

Malgré le fait que DICOM ait été destiné initialement à la radiologie, son champ d'application s'est largement étendu aux autres applications de la médecine grâce à l'intégration d'autres représentants au sein du comité DICOM.

Le problème majeur de DICOM aujourd'hui est sa complexité, et l'absence du document en d'autres langues que l'Anglais.

Chapitre III

METHODES PROPOSEES

III.1 Introduction

La compression des images est un domaine très actif. De plus en plus de méthodes sont créées, testées et implémentées. Chacune de ses méthodes essaye d'éliminer quelques inconvénients présents dans les autres, cela entraîne malheureusement la création de ses propres inconvénients. C'est pour cette raison qu'on essaye de perfectionner des algorithmes de compression pour des domaines spécifiques, ou encore des modalités spécifiques.

Dans ce chapitre, nous allons présenter les deux méthodes de compression que nous avons développées durant notre travail pour la compression des fichiers DICOM, plus précisément, l'image contenu dans ce fichier.

Que ce soit avec ou sans pertes, les deux méthodes ont le même principe :

1. Séparer l'image des autres informations (Voir paragraphe II.8).
2. Traiter l'image à part (compression avec ou sans pertes).
3. Construction du fichier codé en réintroduisant les informations.

C'est la deuxième étape qui différencie les deux méthodes. Pour la méthode sans pertes, l'image est codée par l'algorithme RLE par contre l'image est quantifiée puis codé arithmétiquement, pour la méthode avec pertes.

Rappelons que les images traitées sont des images à niveaux de gris avec une dynamique de 1023 (le cas le plus fréquent).

La programmation a été faite à l'aide de MATLAB 7.7 (R2008b); pour plus de simplicité, nous avons préféré une architecture modulaire pour notre travail. Ainsi, le programme comporte plusieurs fonctions internes qui sont appelées depuis son corps. Il ne traite que les fichiers DICOM qu'il reconnaît, d'abord, par leur extension (.dcm) avant de vérifier leur structure. Si le fichier a une extension DICOM mais n'a pas sa structure, l'opération sera annulée. Pour la décompression, notre programme reconnaît ses propres fichiers codés, avec leur extension et aussi avec un simple tatouage. Si le fichier codé n'est pas généré avec notre programme, l'opération sera aussi annulée.

Notre travail se couronne par la réalisation d'une interface graphique ergonomique qui sera présentée dans le chapitre suivant.

III.2 La méthode sans pertes

Basée principalement sur le codage RLE, cette méthode permet de retrouver pratiquement les valeurs exactes des pixels de l'image avant compression.

III.2.1 La compression

Le schéma général relatif à la compression est donné par la figure suivante :

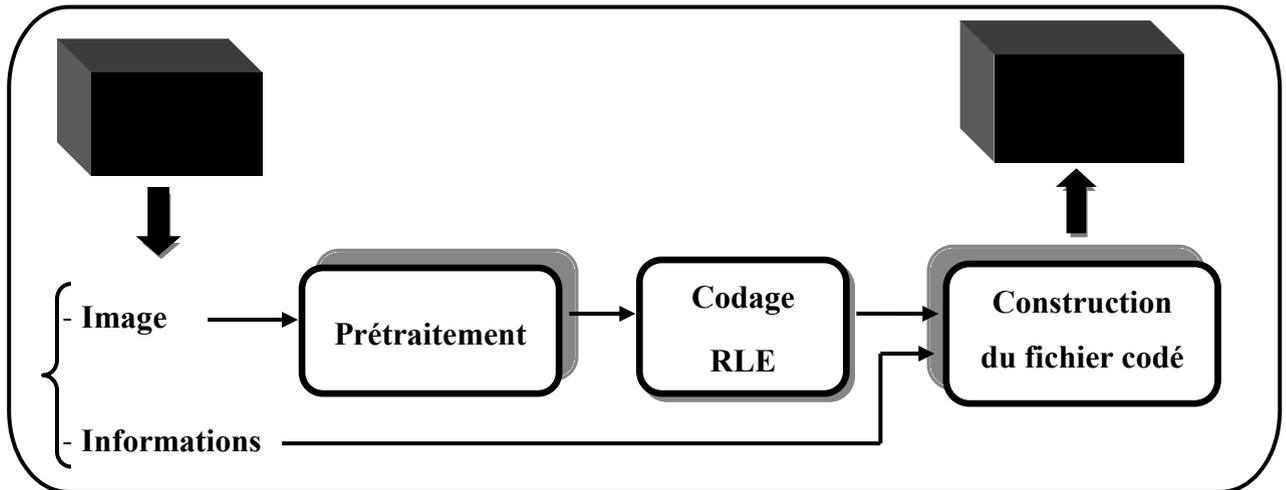


Figure III.1 : Schéma général de la compression Sans perte

Les étapes de cette chaîne sont expliquées avec des illustrations et les commandes Matlab associées :

Etape 0 : Séparer l'image et les informations

En premier lieu, on extrait du fichier DICOM l'image et les informations. Ceci est fait à l'aide des commandes Matlab « dicomread » qui a comme sortie l'image contenue dans le fichier DICOM d'entrée sous forme d'une matrice de dimension égale à la résolution de l'image, et « dicominfo » qui a comme sortie toutes les informations relatives au fichier DICOM d'entrée sous forme d'une structure.

Etape 1 : Le prétraitement

Durant notre travail, nous avons remarqué que toutes les images médicales sont de la même forme, l'objet d'intérêt (cerveau, foie, abdomen...) se trouve centré dans l'image et est entouré de noir (information inutile).

Notre première idée était d'effectuer un léger seuillage sur l'image afin de rendre les nuances proches du noir parfaitement sombres, et nous prévoyions d'utiliser un codage RLE en raison de cette quantité considérable de noir.

Les résultats n'ont pas été concluants, autant pour le taux que pour le temps de compression. La présence du noir et donc des pixels avec des valeurs proches de zéro entraîne un calcul fastidieux et surtout inutile.

L'idée qu'on a adoptée est d'éliminer la zone sombre entourant l'objet d'intérêt, autrement dit, trouver le plus petit rectangle qui l'encadre. Pour définir ce rectangle il nous faut seulement deux points, le plus haut à gauche et le plus bas à droite, ces derniers sont obtenus à partir du contour de l'objet d'intérêt construit à l'aide d'une méthode de détection de contours. Nous avons utilisé une méthode de filtrage utilisant le filtre de Sobel [18] car nous avons remarqué qu'il est très robuste vis-à-vis du bruit présent dans l'image, en outre, il est considéré comme le plus simple et le plus rapide des filtres de détection de contour. La figure suivante représente une démonstration de cette étape :

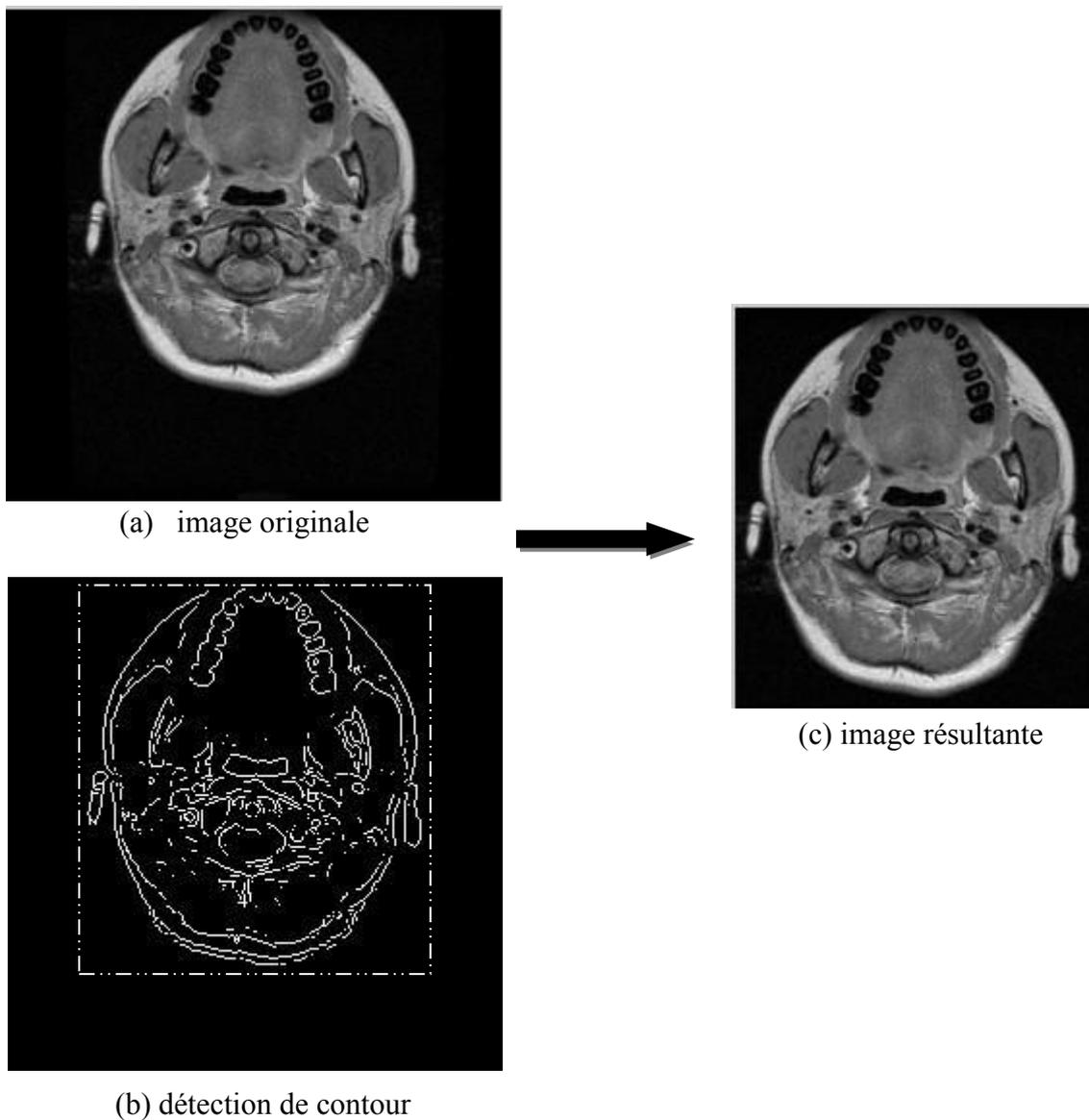


Figure III.2 : Elimination de la zone sombre

On remarque que la dimension de l'image résultante est inférieure à l'originale. Cela permet non seulement d'augmenter le taux de compression final, mais aussi de réduire le temps nécessaire pour la compression puisque toutes les opérations de calcul (addition, multiplication...) sur cette zone ont été éliminées.

Après plusieurs tests, nous avons remarqué que pour quelques images, cette opération a causé des pertes d'information au niveau du contour. Pour y remédier, nous avons rajouté une marge de sécurité de cinq lignes et cinq colonnes de chaque côté du rectangle.

Pour une bonne reconstruction, on gardera le nombre de lignes et/ou de colonnes supprimés de chaque côté de l'image. La zone sombre sera rajoutée lors la reconstruction.

Etape 2 : Le codage RLE

Nous avons opté pour le codage RLE car, dans notre application, cette méthode, expliquée dans le paragraphe I.2.1.3, est plus efficace que les autres.

Avant de procéder au codage, l'image (matrice à deux dimensions) est transformée en un vecteur, en juxtaposant ses colonnes l'une après l'autre. Ce vecteur est par la suite codé avec l'algorithme RLE.

On doit impérativement garder la dimension de l'image avant le codage pour la reconstruire après l'étape le décodage.

Etape 3 : La construction du fichier codé

Le vecteur image codé par l'algorithme RLE généré dans l'étape précédente est utilisé pour construire le fichier codé.

Premièrement, les informations concernant la zone sombre éliminée dans l'étape précédente ainsi que la dimension de l'image obtenue dans l'étape 1 sont ajoutées sur 8 octets tout au début du vecteur image codé.

Les informations relatives au fichier DICOM doivent aussi être ajoutées ; le problème qui se pose est que les informations sont stockées sous forme d'une structure.

Nous avons donc reconstruit un fichier DICOM avec :

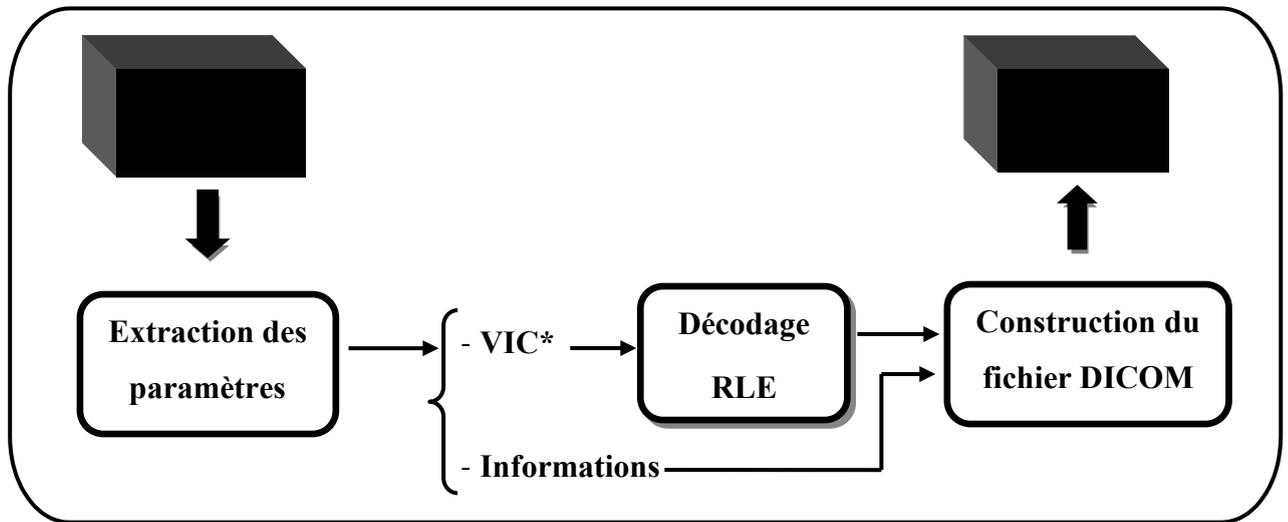
- Le vecteur image codé transformé en matrice,
- Les mêmes informations du fichier DICOM original. Cela est fait grâce à la commande Matlab « dicomwrite ».

Les 128 premiers octets du fichier résultant sont éliminés (Voir paragraphe II.8), et sont remplacés par 6 octets de tatouage contenant le type de compression (avec ou sans pertes) et d'autres caractères spéciaux.

Le fichier compressé est finalement généré avec une extension .μdcm.

III.2.2 La décompression

Le schéma général pour la décompression est donné par la figure suivante :



* **Vecteur Image codé.**

Figure III.3 : Schéma général de la décompression Sans perte

Etape 1 : Extraction des paramètres

Avant toute chose, le fichier compressé est authentifié en vérifiant le tatouage. Si le fichier provient de notre programme, la méthode avec laquelle il a été compressé est identifiée. On recrée un fichier DICOM en réinsérant 128 zéros. A l'aide des commandes Matlab « dicomread » et « dicominfo », on sépare l'image et les informations contenus dans le nouveau fichier généré. Le vecteur image codé est récupéré en transformant la matrice image en un vecteur (colonne par colonne), ce dernier contient les caractéristiques de la zone sombre éliminée durant la compression ainsi que la dimension de l'image à reconstruire.

Les informations sont gradées sous forme de structure.

Etape 2 : Décodage RLE

Le vecteur image codé est décodé avec la méthode RLE. Le décodage RLE est nettement plus rapide par rapport au codage car ce n'est qu'une simple écriture.

Etape 3 : La reconstruction du fichier DICOM

Le vecteur décodé généré de l'étape précédente est transformé en une matrice dont les dimensions ont été récupérées dans l'étape 1.

La zone sombre est par la suite ajoutée (figure III.4).

Rappelons que les nombre de lignes ou de colonne devant être ajoutés de chaque coté ont été récupérés dans l'étape 1.

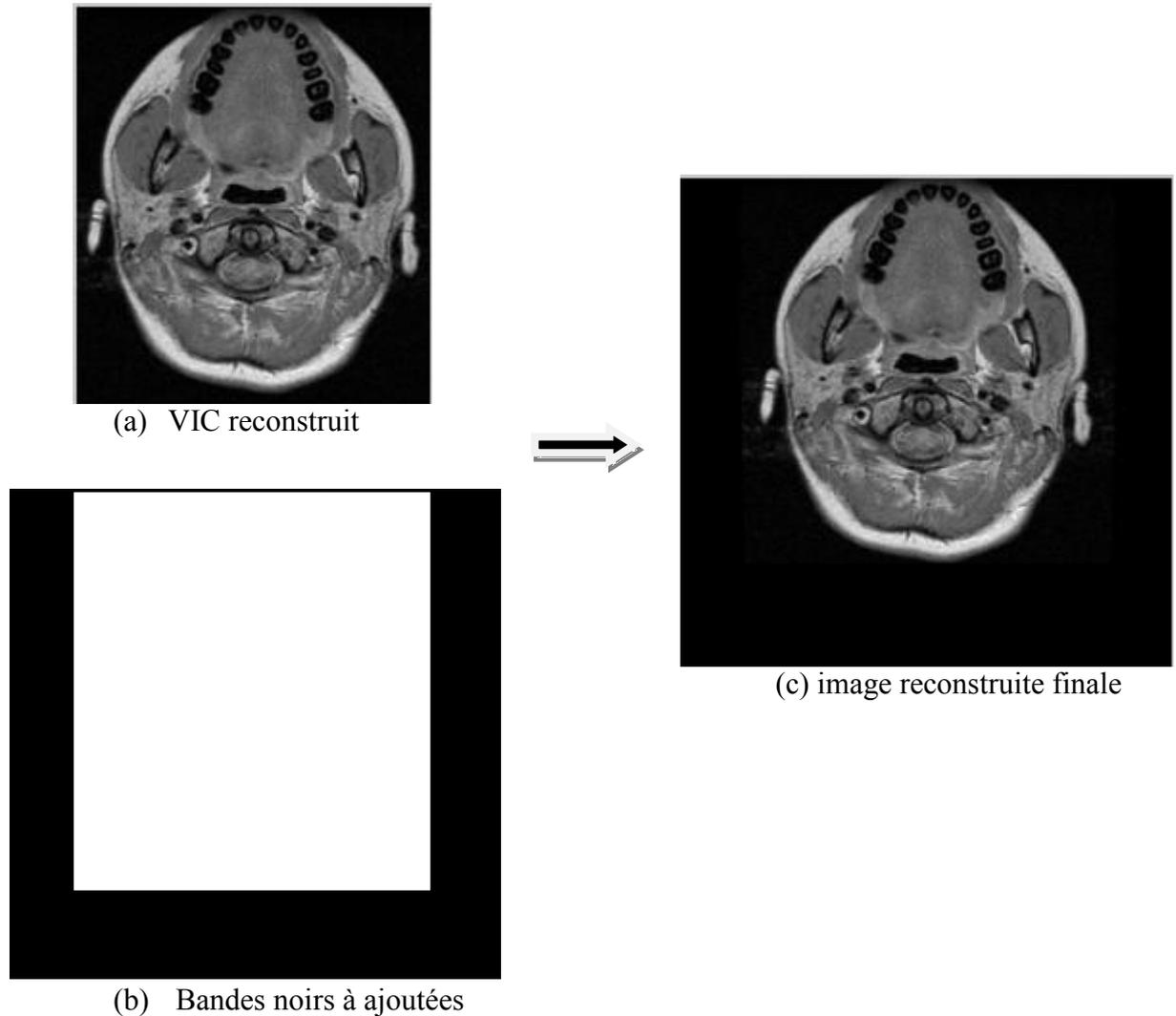


Figure III.4 : Ajout de la zone sombre

A l'aide de la commande Matlab « `dicomwrite` », le fichier DICOM est reconstruit à partir de la dernière image obtenue et des informations récupérées dans l'étape 1. Il a comme extension « `.dcm` ».

III.3 La méthode Avec pertes

Le rapport de compression de la méthode sans pertes n'a pas été satisfaisant (de l'ordre de 4 :1). En plus, pour certaines images de grande taille présentant beaucoup de détails et une importante variété des nuances, le temps de compression était considérable.

La compression avec perte est donc la solution pour contourner cette insuffisance. L'algorithme lié à la compression et la décompression est expliqué dans ce qui suit.

III.3.1 La compression

La méthode avec pertes essaie d'améliorer l'efficacité de la compression en utilisant un algorithme plus performant permettant non seulement d'augmenter nettement les rapports de compression obtenus mais aussi de stabiliser le temps de compression à une résolution donnée. Ces améliorations sont régies par les compromis liés à ce mode de compression (paragraphe I.2.2), la qualité de l'image reconstruite sera donc dégradée.

Comme expliqué dans le chapitre I, le domaine de l'imagerie médicale ne tolère pas une grande dégradation pour les zones sensibles qui, par exemple, représentent une zone pathologique car cela peut entraîner des erreurs de diagnostic, et donc de traitement. La solution la plus utilisée dans ce domaine est les régions d'intérêts (*Region of Interest : ROI*). Une région d'intérêt est une zone qui ne subit pas de quantification (compression), en conséquence on pourra retrouver les valeurs exactes de cette zone lors de la reconstruction. Généralement on utilise cette approche sur des zones critiques comme une tumeur, une fracture, et d'autres anomalies.

Le schéma général relatif à la compression est donné par la figure suivante :

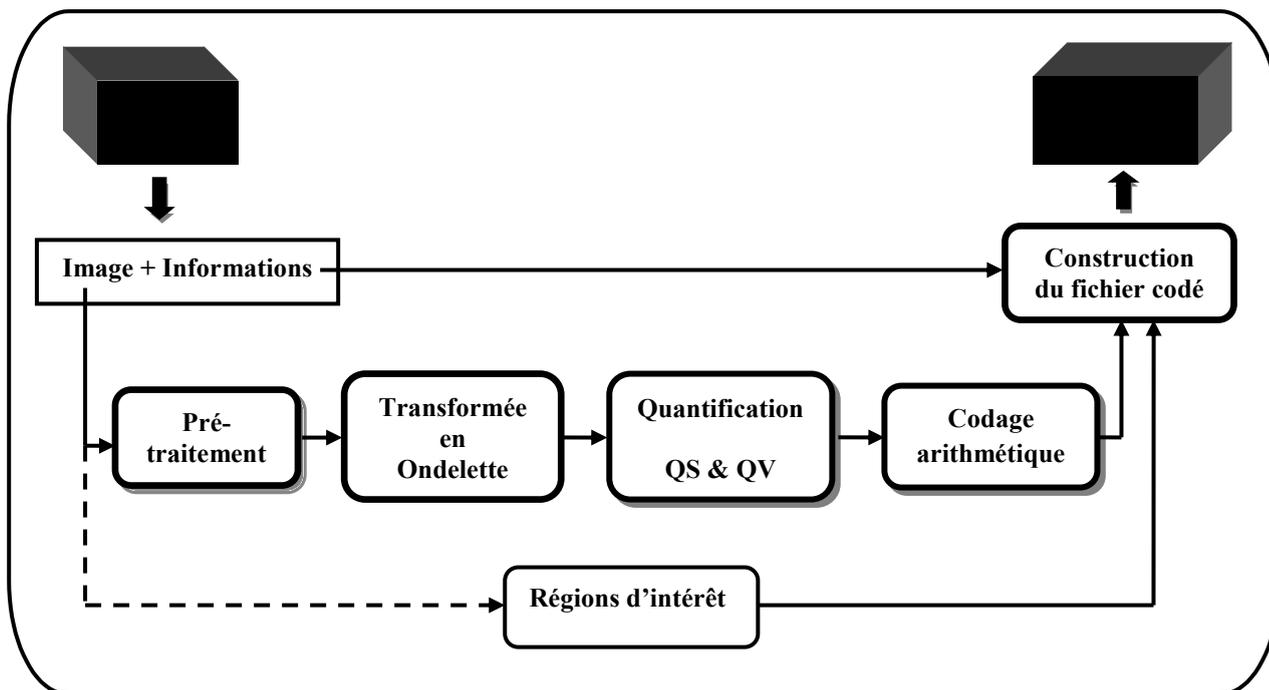


Figure III.5 : Schéma général de la compression Avec perte

Comme nous le voyons, nous allons, à un certain moment de notre algorithme de compression, utiliser une quantification scalaire et vectorielle. Dans le paragraphe I.4.2.3 nous avons vu que le dictionnaire est indispensable lors d'une quantification.

Avant de nous lancer dans l'explication de notre méthode de compression avec pertes, nous allons expliquer l'algorithme que nous avons utilisé pour la construction du dictionnaire : La carte auto organisatrice (Carte de Kohonen, paragraphe I.4.2.3.1.2.3).

Pour la quantification scalaire nous avons utilisé une carte de 10x10 neurones.

La base d'apprentissage était composée de la bande « approximation » de huit images DICOM sur lesquelles nous avons réalisé une décomposition à l'aide de la DWT. Les images utilisées étaient des images IRM du cerveau et de l'abdomen. Toutes ces images avaient une dynamique de 1023.

Pour la quantification vectorielle, nous avons utilisé une carte de 7x7 neurones.

La base d'apprentissage était composée des mêmes images DICOM que pour la base d'apprentissage scalaire sauf que nous avons cette fois pris les bandes « horizontale », « verticale », et « diagonale » résultantes de la décomposition à l'aide de la DWT que nous avons partitionné en vecteurs de dimension 4x4.

Dans les deux cas nous avons utilisé 200 itérations. La fonction d'apprentissage $\beta_{i,k}(t)$ (relation I.6) a été choisie avec les paramètres suivants :

$$\beta_0(t) = 0.2 \frac{T_{max} - t}{T_{max} + t} \quad (III.1)$$

$$\sigma(t) = 1 - \frac{t}{T_{max}} \quad (III.2)$$

Nous avons ainsi obtenu la relation suivante :

$$\beta_{i,k}(t) = \left(0.2 \frac{T_{max} - t}{T_{max} + t} \right) \exp \left(- \frac{\|r_i - r_k\|}{\left(1 - \frac{t}{T_{max}} \right)} \right) \quad (III.3)$$

L'algorithme se déroule comme suit :

Etape 0 : Séparer l'image et les informations

En premier lieu, on extrait du fichier DICOM l'image et les informations, exactement comme l'étape 0 de la compression Sans pertes.

Etape 1 : Les régions d'intérêt

Cette étape ne s'exécute que si le médecin (l'utilisateur en général) juge nécessaire de choisir des régions d'intérêt.

L'utilisateur doit spécifier le nombre de ces zones et les sélectionner l'une après l'autre ; ces zones ne seront pas quantifiées mais seront stockées, avec leurs dimensions et leurs emplacements respectifs, dans le fichier compressé final.

Etape 2 : Le prétraitement

Le prétraitement consiste également à éliminer la zone sombre entourant l'objet d'intérêt (à ne pas confondre l'objet d'intérêt et la zone d'intérêt). La méthode utilisée est la même que celle expliquée dans l'étape 1 de la compression sans pertes, à la seule différence que la résolution de l'image engendrée après cette étape doit être un multiple de 8 ($8n \times 8m$; n et m sont des entiers non nuls) ; ceci assure le bon déroulement des étapes suivantes à savoir la décimation de la transformée en ondelette et le découpage en bloc de 4×4 lors de la quantification vectorielle.

Etape 3 : La transformation

On applique la transformée en ondelettes (un niveau de décomposition) à l'image issue de l'étape précédente, il en résulte quatre autres images (figure III.6) dont la résolution est le quart de celle de l'image initiale.

Nous avons utilisé les filtres de Le Gall 5/3 car ils permettent des opérations de décomposition et de reconstruction sans pertes d'informations. Les coefficients des filtres de Le Gall sont donnés par le tableau suivant :

Filtres de décomposition		
i	Passe-bas	Passe-haut
0	6/8	1
± 1	2/8	-1/2
± 2	-1/8	

Tableau III.1 : Coefficients des filtres Le Gall 5/3 de décomposition [19]

Les nombres 5 correspondent au nombre d'éléments du filtre passe-bas, tandis que le nombre 3 correspond au nombre d'éléments du filtre passe-haut.

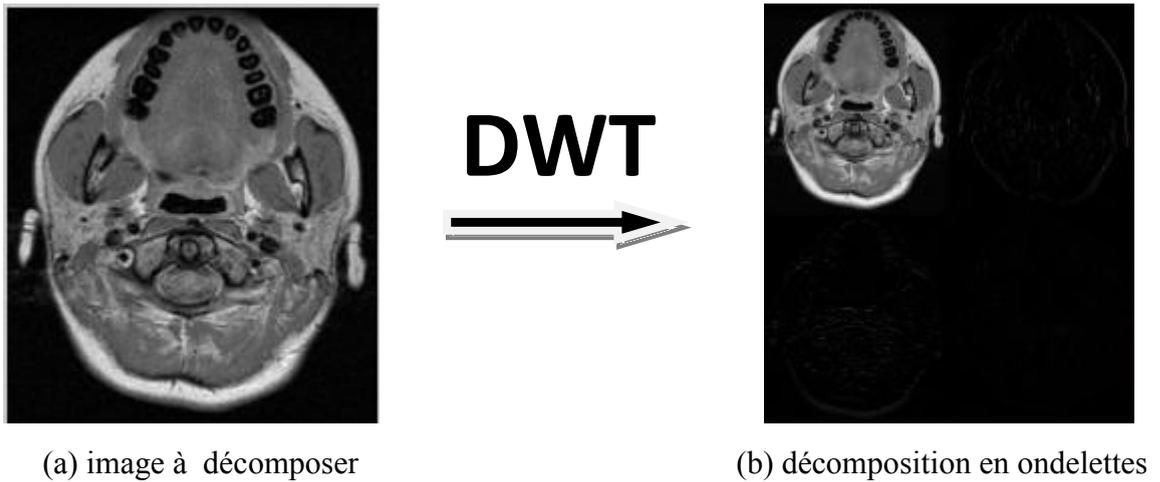


Figure III.6 : Transformation en ondelettes

Cette opération est exécutée à l'aide de la commande Matlab « `dwt2` », en précisant les deux filtres de décomposition en entrée. Le mode d'extension de la transformée doit être modifié au mode « `periodization` » à l'aide de la commande Matlab « `dwtmode` », puisque c'est le seul mode où les dimensions des images à décomposer (resp. à reconstruire) sont divisées (resp. multipliées) par un facteur de 2 lors de l'utilisation des filtres de Le Gall.

Etape 4 : La quantification

L'approximation est quantifiée scalairement afin de garder une bonne qualité, chaque pixel sera remplacé par l'indice de son plus proche voisin dans le dictionnaire scalaire.

Par contre les détails sont quantifiés vectoriellement et pour cela ils sont découpés en bloc de 4x4 dont chacun sera remplacé par l'indice du bloc le plus proche voisin appartenant au dictionnaire vectoriel (figure III.7).

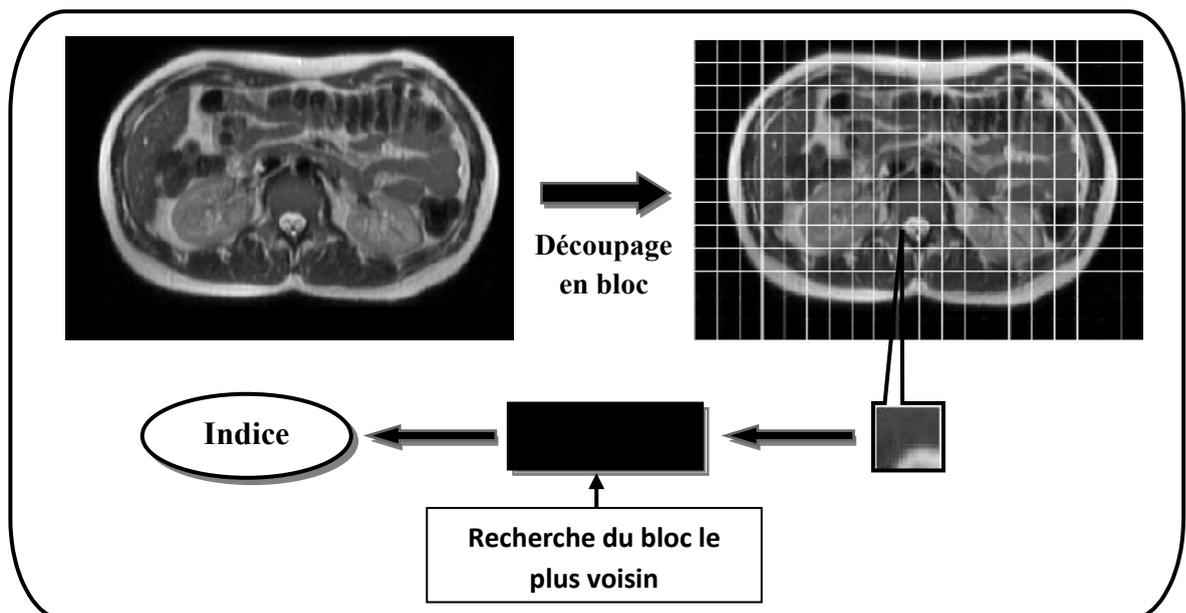


Figure III.7 : Le Principe de la quantification vectorielle

La ressemblance est mesurée à l'aide de la distance euclidienne entre les deux éléments (relation I.6). Si les distances euclidiennes entre un vecteur et deux mot codes différents sont les mêmes le programme choisira un au hasard.

On a pu définir un mot code appartenant à la carte de Kohonen, dont nous avons sauvegardé les dimensions, par un seul indice à la place de deux.

Etape 5 : Le codage arithmétique

Les matrices d'indices obtenues à la sortie de l'étape précédente sont codées arithmétiquement (Paragraphe I.2.1.2)

Avant de procéder au codage, les matrices d'indices sont transformées en vecteurs, non pas colonne par colonne mais en suivant une nouvelle méthode qui nous a permis d'augmenter encore plus le taux de compression avec une négligeable augmentation du temps de compression.

Cette nouvelle façon de lire une matrice et que nous avons nommé la lecture en spirale est représentée dans la figure suivante :

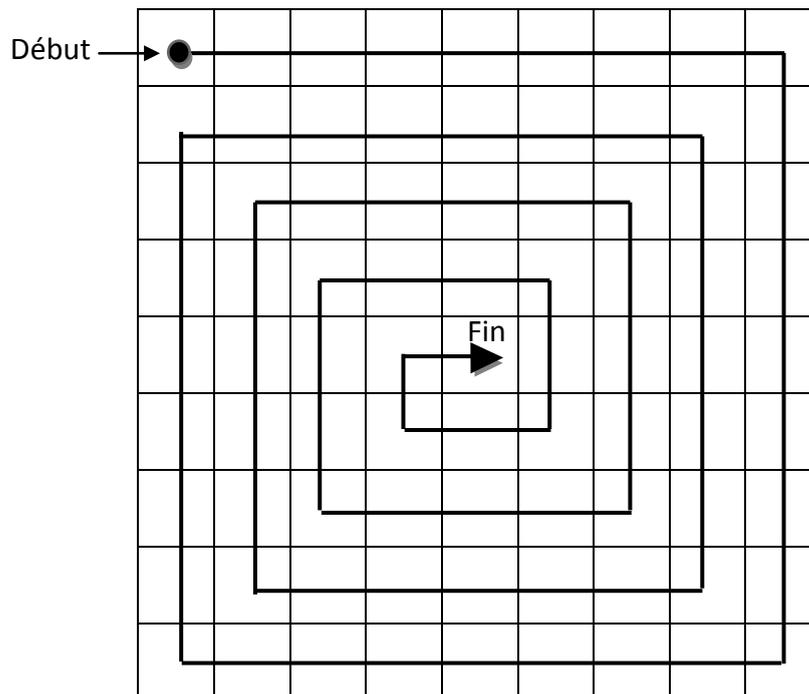


Figure III.8 : Le parcours d'une image quantifiée en spirale

Les quatre vecteurs sont codés ensuite arithmétiquement. On doit garder les dimensions des images avant le codage pour la lecture spirale inverse avant l'étape de décodage.

Etape 5 : La construction du fichier codé

Les vecteurs images codés arithmétiquement générés dans l'étape précédente sont utilisés pour former un seul vecteur.

Premièrement, et exactement comme pour la méthode sans pertes, les informations concernant la zone sombre éliminée dans l'étape 2 ainsi que les dimensions des images obtenues après la transformation sont ajoutées sur 15 octets tout au début.

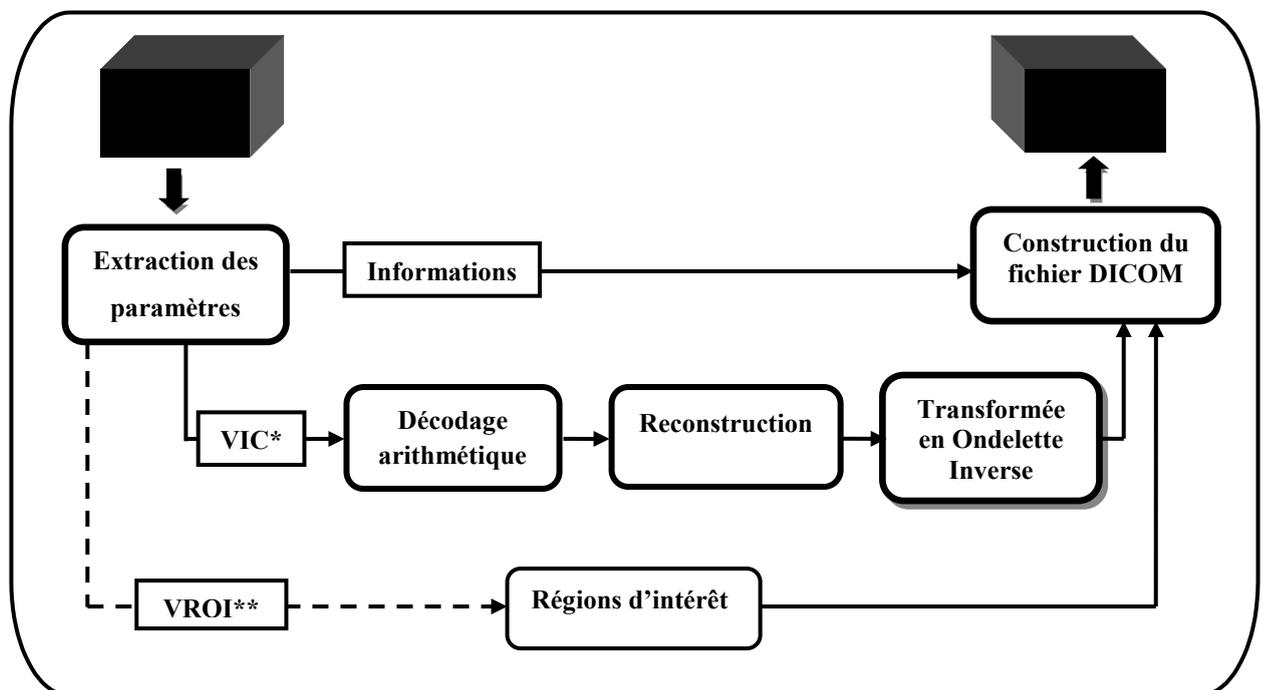
Ensuite, à partir du vecteur image codé transformé en matrice et les mêmes informations, un fichier DICOM est créé à l'aide de la commande Matlab « dicomwrite ».

Les matrices représentant les régions d'intérêt ainsi que leurs dimensions et leurs emplacements dans l'image initiale sont aussi insérées dans le fichier créé.

Après, les 128 premiers octets du fichier résultant sont éliminés, et sont remplacés par 6 octets de tatouage contenant le type de compression (avec pertes) et d'autres caractères spéciaux. Le fichier compressé est finalement généré avec une extension `.µdcm`.

III.3.2 La décompression

Le schéma général pour la décompression est donné par la figure suivante :



*Vecteur Image Codé ; **Vecteurs Régions OF Interest.

Figure III.9 : Schéma général de la décompression Avec perte

Etape 1 : Extraction des paramètres

Dans cette étape, le fichier compressé est authentifié en vérifiant le tatouage. Si le fichier provient vraiment de notre programme, la méthode avec la quelle il a été compressée sera identifiée.

En ce qui concerne les informations et le vecteur image codé, les opérations sont semblables à celles de l'étape d'extraction des paramètres de la décompression sans pertes. Ce qui diffère, c'est les régions d'intérêt si elles existent. Dans ce cas chacune d'elles est définie par un vecteur qui est transformé en matrice, et qui sera superposé avec l'image décompressée lors de la dernière étape.

Etape 2 : Décodage arithmétique

Le vecteur image codé est décodé arithmétiquement. Quatre vecteurs sont engendrés, chacun représente une bande de fréquences de l'image compressée (approximation, horizontale, verticale et diagonale).

Chaque vecteur est transformé en matrice, avec l'inverse de la méthode présentée dans l'étape 5 de la compression avec pertes. Cette méthode est présentée dans la figure suivante :

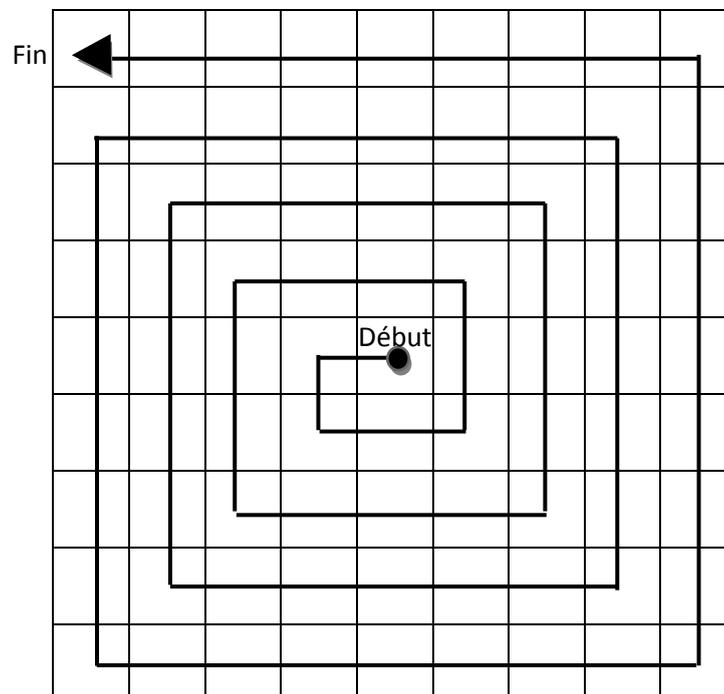


Figure III.10 : Réorganisation d'une image (vecteur) codée

Les matrices ainsi obtenues ne sont que les matrices indices qui représentent les bandes de fréquence (obtenues par la transformée en ondelette) quantifiées.

Etape 3 : La reconstruction

Pour la quantification scalaire et vectorielle, les mêmes dictionnaires sont utilisés. Chaque indice est remplacé par la valeur qui lui correspond dans le cas de la quantification scalaire, et par le bloc qui lui correspond dans le cas de la quantification vectorielle.

Les bandes approximation, horizontale, verticale et diagonale sont à présent établies.

Etape 4 : La transformée en ondelette inverse

On applique une transformée en ondelette inverse sur les quatre bandes obtenues dans l'étape précédente.

Pour cela, on utilise la commande Matlab « idwt2 », en précisant en entrée les filtres de reconstruction de Le Gall dont les coefficients sont donnés dans le tableau suivant :

Filtres de reconstruction		
i	Passe-bas	Passe-haut
0	1	6/8
±1	1/2	-2/8
±2		-1/8

Tableau III.2 : Coefficients des filtres Le Gall 5/3 de reconstruction [19]

Le mode d'extension de la transformée en ondelettes doit être aussi mis au mode « periodization ».

Etape 5 : La reconstruction du fichier DICOM

La zone sombre est ajoutée en premier lieu, puis les régions d'intérêt, si elles existent, sont superposées à l'image obtenue.

Le fichier DICOM est reconstruit à partir de la dernière image et les informations récupérées dans l'étape 1, à l'aide de la commande Matlab « dicomwrite », et a comme extension .dcm.

III.4 Conclusion

Dans ce chapitre, nous avons proposé deux techniques de compression des images DICOM. La première est sans pertes, basée sur le codage RLE de l'image. La deuxième est avec pertes, et qui après la transformée en ondelette, effectue une quantification scalaire pour l'approximation et vectorielle pour le détail pour que les matrices indices subissent un codage arithmétique. Les régions d'intérêt permettent de garder une qualité optimale pour les zones critiques d'un point de vue diagnostique.

Dans les deux cas, les informations propres au fichier DICOM ne suivent pas le même chemin que l'image car on ne tolère pas la moindre perte.

Les résultats de chaque technique seront présentés et discutés dans le chapitre suivant.

Chapitre IV

**PRESENTATION
ET INTERPRETATION
DES RESULTATS**

IV.1 Introduction :

Dans ce chapitre, nous allons donner les résultats que nous avons obtenus pour les deux méthodes (*avec pertes* et *sans pertes*) que nous avons utilisé. Nous utiliserons les critères d'évaluation définis dans le paragraphe I.2.3 à savoir : Le taux de compression, le rapport de compression, le PSNR et le SSIM. Nous donnerons aussi les temps de compression et de décompression, ainsi que les temps qui ont été nécessaires à la création des dictionnaires.

Pour que le lecteur puisse apprécier visuellement la qualité des images reconstruites lors des différents tests, les images originales et reconstruites seront aussi montrées.

Les tests été réalisés sur un ordinateur ayant les caractéristiques suivantes :

- Processeur Intel Core 2 Duo E6700 @2.66GHz avec 4 Mo de mémoire cache.
- Mémoire vive de 1.5Go @533MHz de type DDR2.
- Disque dur de 16Mo de mémoire cache.

Après les résultats, nous présenterons l'interface graphique « DICOM QUANTIZIER » que nous avons réalisée.

IV.2 Compression sans pertes :

Dans ce tableau, nous allons présenter les résultats obtenus dans ce mode de compression.

Type d'Image	Taille du fichier original	Taille du fichier compressé	Taux de compression	Rapport de compression
IRM de cerveau	130 Ko	84 Ko	35.74%	1.54 :1
IRM de cerveau	130 Ko	85 Ko	34.99 %	1.52 :1
IRM Abdomen	514 Ko	193 Ko	62.45%	2.66 :1
IRM Abdomen	514 Ko	162 Ko	68.64%	3.17 :1
	Temps de compression	Temps de décompression	PSNR	SSIM
	~2 secondes	Instantanément	44.82 dB	96.34%
	~2 secondes	Instantanément	46.86 dB	98.89%
	26 secondes	~3 secondes	51.94 dB	97.63%
	25 secondes	~2 secondes	51.02 dB	98.24%

Tableau IV.1 : Résultats de la compression sans pertes

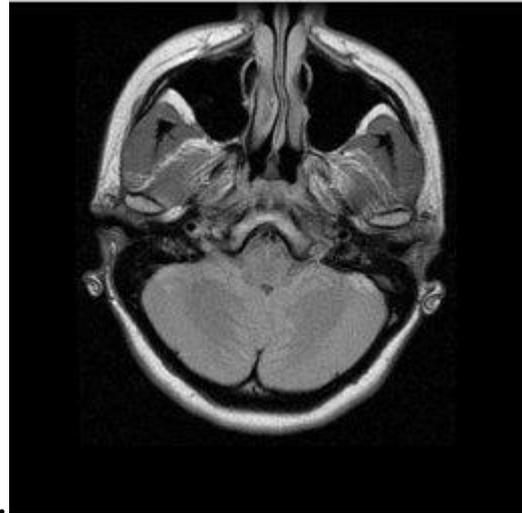
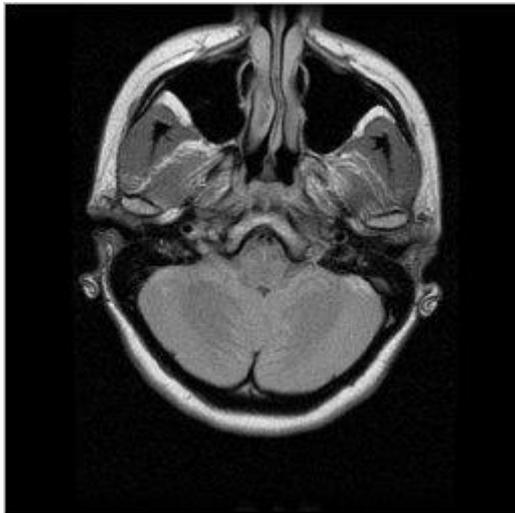
On remarque à travers ce test que :

- Le rapport de compression est inférieur à 4 :1 : C'est là l'inconvénient majeur des méthodes de compression sans pertes.

- Le temps de compression est plus important que le temps de décompression : Lors de la compression, l'algorithme RLE parcourt tout le fichier à coder (pixel par pixel) ce qui représente un temps non négligeable si la taille de l'image augmente (images 3 et 4) alors que lors de la décompression, il y a juste construction d'un vecteur ce qui est relativement rapide.

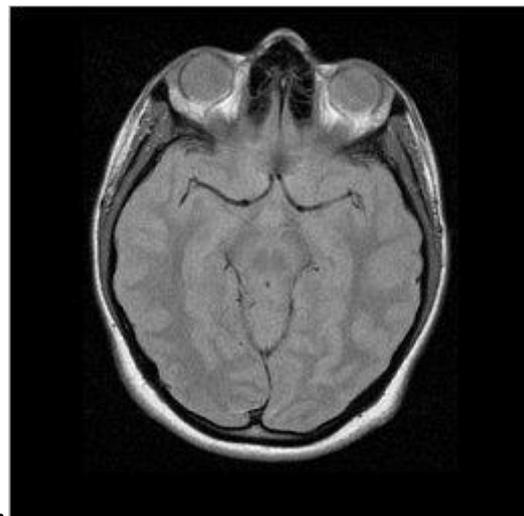
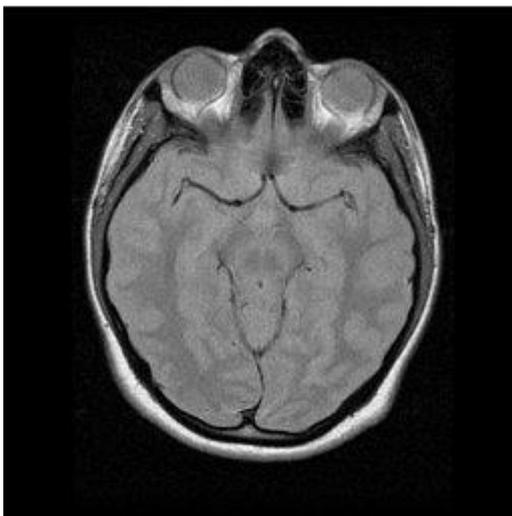
- Plus la taille du fichier augmente, plus le taux de compression augmente : Une image plus grande contient plus de redondances.

- Le PSNR est supérieur à 40dB et le SSIM est proche de 1 : Théoriquement, la valeur du PSNR tend vers l'infini ($MSE = 0$) et celle du SSIM est égale à 1 pour une compression sans pertes



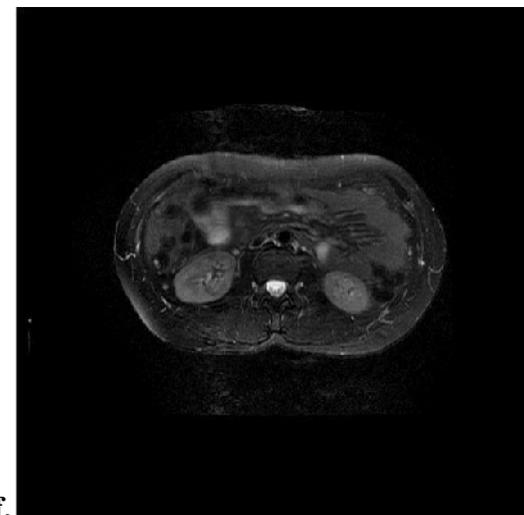
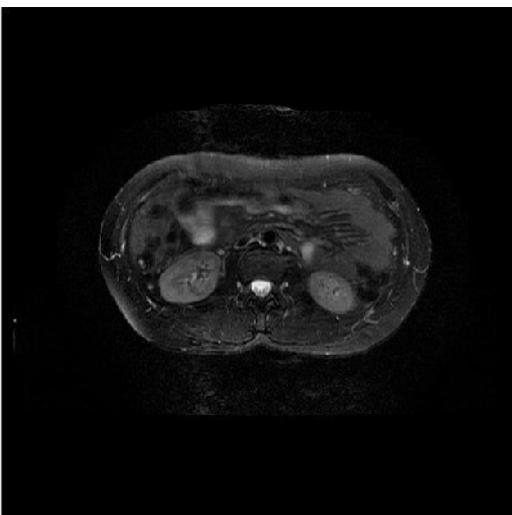
a.

b.



c.

d.



e.

f.

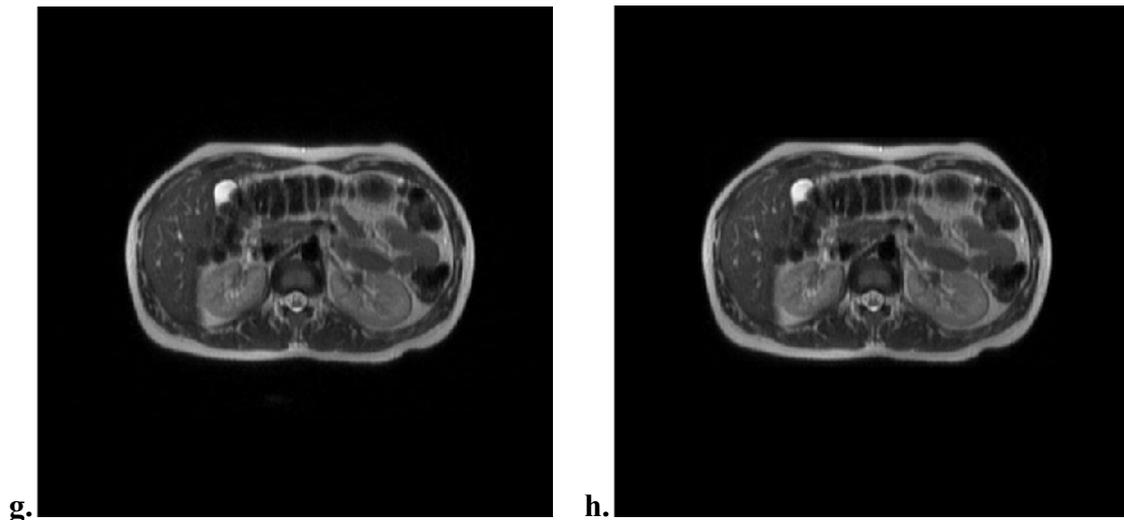


Figure IV.1 : Compression des images avec la méthode sans pertes:

(a, c, e, g) Images originales, (b, d, f, h) Images reconstruites.

On voit bien à travers cette figure que les images originales et les images reconstruites sont presque les mêmes. Ceci est justifié par les valeurs du PSNR et du SSIM qui sont assez importantes, preuve que la méthode est quasiment sans pertes. Une légère distorsion a été introduite lors de l'élimination de la zone sombre (qui ne contient pas d'information utile).

IV.3 Quantification scalaire :

Dans ce test, on applique une quantification scalaire directement sur l'image.

Nous avons utilisé un dictionnaire scalaire (figure IV.2) obtenu à partir d'une carte de Kohonen 10x10.

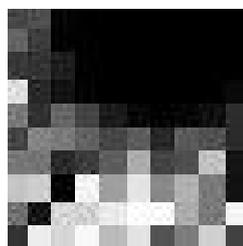


Figure IV.2 : Dictionnaire utilisé pour la quantification scalaire.

Il a fallu 10 heures 11 minutes et 11 secondes pour le calcul de ce dictionnaire.

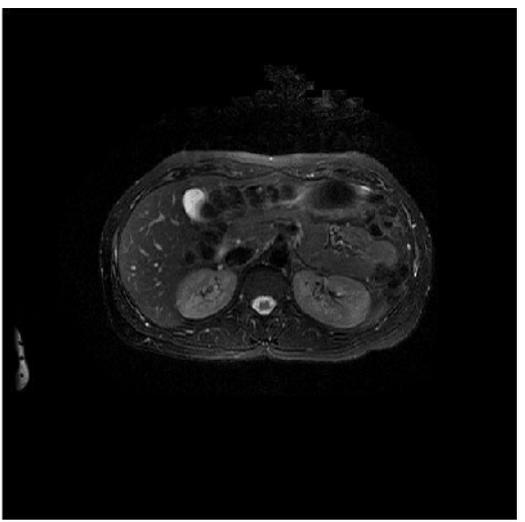
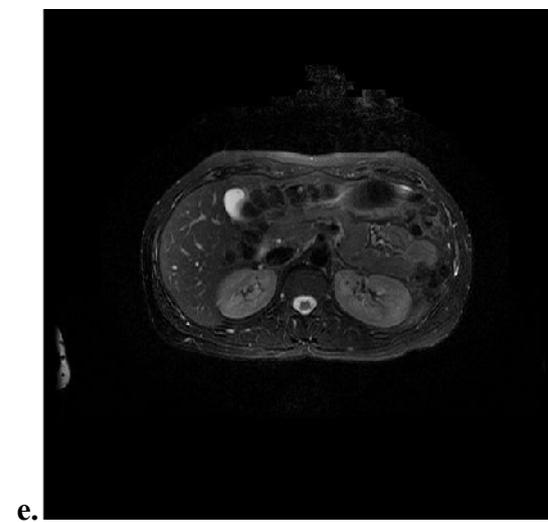
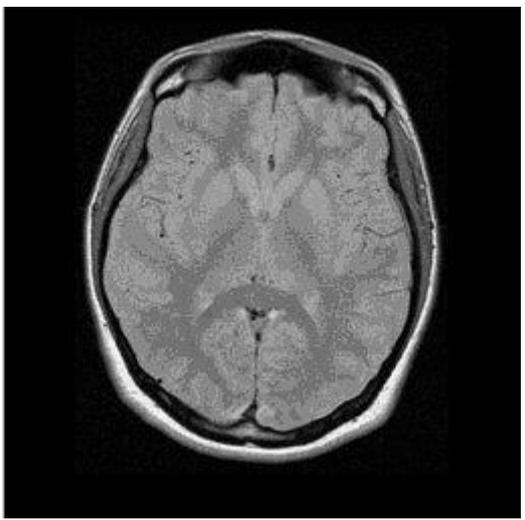
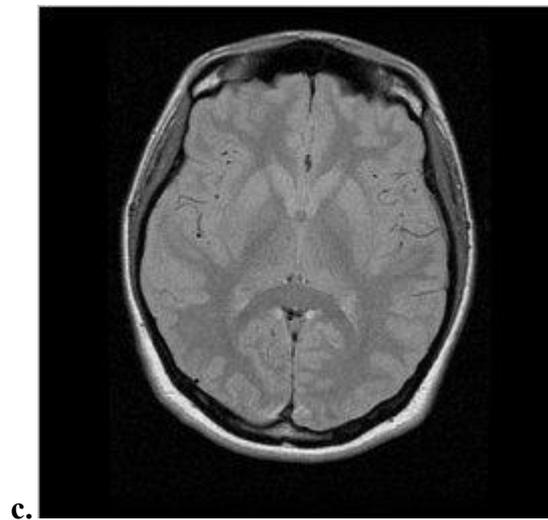
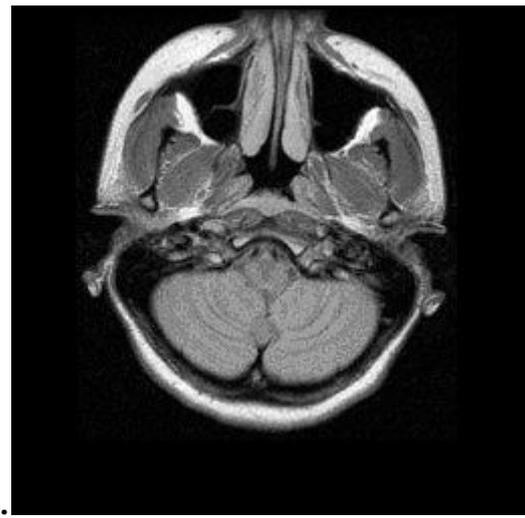
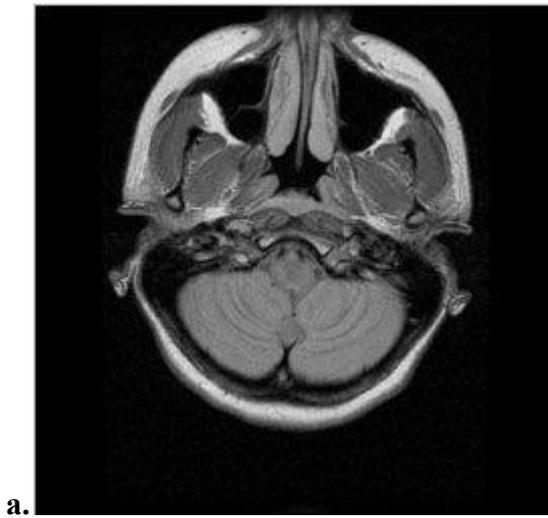
Le tableau suivant présente les résultats de cette méthode.

Type d'Image	Taille du fichier original	Taille du fichier compressé	Taux de compression	Rapport de compression
IRM de cerveau	130 Ko	29 Ko	78.41 %	4.48 :1
IRM de cerveau	130 Ko	25 Ko	81.09 %	5.20 :1
IRM Abdomen	514 Ko	77 Ko	85.46 %	6.67 :1
IRM Abdomen	514 Ko	52 Ko	90.10 %	9.88 :1
	Temps de compression	Temps de décompression	PSNR	SSIM
	~18 secondes	~3 secondes	26.03 dB	87.84 %
	~19 secondes	~3 secondes	25.52 dB	86.31 %
	~57 secondes	~5 secondes	49.94 dB	98.05 %
	~34 secondes	~5 secondes	34.89 dB	96.46 %

Tableau IV.2 : Résultats de la quantification scalaire.

On remarque à travers ce test que :

- Le rapport de compression a dépassé 4 :1.
- Le temps de compression et décompression de cette méthode sont nettement plus grands de ceux de la méthode sans pertes
- Plus la taille du fichier augmente, plus le taux de compression augmente.
- Les valeurs du PSNR et du SSIM sont assez bonnes, ceci peut être vu à travers la figure suivante où nous constatons une légère modification des images reconstruites, qui peut être assimilée à une modification de la valeur moyenne de la luminosité.



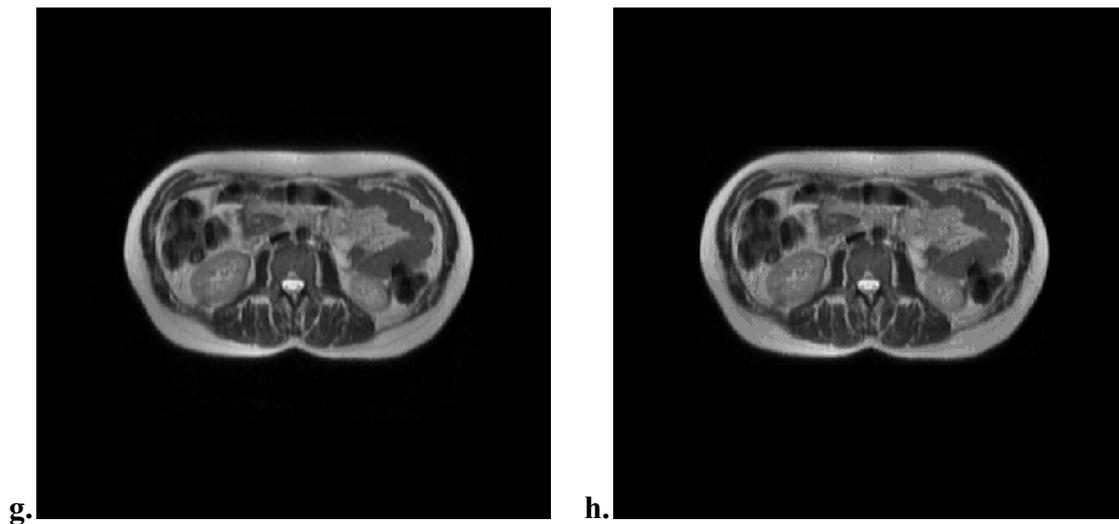


Figure IV.3 : Quantification scalaire des images :

(a, c, e, g) Images originales ; (b, d, f, h) Images reconstruites.

IV.4 Quantification vectorielle :

Dans ce test, on applique une quantification vectorielle directement sur l'image.

Nous avons utilisé un dictionnaire vectoriel (figure IV.4) obtenu à partir d'une carte de Kohonen 10x15. La taille des vecteurs utilisés lors de la quantification vectorielle est de 4x4.

Il a fallu 1 heure 29 minutes et 42 secondes pour le calcul de ce dictionnaire.

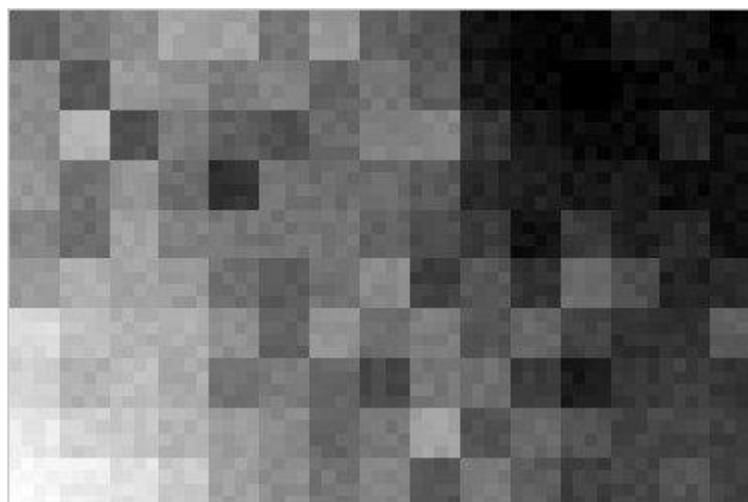


Figure IV.4 : Dictionnaire utilisé pour la quantification vectorielle

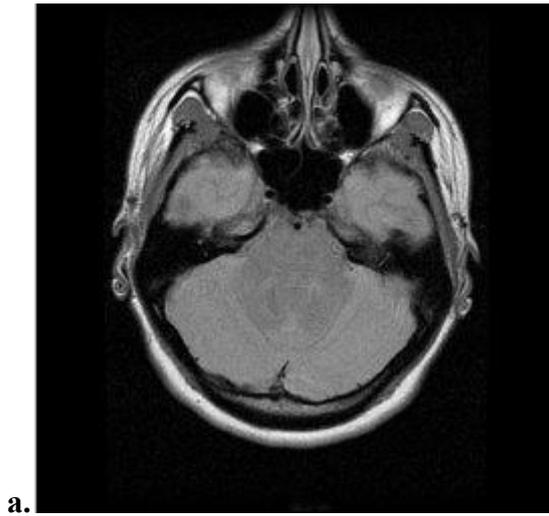
Le tableau suivant présente les résultats de cette méthode :

Type d'Image	Taille du fichier original	Taille du fichier compressé	Taux de compression	Rapport de compression
IRM de cerveau	130 Ko	4 Ko	96.92 %	32.5 :1
IRM de cerveau	130 Ko	3.5 Ko	97.30 %	37.14 :1
IRM Abdomen	514 Ko	6.5 Ko	98.73 %	79.07 :1
IRM Abdomen	514 Ko	6 Ko	98.83 %	85.67 :1
	Temps de compression	Temps de décompression	PSNR	SSIM
	~8 secondes	Instantanément	23.66 dB	86.27 %
	~4 secondes	Instantanément	25.93 dB	85.62 %
	~19 secondes	~2 secondes	36.18 dB	91.04 %
	~9 secondes	Instantanément	31.85 dB	95.31 %

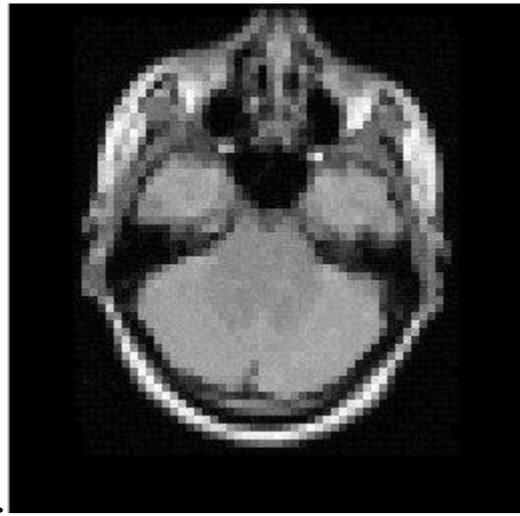
Tableau IV.3 : Résultats de la quantification vectorielle

On remarque à travers ce test que :

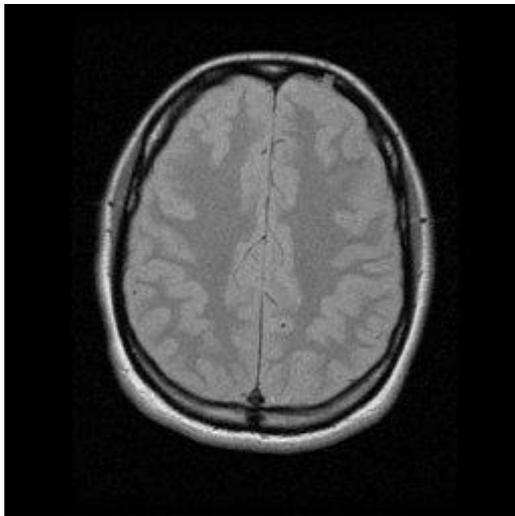
- Le rapport de compression est énorme.
- Les temps de compression et décompression de cette méthode sont nettement plus inférieurs que ceux de la quantification scalaire.
- Plus la taille du fichier augmente, plus le taux de compression augmente.
- La qualité des images varie entre médiocre et mauvaise, en fonction de sa complexité, comme on peut le constater dans la figure suivante :



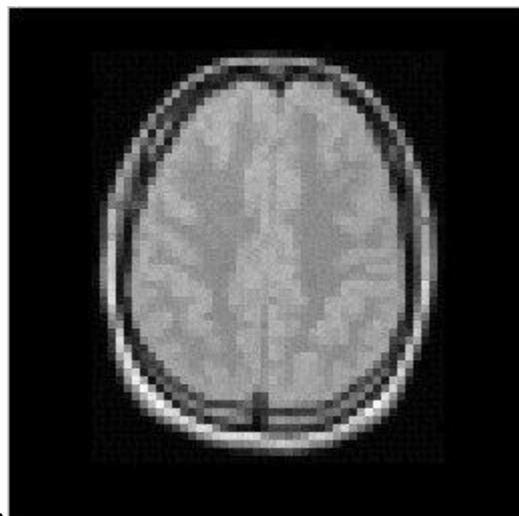
a.



b.



c.



d.



e.



f.

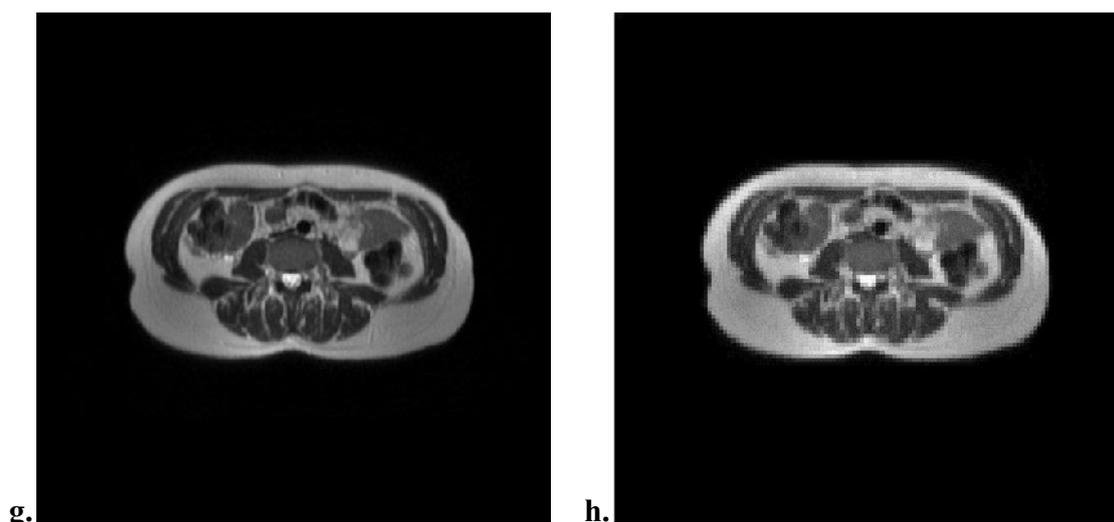


Figure IV.5 : Quantification vectorielle des images :

(a, c, e, g) Images originales ; (b, d, f, h) Images reconstruites.

IV.5 Compression avec pertes :

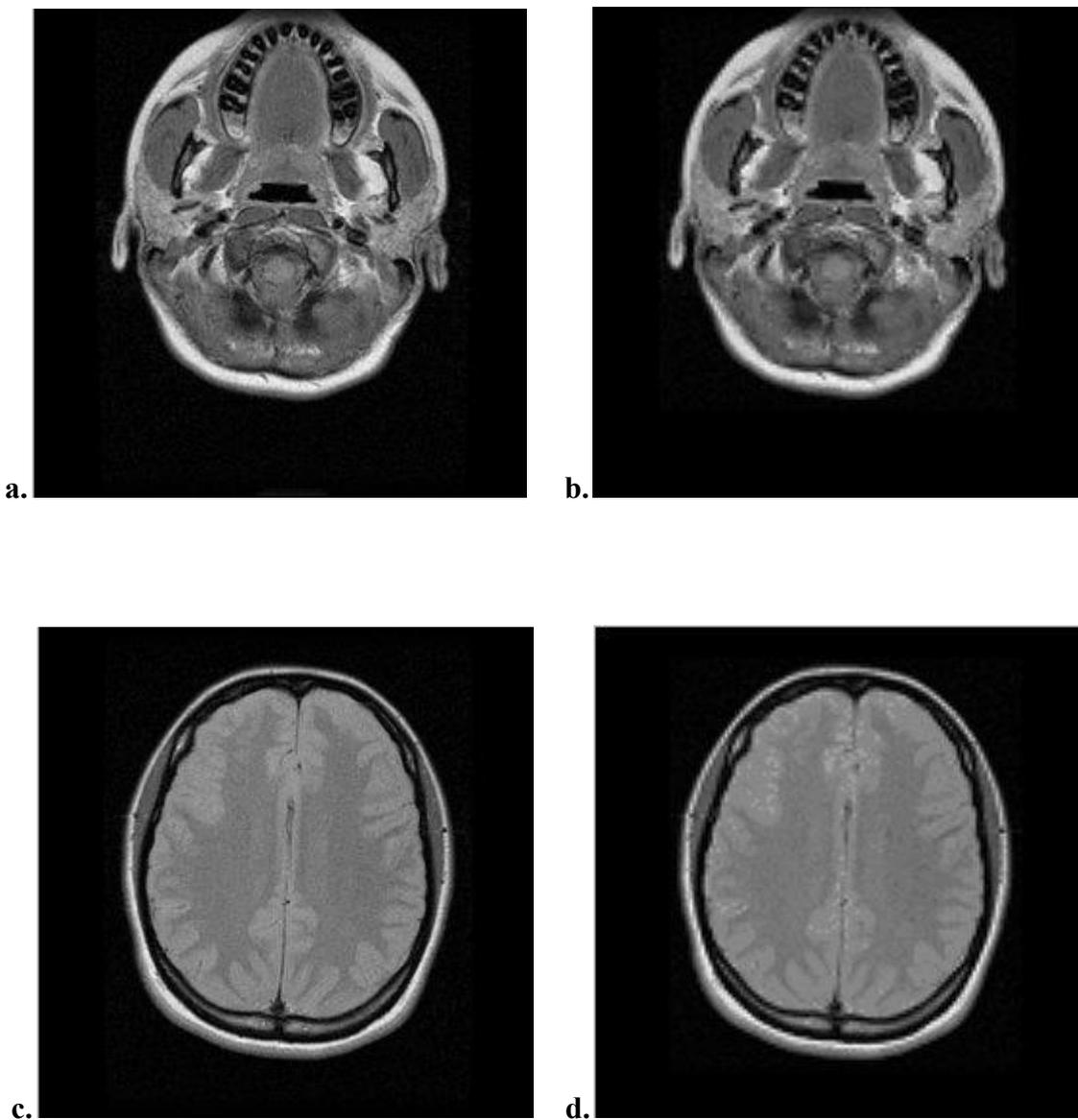
Le dictionnaire scalaire utilisé est obtenu à partir d'une carte de Kohonen 10x10, alors que le vectoriel est obtenu à partir d'une carte 7x7. La taille des vecteurs utilisés lors de la quantification vectorielle est de 4x4.

Type d'Image	Taille du fichier original	Taille du fichier compressé	Taux de compression	Rapport de compression
IRM de cerveau	130 Ko	9 Ko	93.08%	14.44 :1
IRM de cerveau	130 Ko	7 Ko	94.62%	18.57 :1
IRM Abdomen	514 Ko	20 Ko	96.11%	25.7 :1
IRM Abdomen	514 Ko	15 Ko	97.08%	34.26 :1
	Temps de compression	Temps de décompression	PSNR	SSIM
	~6 secondes	~2 secondes	27.48 dB	85.38%
	~6 secondes	~2 secondes	33.73 dB	94.48%
	~19 secondes	~6 secondes	39.96dB	96.07%
	~11 secondes	~4 secondes	44.22 dB	97.09%

Table IV.4 : Résultats de la compression avec pertes.

On remarque à travers ce test que :

- Le rapport de compression est important : Il se situe entre les résultats des deux précédents.
- Le temps de compression est du même ordre que celui de la compression vectorielle, par contre le temps de décompression est un peu plus important, cela étant dû à l'opération supplémentaire de l'ondelettes inverse.
- Les valeurs du PSNR et du SSIM sont assez importantes : La dégradation n'est pas très significative, les valeurs du PSNR et du SSIM le prouvent, cela est dû à l'utilisation de la quantification scalaire.



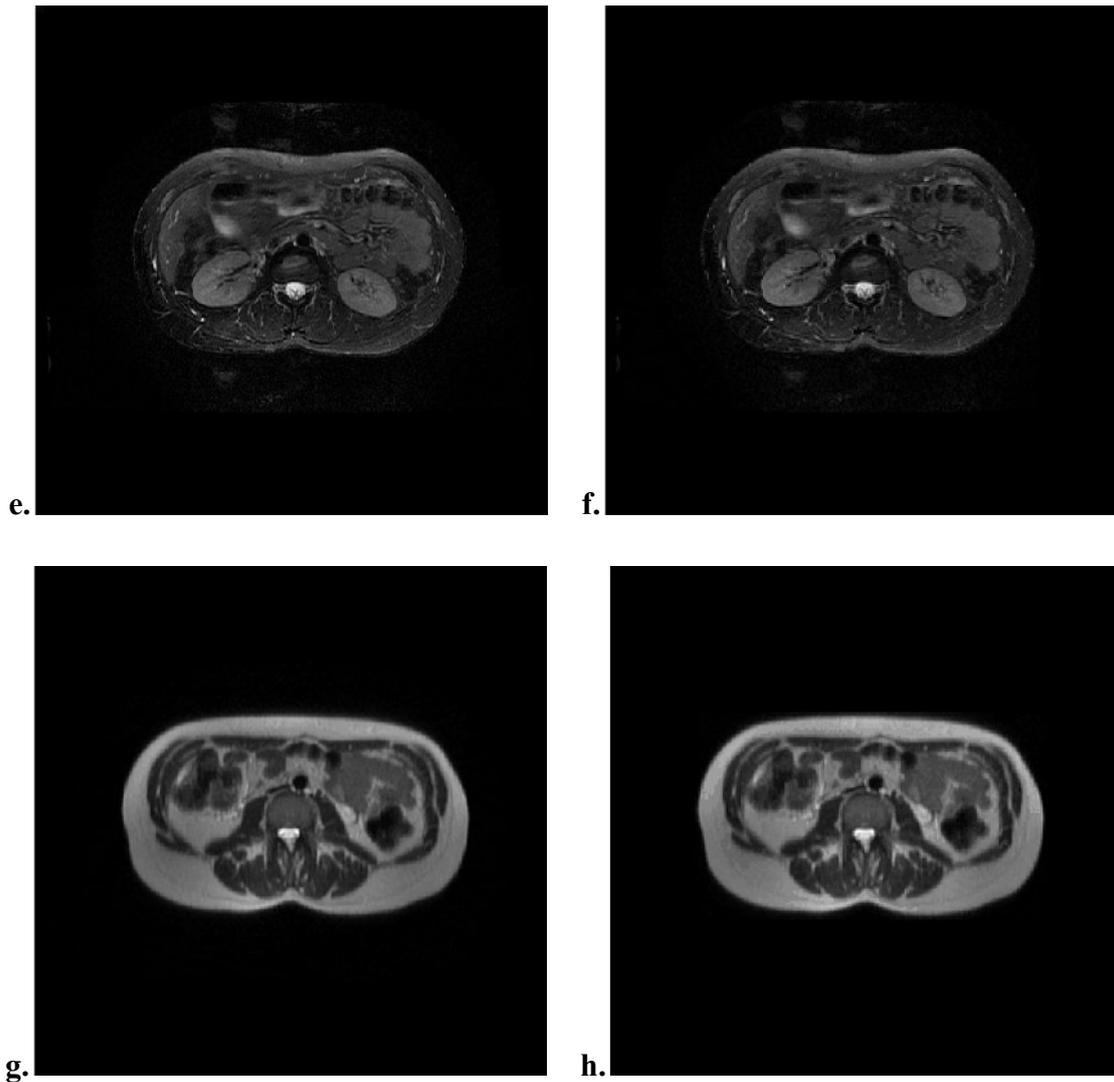


Figure IV.6 : Compression avec pertes :

(a, c, e, g) Images originales ; (b, d, f, h) Images reconstruites.

On voit à travers cette figure qu'il existe quelques différences entre les images originales et les images reconstruites. Cela était bien sur prévisible étant donné la méthode utilisée.

IV.6 L'interface graphique DICOM QUANTIZIER

L'interface qu'on a réalisée est très ergonomique. Toutes les options relatives à la compression et la décompression y sont présentes, elles sont affichées d'une manière séquentielle, comme sera expliqué ultérieurement. L'interface se présente comme suit :

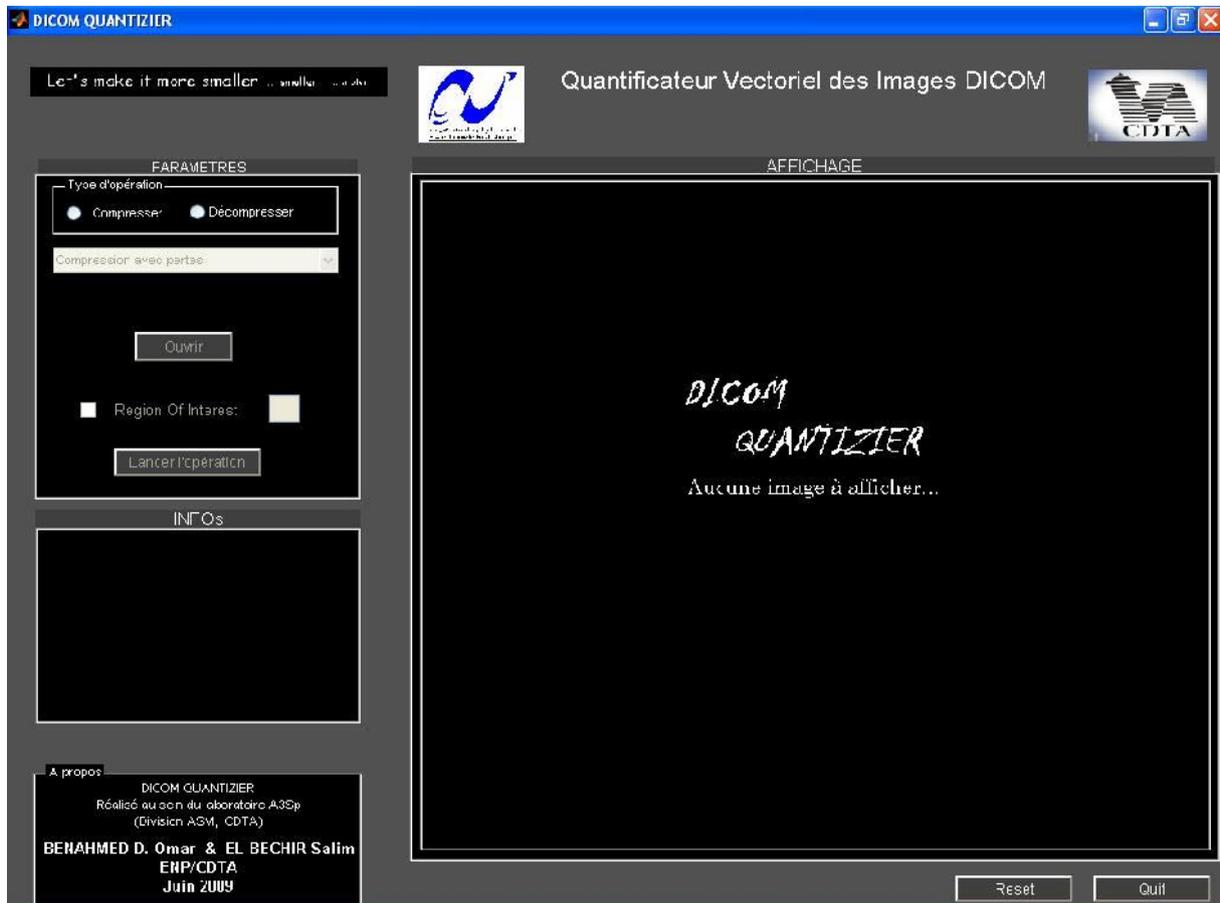


Figure IV-7 : L'interface graphique DICOM QUNTIZIER

Dans la première zone active, Type d'opération, on trouve les deux choix : compresser et décompresser. Un seul choix peut être sélectionné. En fonction de ce choix, d'autres zones seront activées ou désactivées.

Si le choix « compresser » est validé, on peut choisir juste en dessous, 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, l'utilisateur peut cocher la case blanche et entrer le nombre des régions d'intérêt.

Lorsque la décompression est choisie, seul le bouton « Ouvrir » est activé.

Le bouton « Ouvrir » est utilisé pour charger le fichier à traiter. Pour la compression, seuls les fichiers avec l'extension **.dcm** sont affichés. Par contre, seuls les fichiers avec l'extension **.jdicm** sont affichés lors d'une décompression.

Une fois chargé, le nom de fichier ainsi que sa taille sont affichés au dessus de la zone image. L'image entourée des informations textuelles est affichée au début d'une compression ou à la fin d'une décompression.

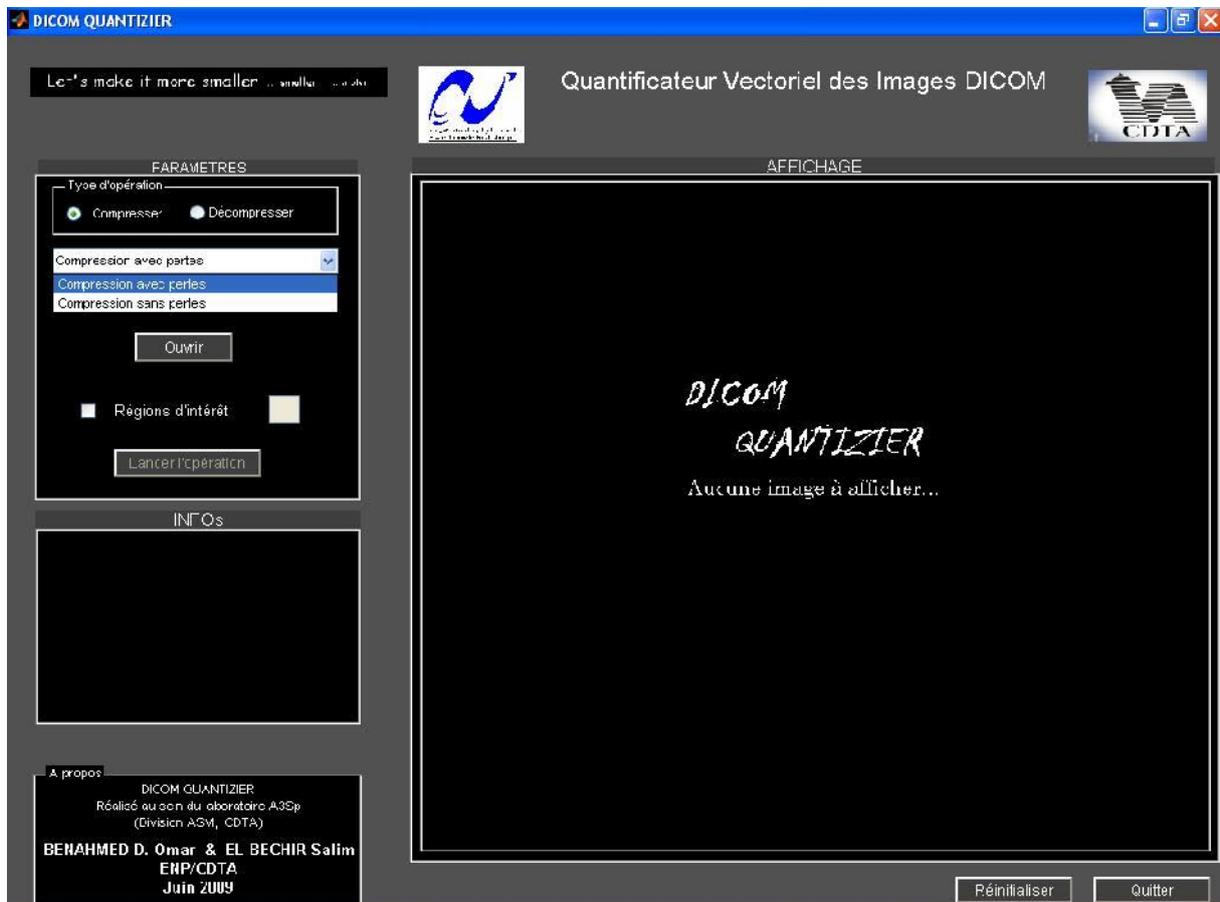
Pour exécuter le process, le bouton « Lancer l'opération » doit être appuyé. Lors de l'exécution, un message est affiché, indiquant que l'opération est en cours. A la fin, les résultats de l'opération sont affichés (en bas, à gauche).

Le bouton « Réinitialiser » réinitialise l'interface.

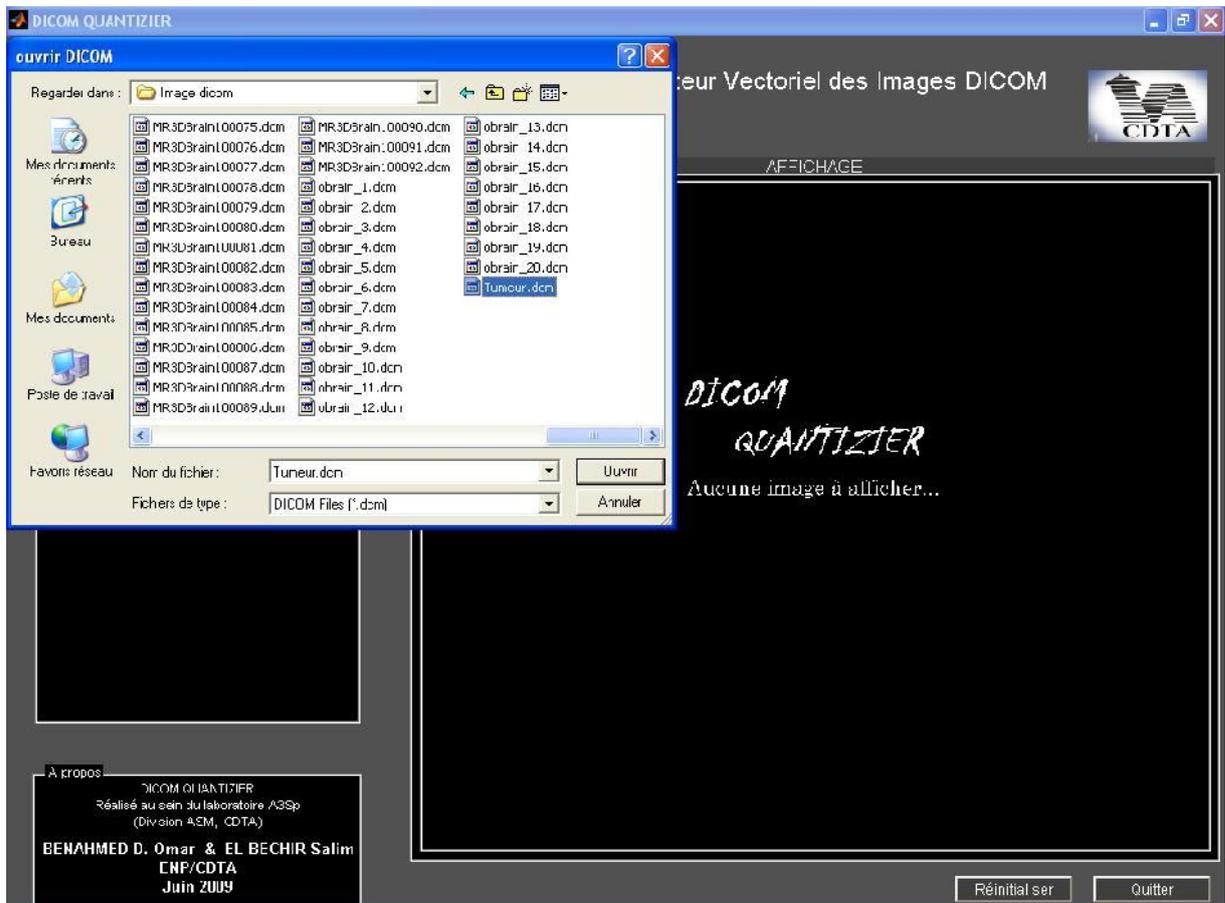
Le bouton « Quitter » ferme l'interface.

IV.7 Exemple de déroulement d'une opération :

IV.7.1 La compression :

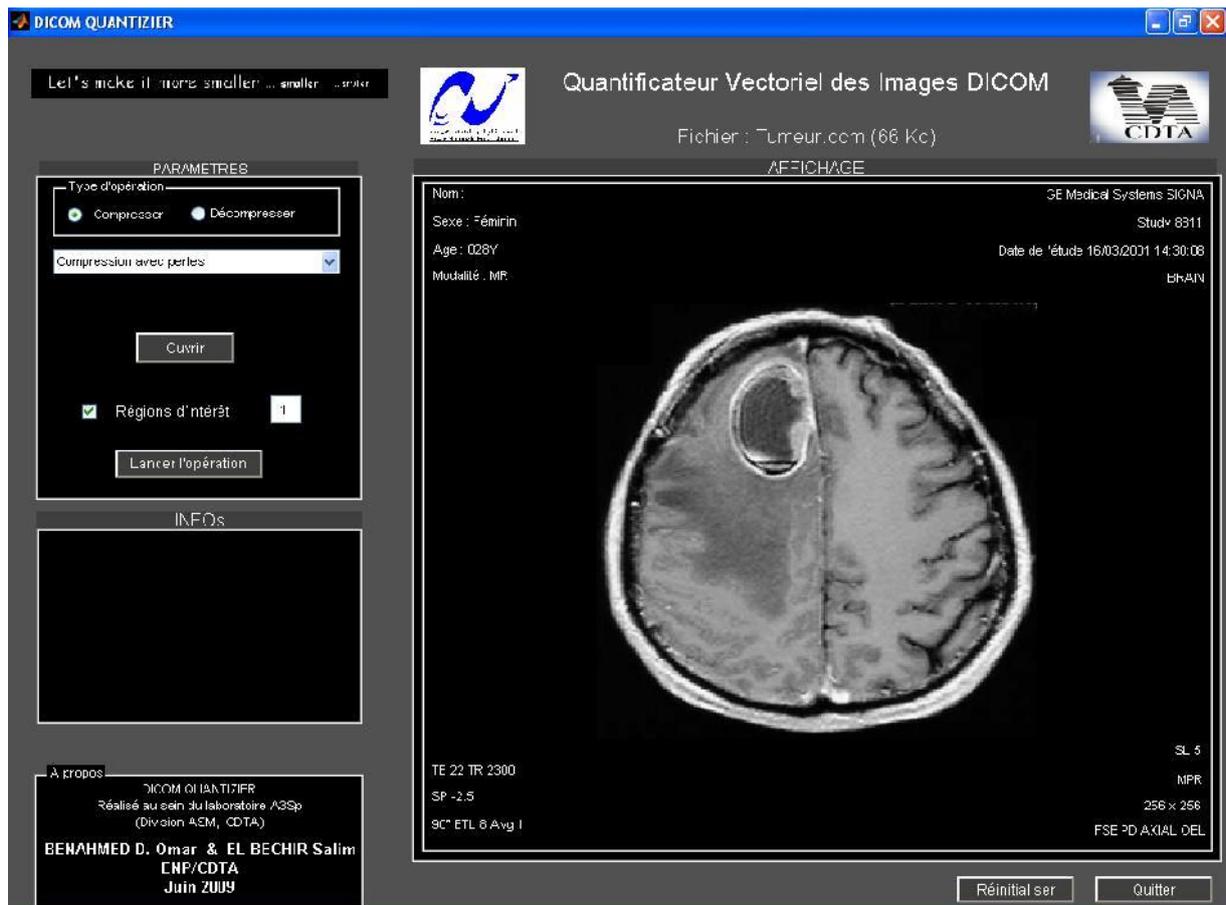


Le bouton « Compresser » est activé en mode « compression avec pertes »



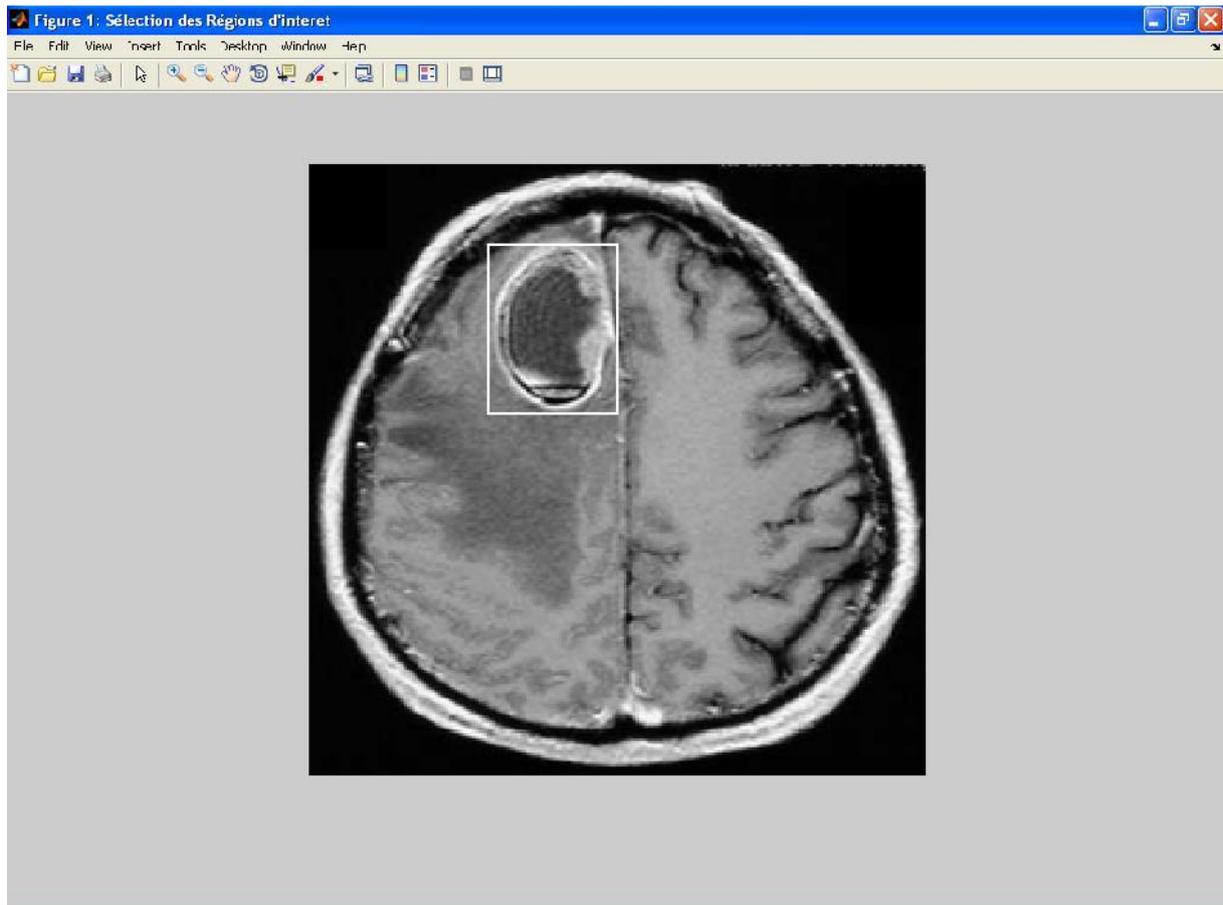
Après avoir appuyé sur le bouton « ouvrir », le fichier à compresser doit être sélectionné.

L'image et les informations seront affichées



Si l'option « Région d'intérêt » est activée, le nombre des régions d'intérêt doit être précisé. Dans la figure précédente, nous avons pris comme exemple une région (1).

Par la suite, la région d'intérêt est sélectionnée. Dans notre exemple, la région représente une tumeur, comme le montre la figure suivante :

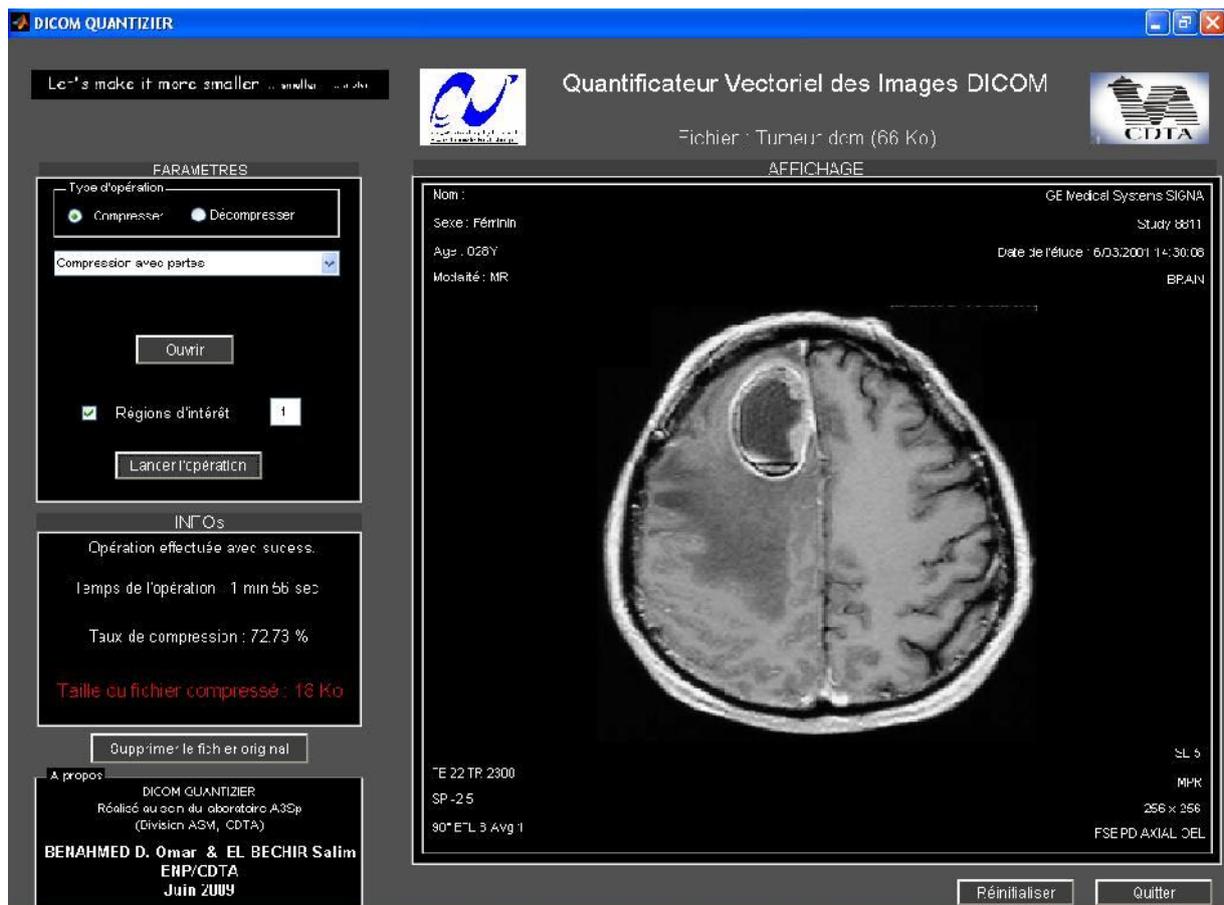


L'opération est lancée après avoir appuyé sur le bouton « Lancer l'opération ». Ce dernier doit être appuyé une seul fois.

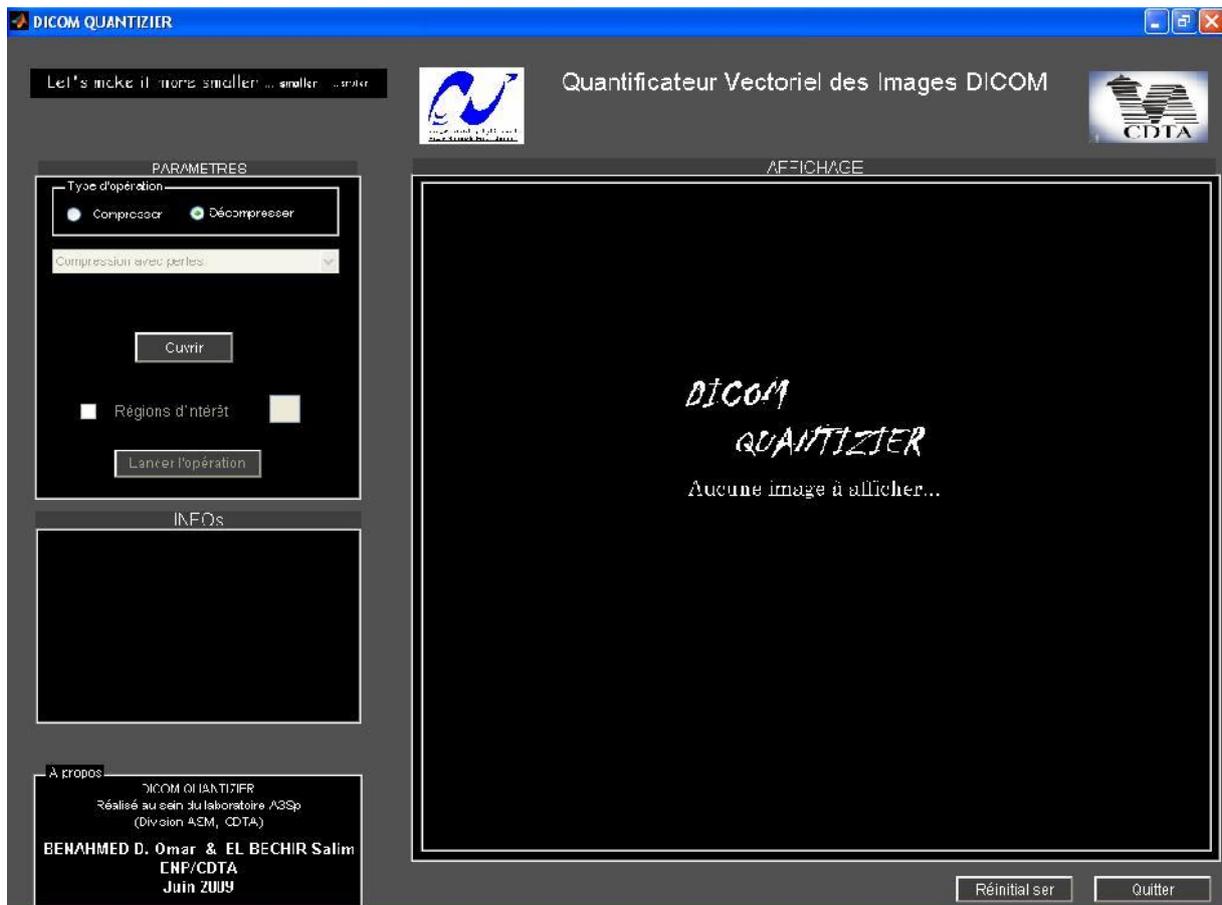
L'opération peut prendre plus d'une minute, en fonction de la taille et de la complexité de l'image ainsi que la puissance de la machine utilisée.

Un fichier .udcm est généré. C'est le fichier compressé.

Finalement, les résultats de l'opération sont affichés comme le montre la figure suivante :

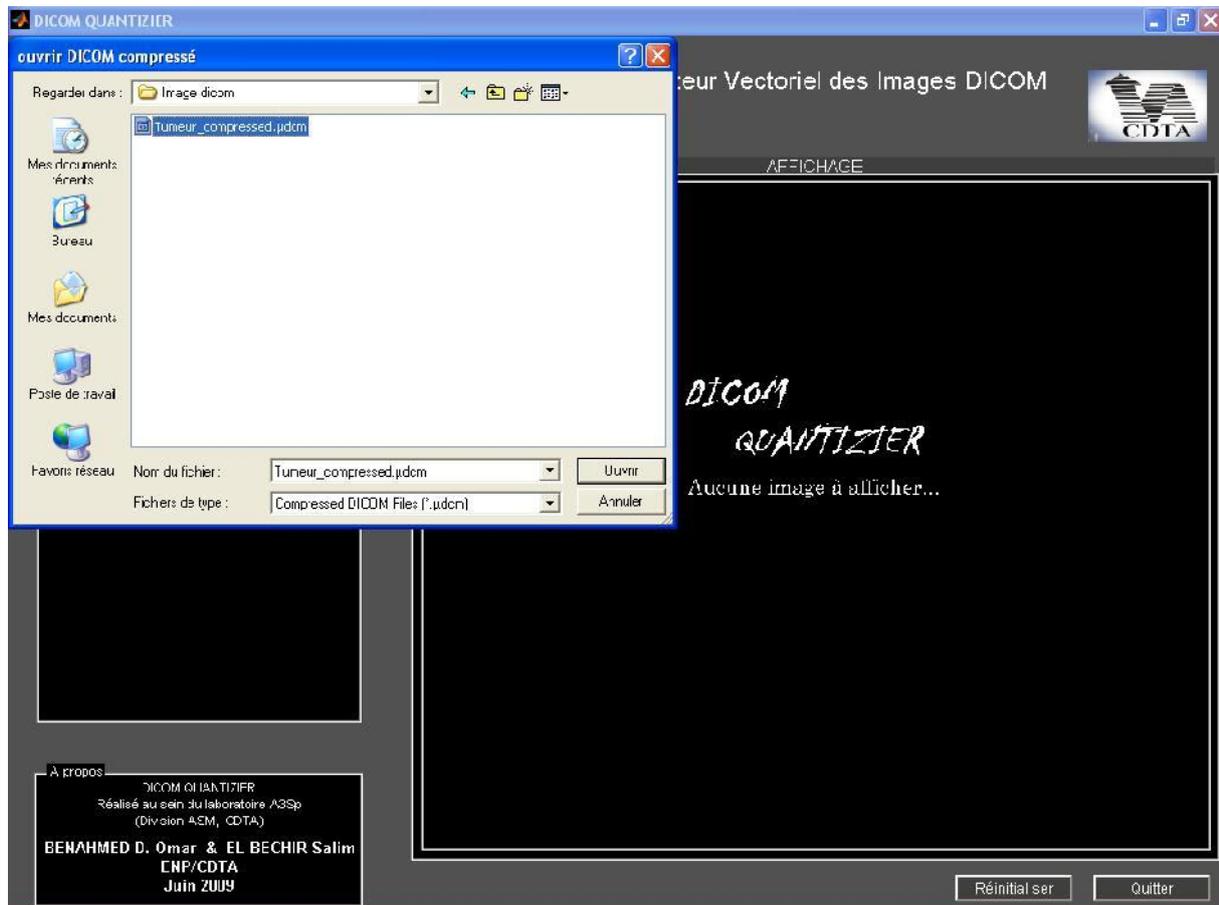


IV.2 La décompression :



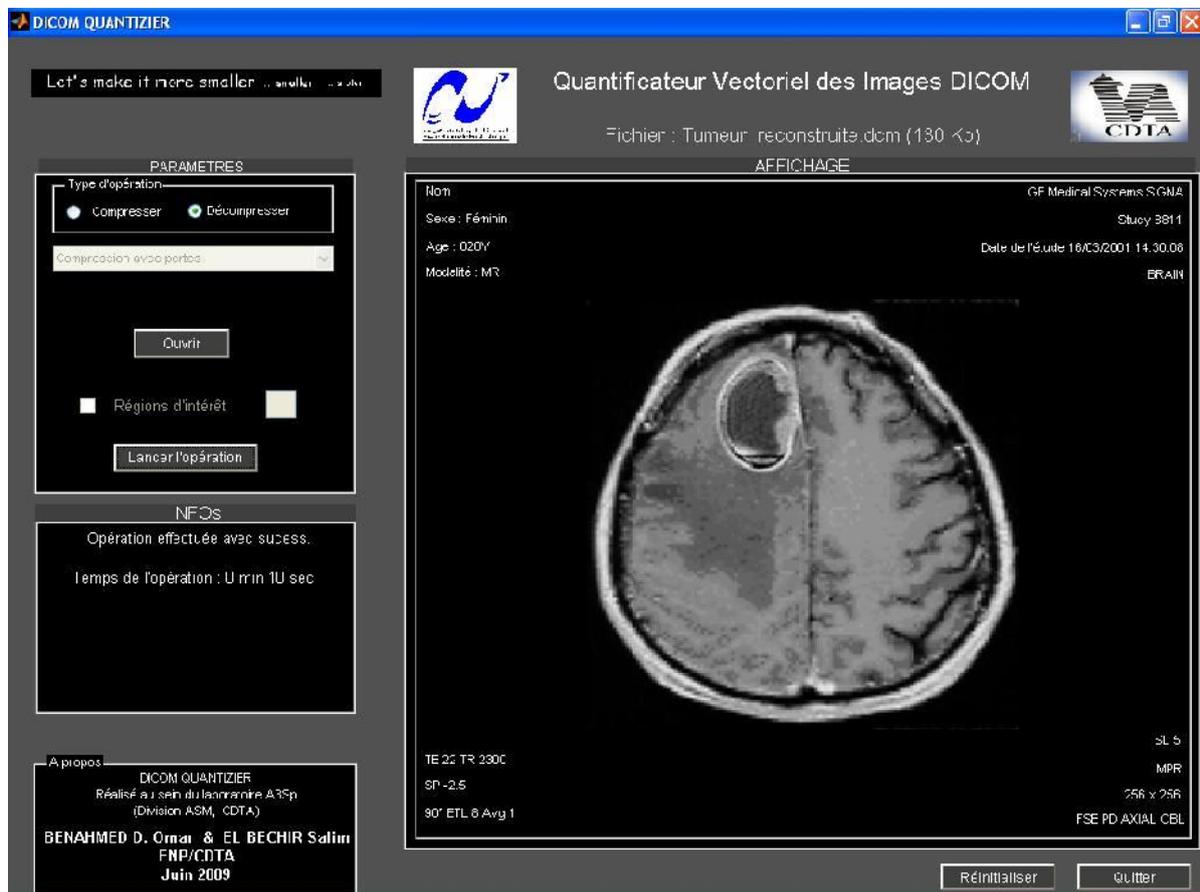
Le bouton « Décompresser » est activé. Un fichier .μdcm doit être chargé.

Comme le montre la figure suivante, seuls les fichiers avec une extension .udcm sont affichés.



L'opération de décompression prend beaucoup moins de temps que celle de compression.

A la fin de l'opération, l'image décompressée et les informations textuelles ainsi que les résultats (temps de décompression) sont affichés.



Pour la compression sans pertes, le déroulement des étapes est le même, sauf que l'option « Régions d'intérêt » n'est pas activée.

IV.8 Conclusion

Dans ce chapitre, nous avons présenté les principaux tests que nous avons effectués lors de l'élaboration des méthodes de compression.

La méthode sans pertes présente une reconstruction quasi-parfaite, mais apporte des faibles rapports de compression ; elle demeure utilisable.

La quantification scalaire donne des images reconstruites de très bonne qualité avec des rapports de compression moyens.

Les images reconstruites après quantification vectorielle sont d'une qualité très inférieure à celle des originales.

Notre deuxième méthode (compression avec pertes) essaye d'équilibrer les caractéristiques des deux types de quantification, à l'aide de la transformée en ondelettes qui nous permet de séparer les composantes basses fréquences des composantes hautes fréquences, ensuite effectuer une quantification scalaire sur les premières pour préserver la qualité et une quantification vectorielle sur les secondes pour améliorer le taux de compression avec une légère dégradation de la qualité.

L'interface ergonomique que nous avons réalisée est présentée dans ce chapitre, elle permet de rendre les opérations de compression et de décompression d'une grande simplicité.

CONCLUSION GENERALE

Dans le cadre de ce projet, nous avons implémenté sous Matlab deux algorithmes pour la compression des fichiers issus de la norme DICOM.

Dans le premier cas, l'algorithme est sans pertes basé sur la méthode RLE qui, vu sa simplicité, offre des performances appréciables, néanmoins insuffisantes pour remédier efficacement aux problèmes de stockage et de temps de transmission rencontrés dans le domaine médical.

Le second algorithme, avec pertes, est bien plus performant. Nous avons d'abord effectué un prétraitement, à l'aide de la transformée en ondelettes, qui nous a permis de mettre au point un quantificateur hybride profitant de la qualité offerte par la quantification scalaire et des performances offertes par la quantification vectorielle.

Les dictionnaires, scalaire et vectoriel, qui sont les éléments clés de notre chaîne de compression ont été obtenus à partir du réseau neuronal de Kohonen qui possède des propriétés bien adaptées à la compression des images.

A travers ces deux méthodes, 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 très bonnes performances tout en fournissant des images reconstruites d'une qualité bien plus qu'appréciable.

Nous avons mis la lumière sur l'importance des normes dans le domaine des systèmes d'information des établissements médicaux. Dans ce sens, nous avons vu que la norme DICOM a apporté des améliorations considérables, si bien qu'elle est aujourd'hui adoptée par les plus grands constructeurs.

Le développement de ce projet nous permis d'apprécier, d'une part, l'importance de la compression des données pour l'optimisation des systèmes d'archivage et de communication et, d'autre part, les améliorations considérables apportées par une norme dans un domaine donné à travers l'étude d'un cas réel.

Nous estimons avoir apporté une contribution au domaine de la compression des images médicales et espérons, qu'à l'avenir, des améliorations y seront apportées notamment :

- En sécurisant de manière plus efficace les fichiers compressés.
- En construisant des dictionnaires à partir de plus grandes bases de données permettant l'application de notre chaîne de compression à plus de modalités de l'imagerie médicale.
- En améliorant les performances de notre approche en faisant une implémentation hardware et/ou en utilisant plus de niveaux de décomposition de la transformée en ondelettes.
- En exploitant la topologie de la carte de Kohonen pour construire une base de données de dictionnaires, donnant des données statistiques permettant d'aboutir à une méthode algébrique pour la construction des dictionnaires.

ANNEXES

Annexe : Dictionnaire DICOM

Nous allons donner dans ce qui suit les différentes modalités supportées par le format DICOM ainsi que quelques codes utilisés pour la représentation des informations et des services lors des communications. Le dictionnaire complet DICOM est disponible dans la partie 6 du standard [12].

A.1 Liste des modalités supportées :

Dans cette partie, nous donnons les modalités actuellement supportons par DICOM ainsi que les codes qui les représentent.

AS: Angioscopy

AU: Audio

BI: Biomagnetic imaging

CD: Color flow Doppler

CF: Cinefluorography

CT: Computed Tomography

DD: Duplex Doppler

DX: Digital Radiography

ECG: Electrocardiography

EPS: Cardiac Electrophysiology

ES: Endoscopy

GM: General Microscopy

HC: Hard Copy

HD: Hemodynamic Waveform IO Intra-oral Radiography

IVUS : Intravascular Ultrasound

LS : Laser surface scan

MG: Mammography

MR: Magnetic Resonance

NM: Nuclear Medicine

OP: Ophthalmic Photography

OT: Other

PR: Presentation State

PT: Positron emission tomography (PET)

PX: Panoramic X-Ray

RF: Radio Fluoroscopy

RG: Radiographic imaging (conventional film/screen)

RTDOSE: Radiotherapy Dose

RTIMAGE: Radiotherapy Image

RTPLAN: Radiotherapy Plan

RTRECORD:

RT: Treatment Record

RTSTRUCT: Radiotherapy Structure Set

SM: Slide Microscopy

SMR: Stereometric Relationship

SR: SR Document

ST: Single-photon emission computed tomography (SPECT)

TG: Thermography

US: Ultrasound

XA: X-Ray Angiography

XC: External-camera Photography

A.2 Balises utilisées pour le codage des attributs :

Cette partie est consacrée aux codes des balises servant à identifier les différents attributs d'un *IOD*. Nous ne donnerons ici que les balises jugées importantes et présentes dans la majorité des fichiers.

(0002, 0010) Transfer Syntax UID

(0008, 0008) Image Type

(0008, 0016) SOP Class UID

(0008, 0018) SOP Instance UID

(0008, 0020) Study Date

(0008, 0021) Series Date

(0008, 0022) Acquisition Date

(0008, 0030) Study Time

(0008, 0031) Series Time

(0008, 0032) Acquisition Time

(0008, 0060) Modality

(0008, 0070) Manufacturer

(0008, 0080) Institution Name

(0008, 0081) Institution Address

(0008, 1010) Station Name

(0008, 1030) Study Description

(0008, 103E) Series Description

(0008, 1090) Manufacturer's Model Name

(0010, 0010) Patient's Name

(0010, 0020) Patient ID

(0010, 0030) Patient's Birth Date

(0010, 0032) Patient's Birth Time

(0010, 0040) Patient's Sex

(0010, 1010) Patient's Age

(0018, 0015) Body Part Examined

(0018, 0022) Scan Options

(0018, 0050) Slice Thickness

(0018, 0088) Spacing Between Slices

(0018, 1020) Software Versions(s)

(0018, 1030) Protocol Name

(0020, 000D) Study Instance UID

(0020, 000E) Series Instance UID

(0020, 0010) Study ID

(0020, 0011) Series Number

(0020, 0013) Image Number

(0020, 0032) Image Position (Patient)

(0020, 0037) Image Orientation (Patient)

(0020, 4000) Image Comments

(0028, 0002) Samples per Pixel

(0028, 0010) Rows

(0028, 0011) Columns

(0028, 0100) Bits Allocated

(0028, 0106) Smallest Image Pixel Value

(0028, 0107) Largest Image Pixel Value

(7FE0, 0010) Pixel Data

A.3 SOP Classe UIDs :

Dans cette partie, nous définirons quelques des *UIDs* des classes de services les plus utilisées.

1.2.840.10008.5.1.1.29: Hardcopy Grayscale Image Storage

1.2.840.10008.5.1.1.30: Hardcopy Color Image Storage

1.2.840.10008.5.1.4.1.1.1: Computed Radiography Image Storage

1.2.840.10008.5.1.4.1.1.1.1: Digital X-Ray Image Storage – For Presentation

1.2.840.10008.5.1.4.1.1.1.1.1: Digital X-Ray Image Storage – For Processing

1.2.840.10008.5.1.4.1.1.2: CT Image Storage

1.2.840.10008.5.1.4.1.1.2.1: Enhanced CT Image Storage

1.2.840.10008.5.1.4.1.1.4: MR Image Storage

1.2.840.10008.5.1.4.1.1.4.1: Enhanced MR Image Storage

1.2.840.10008.5.1.4.1.1.4.2: MR Spectroscopy Storage

1.2.840.10008.5.1.4.1.1.6.1: Ultrasound Image Storage

1.2.840.10008.5.1.4.1.1.9.1.2: General ECG Waveform Storage

1.2.840.10008.5.1.4.1.1.9.1.3: Ambulatory ECG Waveform Storage

1.2.840.10008.5.1.4.1.1.9.2.1: Hemodynamic Waveform Storage

1.2.840.10008.5.1.4.1.1.9.3.1: Cardiac Electrophysiology Waveform Storage

1.2.840.10008.5.1.4.1.1.88.11: Basic Text SR

1.2.840.10008.5.1.4.1.1.88.22: Enhanced SR

1.2.840.10008.5.1.4.1.1.88.33: Comprehensive SR

A.4 Transfer Syntaxe UIDs

Cette dernière partie énumère des *UIDs* de syntaxes de transfert utilisées dans DICOM lors des transmissions pour définir la méthode de compression et de communication utilisée.

1.2.840.10008.1.2: Implicit VR Little Endian: Default Transfer Syntax for DICOM

1.2.840.10008.1.2.1: Explicit VR Little Endian

1.2.840.10008.1.2.1.99: Deflated Explicit VR Little Endian

1.2.840.10008.1.2.2: Explicit VR Big Endian

1.2.840.10008.1.2.4.50: JPEG Baseline (Process 1): Default Transfer Syntax for Lossy JPEG 8 Bit Image Compression

1.2.840.10008.1.2.4.51: JPEG Extended (Process 2 & 4): Default Transfer Syntax for Lossy JPEG 12 Bit Image Compression (Process 4 only)

1.2.840.10008.1.2.4.57: JPEG Lossless, Non-Hierarchical (Process 14)

1.2.840.10008.1.2.4.70: JPEG Lossless, Non-Hierarchical, First-Order Prediction (Process 14 [Selection Value 1]): Default Transfer Syntax for Lossless JPEG Image Compression

1.2.840.10008.1.2.4.80: JPEG-LS Lossless Image Compression

1.2.840.10008.1.2.4.81: JPEG-LS Lossy (Near-Lossless) Image Compression

1.2.840.10008.1.2.4.90: JPEG 2000 Image Compression (Lossless Only)

1.2.840.10008.1.2.4.91 JPEG 2000 Image Compression

1.2.840.10008.1.2.4.100 MPEG2 Main Profile @ Main Level

1.2.840.10008.1.2.5 RLE Lossless

BIBLIOGRAPHIE

Bibliographie

- [1] David Salomon, “*Data Compression Third Edition*”, Springer-Verlag London Limited 2007.
- [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] András CZIHÓ, “*Quantification Vectorielle et Compression D’image. Application à L’imagerie Médicale*”, thèse doctorat de l’université de Rennes 1, 5 mai 1999.
- [4] 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.
- [5] Omar Hammami, “*Etudes d’optimisation algorithmiques de JPEG2000 (EIRE)*”, ministère de l’industrie dans le cadre du Réseau National de la Recherche en Télécommunications (RNRT), février 2004.
- [6] Yann-Gaudeau, “*Contribution en compression d’images médicales 3D et d’images naturelles 2D*”, thèse de l’université Henri Poincaré Nancy, 2006.
- [7] 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.
- [8] Zhou Wang and Eero P. Simoncelli, “*Translation insensitive image similarity in complexwavelet domain*”, IEEE Inter. Conf. Acoustic, Speech & Signal Processing Volume II, Pages 573-576, March 2005
- [9] 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.

- [10] Tinku Acharya and Ping-Sing Tsai, “*JPEG2000 Standard for Image Compression Concepts, Algorithms and VLSI Architectures*”, Canada 2005 by John Wiley & Sons, Inc.
- [11] Cours “*Théorie de l’information*”, Pr. D.BERKANI, Département d’électronique, Ecole Nationale Polytechnique d’Alger.
- [12] DICOM Committee, Digital Imaging and Communication in Medicine. Release 2007 by The NEMA.
- [13] A.NAÏT-ALI et C.CAVARO-MENARD, “*Compression des images et des signaux médicaux*”, LAVOISIER, Hermes Sciences publications 2007
- [14] <http://eviewbox.sourceforge.net/tutorial/intro.html>
- [15] <http://www.rsna.org/Technology/DICOM/intro/elemental.cfm>
- [16] http://fr.wikipedia.org/wiki/Open_Systems_Interconnect
- [17] M.BUREL, C.OBERT et L.TERRACOL, “*DICOM et Quantification vectorielle* ”, Rapport de recherche, INSA de Rouen, 2005
- [18] www.tsi.enst.fr/~tupin/TDI/poly_contours.pdf
- [19] 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 2000.
- [Images biomédicales] <http://www.medavis.com/>