

**COMPRESSION DES IMAGES MEDICALES FIXES
PAR RESEAU DE NEURONES**

Nacéra Benamrane,

Maître de conférences en Informatique
nabenamrane@yahoo.com, + 213 41 41 53 22

Zakaria Benahmed Dahou

Etudiant doctorant
z-dahou@lycos.com

Jun Shen,

Professeur en Informatique
shen@egid.u-bordeaux.fr +33 5 57 12 10 26

Adresse professionnelle

Université des Sciences et de la Technologie d'Oran, Département d'Informatique ★ BP 1505 ★
EL-Mnaouer 31000 Oran, Algérie

Résumé : Dans cet article, nous proposons une technique de compression des images fixes basée sur le réseau de Kohonen. Pour palier à l'effet de blocs « ou de pixellisation » causé par les taux de compressions élevés, les blocs de l'image à compresser sont classifiés, selon leurs degrés d'activités, avant de les présenter au réseau de Kohonen pour une quantification vectorielle. Cette classification réalise une discrimination entre les blocs de faibles activités et les blocs contours de hautes activités. Les blocs homogènes et les blocs contours sont codés séparément en utilisant des codebooks différents. Par cette technique, nous avons obtenu une amélioration sensible de la qualité visuelle des images médicales reconstruites tout en maintenant un taux de compression élevé

Summary: The image blocks to be compressed are classified according to their activity degree before presenting them to Kohonen's network for a vector quantization. This classification allows discrimination between the blocks with high activity (edge blocks) and blocks with low activity. The blocks of high activity are divided into small blocks of 2x2 pixels. Blocks of high and low activity are coded separately with different codebooks. We have obtained a noticeable improvement of visual quality of all the rebuilt medical images while keeping an important compression rate.

Mots clés : Compression, Images médicales, Réseau de Kohonen, Quantification vectorielle, Classification.

Compression des images médicales fixes par réseau de neurones

Dans de nombreux domaines, l'image numérisée remplace les images analogiques classiques. Dans le domaine médical, l'utilisation des images radiographies, ultrasonores, IRM, ... pose un grand problème de stockage et d'archivage. Par exemple ; un hôpital de 200 lits, produit en moyenne chaque année 875 Go de données images. En plus du problème de stockage, si de telles images doivent être transmises via un réseau, la durée de la transmission est souvent trop longue. Pour palier à tous ces problèmes, la compression de ces images devient une opération nécessaire et impérative. Le but principal de la compression des images est de réduire la quantité de bits nécessaire pour les décrire tout en gardant un aspect visuel acceptable des images reconstruites.

La compression des images fixes a été réalisée par plusieurs techniques parmi les plus connues : le JPEG qui est une méthode avec perte standardisé par ISO en août 1990, son principe est détaillé par Gutter (1995), le codage en sous bandes proposé par Woods (86) et Nezit-Gerek (2000), la transformée en Ondelette proposée par Mallat (1989), le JPEG 2000 qui est une nouvelle norme développée par ISO en 2000, son principe est détaillé par Ordóñez et al(2003) ect ... Dans le contexte particulier de l'imagerie médicale, Le duff a proposé une méthode utilisant séparément l'ondelettes et le JPEG pour la compression de séquences spatiales d'IRM cérébrales, les résultats obtenus par ondelettes sont supérieurs en terme de qualité des images reconstruites. Ordóñez et al (2003) ont proposé une technique basée sur le JPEG 2000 pour compresser des images tumeurs de cerveau en vue d'une indexation basée sur les informations spectrale et spatiale.

Ces méthodes effectuent la compression en réalisant une quantification scalaire (*QS*) sur les valeurs obtenues après transformations. L'inconvénient de la quantification scalaire est qu'elle ne permet pas d'exploiter la corrélation spatiale qui existe entre les différents pixels de l'image. Une autre façon plus intéressante pour réaliser la compression est de coder non pas les valeurs individuellement les unes après les autres, mais de coder un ensemble de valeurs simultanément. Cette manière de procéder est appelée «quantification vectorielle» (*QV*).

La quantification vectorielle (*Vector Quantization*) a été utilisée avec succès pour le codage du signal vocal par Linde (80) ainsi que pour la compression des images fixes par Nasrabasi (1988), Gerso (1982) et Ramamurthi (1986). Pour réaliser la quantification vectorielle plusieurs systèmes ont été proposés par Ramamurthi (1986) et Gray (1984). Ces systèmes se basent sur l'algorithme de Lynd, Buzo et Gray (1980).

Les approches utilisant les réseaux de neurones artificiels pour le traitement intelligent des données semblent être très prometteuses, ceci est essentiellement dû à leurs structures offrant des possibilités de calculs parallèles ainsi que l'utilisation du processus d'apprentissage permettant au réseau de s'adapter sur les données à traiter. Les nouvelles techniques basées sur les réseaux de neurones comme outils de compression ont été proposées par Jiang (1999), Robert (1995) et Stanley et al (1990). Le réseau de Kohonen (1990) est un réseau de neurones particulier, il peut être utilisé comme quantificateur vectoriel pour les images.

Cet article traite la méthode de compression basée sur le réseau de neurones de Kohonen pour des images médicales. Ce papier comprend quatre sections. Dans la section 2, le réseau Kohonen est présenté ainsi que son utilisation pour réaliser la quantification vectorielle. Dans la section 3, il sera présenté les résultats de compression obtenus sur des images médicales. Le problème produit par des blocs de taille importante sera aussi mis en évidence. Une amélioration de la méthode sera proposée dans la section 4. Enfin, la dernière section est consacrée pour la conclusion et les perspectives.

1 - QV PAR LE RESEAU DE KOHONEN

Le réseau de Kohonen (1990) (le *Self-Organizing Feature Map*) est un réseau à compétition (Jodouin, 1994). Ce réseau est composé d'une couche de neurones qui reflète passivement les données d'entrée présentées au réseau, et d'une carte topologique. Cette dernière est composée d'un certain nombre de neurones organisés selon une structure bien définie. Cette structuration lie les neurones ensemble en une surface «élastique», et les

contraint à respecter un certain comportement lors de l'apprentissage.

Lorsqu'une entrée est présentée au réseau via la couche d'entrée, les neurones de la carte topologique s'activent différemment et une compétition s'installe entre ces neurones. Le neurone dont le vecteur des poids synaptiques se rapprochant le plus de l'entrée est déclaré vainqueur de la compétition.

Le réseau de Kohonen a été utilisé dans plusieurs applications ; par exemple : la reconnaissance des formes, traitement acoustique du signal vocal, problèmes de classifications par Kohonen (1990) et la compression par Lebail (1989) et Nasrabadi (1988). Pour réaliser la quantification vectorielle, un dictionnaire de données (Codebook) est nécessaire (Figure 1).

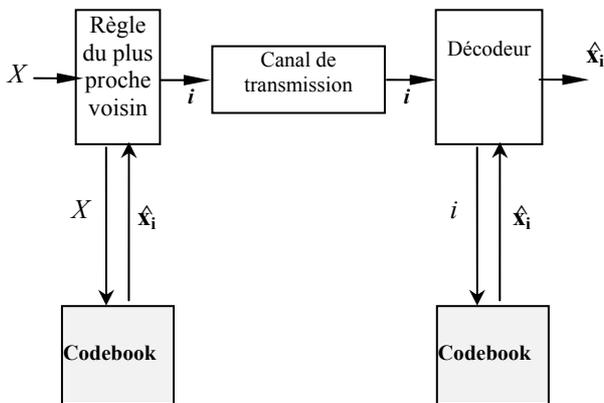


Figure 1. Système de compression Par QV

La quantification vectorielle du vecteur d'entrée X produira en sortie l'indice i de son meilleur représentant \hat{x}_i choisi dans le Codebook suivant la règle du plus proche voisin (Nearest neighbor rule) :

$$\hat{x}_i \text{ est le meilleur représentant de } X \Leftrightarrow d(X, \hat{x}_i) \leq d(X, \hat{x}_j) ; \forall i, j \quad [1]$$

avec d est une mesure de distance.

Plusieurs recherches ont montré que l'utilisation du réseau de Kohonen permet de générer un Codebook assurant une meilleure représentation des données à traiter, ce qui entraîne une diminution du bruit de quantification.

Le réseau de Kohonen effectue une projection de l'espace des vecteurs d'entrées dans un sous-ensemble fini de vecteurs de représentants (le Codebook), produisant ainsi

un Codebook localement optimal. Kohonen (1990) a développé un algorithme dans lequel les poids synaptiques w_i du réseau reflètent, enfin du processus d'apprentissage, le contenu des vecteurs des représentants. L'adaptation des poids du réseau se fait selon les règles suivantes :

- 1) trouver le neurone c , vainqueur de la compétition tel que :

$$d(X, w_c) \leq d(X, w_i) ; \forall i ; \quad [2]$$

- 2) mettre à jour les poids w_i du réseau :

$$w_{i(t+1)} = w_{i(t)} + h(c, i, t) * [X - w_{i(t)}] \quad [3]$$

où $w_i(t)$ est le vecteur poids du neurone i à l'instant t , h est une fonction définie par :

$$h(c, i, t) = \begin{cases} \alpha(t) & \text{si } i \in N(c, t) ; \\ 0 & \text{sinon} \end{cases} \quad [4]$$

avec $\alpha(t) \in [0, 1]$

La fonction h définit l'ampleur de la correction apportée au neurone vainqueur c ainsi qu'à son voisinage. Le voisinage, à l'instant t , du neurone vainqueur c est déterminé par la fonction N qui est une fonction décroissante par rapport au temps. Le voisinage final d'un neurone se compose du neurone lui-même. La règle [4] permet d'attribuer la même correction $\alpha(t)$ à tous les neurones appartenant au voisinage du neurone vainqueur à un instant t . Une autre règle peut être utilisée stipulant que plus un neurone est loin du vainqueur, plus faible sera sa correction. Dans ce cas, le pas d'apprentissage sera en fonction du temps t et de la distance entre les neurones de la carte topologique.

Kohonen a démontré l'efficacité de son réseau dans la reconnaissance et la compression du signal vocal. Dans la section qui suit sont exposés les résultats de compression obtenus sur des images médicales.

2 - RESULTATS EXPERIMENTAUX ET DISCUSSIONS

Les images utilisées dans le processus d'apprentissage sont de type médical de format 256 x 256 pixels. Chaque pixel est codé sur 8 bits. La figure 2, montre une des images utilisées pour l'apprentissage.

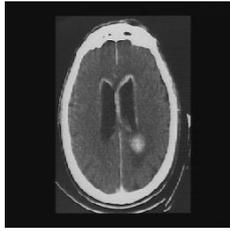


Figure 2. Une image utilisée dans l'apprentissage

Lors de l'entraînement, les images sont découpées en blocs carrés qui constituent les vecteurs d'entrées. Après une opération de normalisation, les blocs sont utilisés pour la génération du Codebook suivant les règles [2], [3] et [4].

La qualité des images reconstruite est évaluée par le PSNR (Peak Signal Noise Ratio) et par l'EQM (Erreur Quadratique Moyenne) :

$$EQM = \frac{1}{T} \sum_{i=1}^T (\hat{n}_i - n_i)^2 \quad [5]$$

$$PSNR = 10 \log_{10} \left(\frac{255^2}{EQM} \right) \text{ (en dB)} \quad [6]$$

avec T : taille de l'image, n_i : $i^{\text{ème}}$ pixel de l'image originale, \hat{n}_i : $i^{\text{ème}}$ pixel de l'image reconstruite.

Les images de tests utilisées sont présentées ci-dessous :

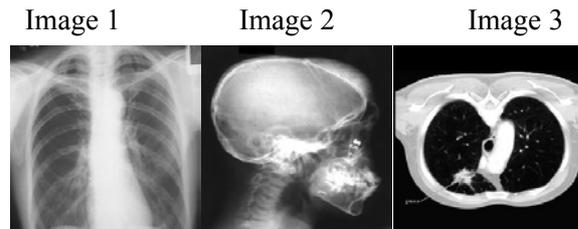


Figure 3. Les images utilisées dans les tests.

2.1 - Taille des vecteurs d'entrées égale à 4 pixels avec une carte topologique de 256 neurones

Dans cette expérience, une carte topologique carrée de 16 x 16 neurones a été utilisée. Chaque image à compresser est découpée en blocs non recouvrant de taille 2 x 2 pixels. Dans les figures 4, 5 et 6, les images reconstruites ainsi que les images différences sont présentées.

La qualité des images reconstruites est très bonne, néanmoins, les images différences montrent que les erreurs de codage sont situées principalement aux contours des images (zones en gris clair). Dans le tableau 1 est présenté le résultat de l'évaluation du $PSNR$ [6], EQM [5], du Taux de Compression (TC) et du Nombre de Bits par Pixels ($bit\ rate$) (NBP) :

$$NBP = \frac{\log_2(\text{nombre de neurones dans la carte})}{\text{taille des vecteurs d'entrées}}$$

[7]

$$TC = 100 * (1 - (NBP/8)) \text{ (en \%)} \quad [8]$$

Pour un nombre de bits par pixel égal à 2 (taux de compression égal à 75 %), le $PSNR$ est supérieure à 30 dB pour l'ensemble des images de tests.

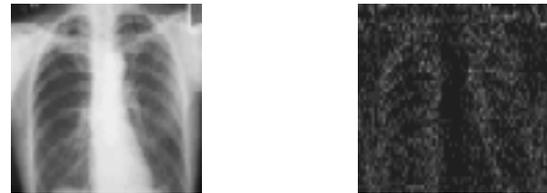


Figure 4. Image 1 reconstruite et l'image différence.

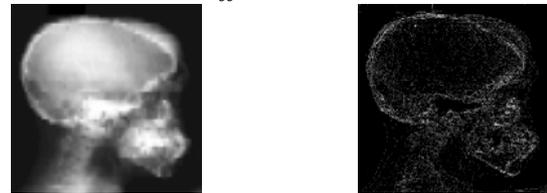


Figure 5. Image 2 reconstruite et l'image différence

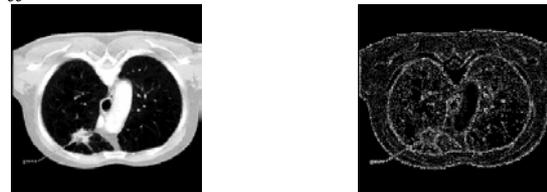


Figure 6. Image 3 reconstruite et l'image différence

	$PSNR$ (dB)	EQM	TC %	NBP bits/pixel
Image 1	36.21	15.52	75	2
Image 2	35.36	18.90	75	2
Image 3	31.08	50.71	75	2

Tableau 1. Résultats de compressions.

Dans l'expérience qui suit, la taille des vecteurs d'entrées est augmentée, ceci permet d'obtenir des taux de compression plus importants.

2.2 Taille des vecteurs d'entrées égale à 16 pixels

Dans cette expérience, les images à compresser sont découpées en blocs non recouvrant de taille 4 x 4 pixels, ceci permet d'obtenir un taux de compression égale à 93,75 % (0.5 bits / pixel) avec une carte de 256 neurones. Cette augmentation intéressante du taux de compression est suivie par une dégradation assez importante de la qualité des images reconstruites. La Figure 7, montre l'image n°3 reconstruite où l'effet de blocs apparaît clairement. Le PSNR obtenu est égal à 24.79 dB ce qui correspond à une perte de 6.29 dB par rapport au premier test.

Les résultats obtenus pour les trois images de tests sont présentés dans la tableau qui suit :

	PSNR (dB)	EQM	TC %	NBP bits/pixel
<i>Image 1</i>	32.03	40.72	93.75	0.5
<i>Image 2</i>	30.03	64.43	93.75	0.5
<i>Image 3</i>	24.79	215.74	93.75	0.5

Tableau 2. Résultats de compressions. Cas des vecteurs d'entrées de 16 pixels avec une carte de 16 x 16 neurones.

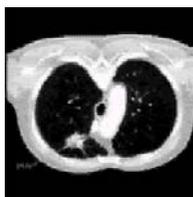


Figure 7. Image 3 reconstruite.

Le tableau 2 montre qu'en utilisant des vecteurs de dimensions 16, un gain important de 1.5 bits par pixel peut être atteint. Mais, en contre partie, une baisse moyenne de 5.26 dB est enregistrée dans la qualité des images reconstruites.

Donc, les performances (en terme de qualité des images) du système de compression par le réseau de Kohonen diminuent quand la taille des blocs d'entrées augmente, l'effet de bloc devenant en outre de plus en plus sensible

surtout au niveau des contours de l'image. Cette baisse de qualité peut très bien être expliquée ; plus la taille des blocs est importante, plus leurs contenus sont variés et donc les chances de trouver des blocs fortement ressemblants diminuent.

Une première tentative pour résoudre ce problème consiste à augmenter la taille de la carte topologique. Dans le test qui suit, la taille de la carte topologique est doublée à 512 neurones (tableau 3).

Les résultats obtenus montrent que même en augmentant la taille de la carte topologique à 512 neurones l'effet de blocs persiste. Le PSNR augmente de 0.52 dB en moyenne.

	PSNR (dB)	EQM	TC %	NBP bits/pixel
<i>Image 1</i>	32.85	33.70	92.96	0.5625
<i>Image 2</i>	30.67	55.68	92.96	0.5625
<i>Image 3</i>	24.91	209.87	92.96	0.5625

Tableau 3. Résultats de compressions. Cas des vecteurs d'entrées de 16 pixels avec une carte de 16 x 32 neurones.

Pour déterminer des solutions efficaces à ce problème, examinons le Codebook obtenu dans le cas des blocs de 16 pixels avec une carte de 256 neurones.



Figure 8. Codebook obtenu par un réseau de Kohonen de 256 neurones dans la carte avec des blocs de taille 4 x 4.

Dans la figure 8 est observé que les blocs contours sont faiblement représentés contrairement aux blocs homogènes. Ces derniers occupent une grande partie du Codebook. Ainsi, la cause effective de l'effet de blocs peut être expliqué par le fait qu'il y a un nombre insuffisant de vecteurs dans le Codebook permettant de représenter les différents blocs contours contenus dans les images aussi, la mesure de distance d utilisée, pour déterminer le représentant du vecteur d'entrée, ne peut garantir qu'un bloc contour

aura comme représentant un vecteur contour dans le *Codebook*. Par conséquent, le *Codebook* ne peut coder avec une bonne qualité, les différentes variétés des blocs contours que les images peuvent contenir.

La qualité des contours est très importante pour assurer une bonne qualité visuelle des images reconstruites. En plus, dans le cas de l'image médicale, les images décodées doivent être de très bonnes qualités pour éliminer le risque d'un diagnostic médical erroné dû à une image de mauvaise qualité visuelle. Une solution à ce problème consiste à créer plusieurs *Codebooks*, chacun étant spécialisé dans la représentation d'un certain type de blocs dans l'image.

3 – COMPRESSION DES BLOCS AVEC CLASSIFICATION

3.1- Taille des blocs égale à 16 pixels

Dans cette série d'expériences, lors de l'entraînement, les blocs des images utilisées sont classifiés selon la nature des détails contenus dans chaque bloc. Ainsi, plusieurs *Codebooks* seront créés par ce procédé. Chaque *Codebook* sera spécialisé dans le codage d'un type bien défini de blocs. Pour réaliser la classification d'un bloc, plusieurs manières existent ; par exemple : Gersho (1982), Ramamurthi (1986) et Le bail [1982] utilisèrent une classification basée sur les directions du gradient. Nansrabadi (1988) a utilisé la valeur de la variance pour distinguer entre deux classes de blocs. Dans nos expériences réalisées, les blocs sont classés selon leur degré d'activité. Le degré d'activité d'un bloc B est mesuré grâce à la fonction d'activité A_b définie par :

$$A_b(B) = \sum_{m,n} A_p(x_{m,n}); \quad x_{m,n} \in B \quad [9]$$

avec $x_{m,n}$: pixel appartenant au bloc B ;

A_p : fonction d'activité du pixel $x_{m,n}$;

$$A_p(x_{m,n}) = \sum_{i=-l}^{+l} \sum_{j=-l}^{+l} (x_{m,n} - x_{m-i,n-j})^2 \quad [10]$$

La fonction d'activité a pour objet l'identification et la discrimination entre deux classes de blocs ; les blocs de fortes activités et les blocs de faibles activités. Ceux qui appartiennent à la classe de hautes activités sont subdivisés encore en quatre classes selon l'orientation des détails contenus dans ces

blocs. Dans un bloc donné, les orientations de la structure des détails sont : horizontales (h), verticales (v), et les deux orientations diagonales (d et e). Pour définir ces orientations, des fonctions d'orientations sont utilisées :

$$B_h = \frac{1}{M(M-1)} \cdot \sum_{m=1}^M \sum_{n=1}^{M-1} (x_{m,n} - x_{m,n+1})^2 \quad [11]$$

$$B_v = \frac{1}{M(M-1)} \cdot \sum_{m=1}^{M-1} \sum_{n=1}^M (x_{m,n} - x_{m+1,n})^2 \quad [12]$$

$$B_d = \frac{1}{(M-1)^2} \cdot \sum_{m=1}^{M-1} \sum_{n=1}^{M-1} (x_{m,n} - x_{m+1,n+1})^2 \quad [13]$$

$$B_e = \frac{1}{(M-1)^2} \cdot \sum_{m=2}^M \sum_{n=1}^{M-1} (x_{m,n} - x_{m-1,n+1})^2 \quad [14]$$

avec : $M = 4$; et $x_{n,m}$: pixel du bloc B ;

n et m : la position du pixel dans le bloc.

Chacune des fonctions d'orientations définit une classe de blocs. La classification est réalisée de la manière suivante :

- Pour chaque bloc appartenant à la classe haute activité, calculer les valeurs des quatre fonctions d'orientation B_h , B_v , B_d et B_e .
- La plus petite valeur parmi les quatre fonctions détermine l'orientation du bloc. Une fois les cinq *Codebooks* construits, le processus de compression pourra être entamé. Les étapes de la compression d'un bloc sont les suivantes (Figure.9) :
- L'activité du bloc X à compresser est calculée selon [9] ;
- Déterminer si le bloc a une forte ou faible activité (selon un certain seuil à déterminer) ;
- Si le bloc appartient à la classe de haute activité, déterminer l'orientation des contours dans le bloc selon [11], [12], [13] et [14] ;
- Déterminer le *Codebook* à utiliser pour le codage.
- Envoyer au décodeur l'indice du meilleur représentant du bloc X .

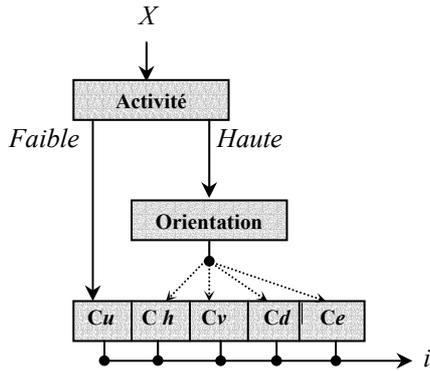


Figure 9. Système de compression par classification.

Ch :Codebook des blocs d'orientation horizontale ;
Cv :Codebook des blocs d'orientation verticale ;
Cd :Codebook des blocs d'orientation diagonale selon la première bissectrice;
Ce :Codebook des blocs d'orientation diagonale selon la seconde bissectrice;
Cu :Codebook des blocs de faibles activités.

Les résultats de compression en utilisant la classification sont présentés dans le tableau 4. En comparant les tableaux 4 et 2, on remarque que pour le même taux de compression (93.75 % et 0.5 bits / pixel), la qualité des images compressées par classification est assez meilleur. Un gain moyen de 0.58 dB est obtenu sur les images décomposées.

	PSNR (dB)	EQM	TC (%)	NBP bits/pixel
Image 1	32.41	37.31	93.75	0.5
Image 2	30.68	55.58	93.75	0.5
Image 3	25.51	182.5 2	93.75	0.5

Tableau 4. Résultats de compressions par classification. Cas des vecteurs d'entrées de 16 pixels avec cinq Codebooks de taille 256 représentants.

En utilisant cette technique, les différents tests réalisés sur les images montrent que l'effet de blocs, observé sur les contours des images, est réduit mais il n'est pas tout à fait éliminé. Ceci est expliqué par le fait de répartir les blocs en cinq classes est insuffisant. Les blocs de taille 16 pixels sont très variés et nécessitent alors un nombre élevé de classes.

Pour résoudre ce problème, tout en gardant une complexité raisonnable de la technique de compression, nous proposons d'ajouter une classification sur les blocs contours après les avoir découpés en petits blocs de taille 2 x 2 pixels.

3.2- Classification avec blocs à compresser de taille variable, contours de tailles 4 pixels et blocs homogènes de tailles 16 pixels

La technique que nous proposons maintenant se base sur les observations faites dans plusieurs séries de tests. En effet, nous avons constaté que :

1. La *QV* réalisée en utilisant des blocs de tailles 4 pixels assure une bonne reconstruction des images. Les erreurs de codage se situent généralement sur les contours des images reconstituées ;
2. Le taux de compression assuré par les blocs de 16 pixels est très intéressant néanmoins, l'effet de blocs est très visible sur les contours des images décompressées ;
3. La classification (en cinq classes) réalisée sur les blocs de taille 16 pixels n'augmente pas de manière considérable la qualité des images reconstituées ;
4. Le taux de compression et la qualité des images sont beaucoup plus influencés par la variation de la taille des blocs que par la taille des *Codebooks* utilisés.

Pour assurer un taux de compression assez élevé et une bonne qualité visuelle des images, nous proposons donc la technique suivante (figure 10) :

- Découper l'image en blocs de taille 4 x 4 pixels ;
- Déterminer l'activité du bloc à compresser *X* ;
- Si le bloc *X* appartient à la classe de haute activité, le subdiviser en 4 sous blocs (*Y_k*) de taille 2 x 2 pixels ;
- Déterminer l'activité pour chaque sous bloc *Y_k* crée ;
- Si le sous bloc appartient à la classe de haute activité, déterminer l'orientation des contours contenant le sous bloc ;
- Déterminer le *Codebook* à utiliser pour le codage ;
- Envoyer au décodeur l'indice du meilleur représentant du bloc.

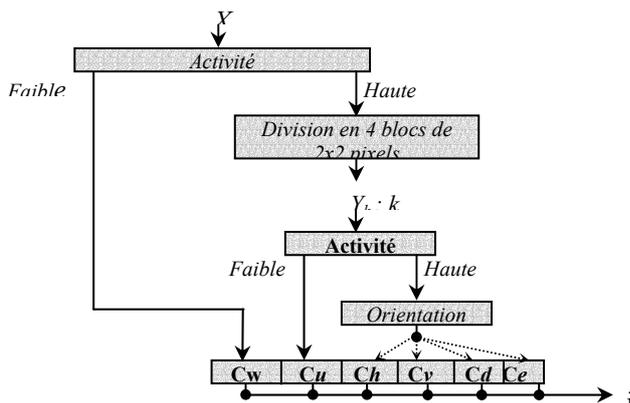


Figure 10. Le principe de la méthode proposée

C_w : Codebook pour quantifier les blocs 4 x 4 de faible activité ;

C_u : Codebook pour quantifier les blocs 2 x 2 de faible activité ;

C_h, C_v, C_d, C_e : Codebooks pour quantifier les blocs 2 x 2 de hautes activités (blocs contours).

Les résultats obtenus sur les images 1, 2 et 3 sont :

	PSNR (dB)	EQM	TC (%)	NBP bits/pixel
<i>Image 1</i>	34,43	23,40	90,14	0,78
<i>Image 2</i>	33,61	28,29	91,20	0,70
<i>Image 3</i>	29,64	70,57	91,91	0,64

Tableau 5. Résultats de compressions par la méthode proposée.

Ces résultats ont été obtenus en utilisant six codebooks. Le codebook associé aux blocs de 16 pixels comporte 256 neurones assurant ainsi une bonne fidélité des blocs homogènes. Les blocs de petite taille sont peu variés (par rapport aux blocs plus grands), des petits codebooks sont alors suffisants pour les coder, nous avons utilisés des codebooks de 64 neurones. L'image 3 reconstruite est présentée en figure 11.



Figure 11. Image 3 reconstruite.

La technique proposée permet d'augmenter la qualité des images reconstruites tout en gardant un taux de compression assez élevé. Pour l'image 3, par exemple ; un PSNR égal à 29.64 dB est obtenu avec un codage sur 0.64 bits/pixel (Figure 11). En comparant les résultats obtenus avec la même image dans le tableau 2, un gain dans le PSNR égal à 4.85 dB en plus, l'effet de blocs est éliminé (comparer la figure 11 avec la figure 7).

4 - CONCLUSION

Dans cet article, une technique de compression des images médicales fixes est présentée. L'utilisation des blocs d'images de tailles 16 pixels permet de réaliser un codage à 0.5 bits par pixel. L'augmentation du taux de compression est suivie par une dégradation visuelle de la qualité des images décodées et ceci est essentiellement dû à l'effet de blocs. Pour palier à cet inconvénient, une classification des blocs a été réalisée. En affinant la compression des blocs contours, en créant des sous blocs plus petits qui subissent une pré-classification avant de les quantifier, les contours des images reconstruites sont d'une très bonne qualité. L'utilisation des blocs homogènes de taille 16 pixels permet d'augmenter le taux de compression. L'intégration de la quantification vectorielle par le réseau de Kohonen dans un système de compression classique (par Ondelettes par exemple) permettra de tirer avantage de la simplicité et des performances du réseau pour diminuer le bruit de quantification et d'augmenter la qualité des images reconstruites.

Remerciements

Nous témoignons notre reconnaissance aux aides financières dont nous avons bénéficié dans le cadre du projet PNR-ANDRS (agence nationale de recherche en médecine) n°05/04/01/98/071.

BIBLIOGRAPHIE

Le bail, E., Mitche, A. (1989), "Quantification vectorielle d'images par le réseau neuronal de kohonen", Traitement du signal, Vol. 6, N° 6, 1989.

Le duff, A., "La compression d'images : une nécessité pour l'archivage et la

- télé-médecine”, http://www.eseo.fr/~aleduff/Articles/Pria/article_pria.htm.
- Gersho, A., Ramamurthi, B. (1982), *image coding using vector quantization*.
- Gray, R. (1984), “Vector quantization”, *IEEE ASSP Magazine*.
- Gutter, H. (1995), *La Compression des images numériques*, Edition Hermes, Paris
- Jiang, J. (1999), “Image compression with neural networks – A survey”, *Signal processing : image communication n°14*, pp 737-760.
- Jodouin, J.F. (1994), *Les réseaux neuromimétiques*, Edition Hermes
- Kohonen, T. (1990), “The Self-organizing map”, *Proceedings of the IEEE, vol. 78, n° 9*.
- Linde, Y., Buzo, A., Gray, R. (1980), “An algorithm for vector quantizer design”, *IEEE Transactions on Communications*, vol com-28, n° 1.
- Mallat, S. G. (1989), “A theory of multiresolution signal decomposition : The Wavelet representation”, in *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol 11, n° 7..
- Nasrabadi, M.N., Feng, Y. (1988), “Vector Quantisation of Images Based Upon the Kohonen Self-Organizing Feature Map.”, in *IEEE International Conference on Neural Networks, San Diego, California*, pp101-108.
- Nasrabadi, M.N., King, R.A. (1988), “Image Coding Using Vector Quantization : A Review”, in *IEEE Transactions on Communications*, vol 36, n° 8.
- Nezit-Gerek, O., Enis Cetin, A. (2000), “Adaptive polyphase subband decomposition structures for image compression”, in *IEEE Transactions on Image Processing*, vol 9, n°10.
- Ordóñez, J.R., Gazuguel, G, Puentes, J., Solaiman, B., Roux, C. (2003), “Indexation d’images médicales basée sur les informations spectrale et spatiale extraites de JPEG-2000”, *Actes de CORESA’03, Lyon*.
- Ramamurthi, B., Gersho, A. (1986), “Classified vector quantization of images”, in *IEEE Transactions on Communications*, vol com-34, n° 11.
- Robert, D. D., Haykin, S. (1995), “Neural network approaches to images compression”, in *Proceedings of the IEEE*, vol 82, n° 2..
- Stanley, C. and al. (1990), “Competitive learning algorithms for vector quantization”, in *Neural networks*, vol 3, pp277-290.
- Woods, J.W., O’neil, S.D. (1986), “Subband Coding”, in *IEEE Transaction on Acoustics, Speech, and Signal Processing*, vol 34, n° 5.