

Technique de compression des images médicales 4D

Leila Belhadef¹ et Zoulikha Mekkakia¹

¹ Département d'Informatique, USTO-MB,
BP 1505 El Mnaouer, Oran, Algérie
l.belhadef@gmail.com, mekkakia@univ-usto.dz

Résumé. L'image médicale 4D représente une succession d'images volumiques sur l'axe temporelle, ce qui représente une énorme quantité de données à compresser afin de la stocker ou de la transmettre. Dans cet article, nous présentons une technique de compression d'images médicales 4D basée sur le SPIHT3D. Cette technique applique la transformée en ondelettes 3D afin d'éliminer efficacement les redondances spatiales et temporelles entre les coupes 2D composant l'image 4D. La démarche consiste en premier lieu à représenter les coupes 2D sous forme de GOS (Group of slices), ensuite ces GOS sont décomposés spatialement par les ondelettes. Enfin les coefficients obtenus sont codés à l'aide du SPIHT3D.

Mots clés Compression, images médicales 4D, SPIHT3D, transformée en ondelettes.

1 Introduction

Les images médicales ne cessent de se développer en donnant une représentation de plus en plus précise des différentes parties du corps humain. Cependant plus l'image est précise plus la quantité des données engendrées est grande. Les images médicales tel que l'IRM fonctionnelle, l'image tomographique dynamique et l'échocardiographie 3D dynamique sont de plus en plus utilisées, car elles sont considérées parmi les techniques les plus performantes en imagerie médicale, mais elles produisent les données les plus volumineuses, d'où la nécessité de leur compression pour des fins de stockage et/ou de transport via des réseaux de télécommunication.

Il existe de nombreuses techniques pour compresser les images médicales 4D, la technique la plus intuitive pour compresser les images médicales 4D est d'étendre les techniques issues de la compression des images 2D et 3D au cas 4D. Dans [1] trois approches pour compresser les images médicales 4D (fMRI) sont comparées en terme de PSNR: une approche 2D, une approche 3D et une approche 4D. Dans la première approche chaque coupe 'xy' est traité comme une image 2D et compressée en utilisant JPEG2000. La deuxième approche compresse chaque volume indépendamment en utilisant le 3D-JPEG2000 [2]. Cette technique réalise d'abord une transformation par ondelettes 1D le long de la troisième dimension à travers les coupes, ensuite les coupes transformées résultantes sont comprimées en utilisant JPEG2000. Pour les ensembles de données 4D, on peut utiliser la approche 3D soit sur les cubes « xyt »

ou sur les cubes « xyz ». La troisième approche consiste à utiliser la transformation par ondelettes pour décorrélérer les données le long de la quatrième dimension. Dans ce cas la transformation par ondelettes 1D est appliquée le long de t suivi d'une transformation par ondelettes 1D le long de z ensuite les coefficients résultants d'ondelette sont compresser en tant que des coupe 2D en utilisant JPEG2000. D'après les résultats obtenus on déduit que la méthode 2D est la moins adapté pour la compression des images médicales 4D en comparaison avec les autres méthodes. C'est l'approche 4D qui donne les meilleurs résultats mais ces résultats sont perçus que les mêmes obtenus avec l'approche 3D réalisée sur les cubes « xyt ». Puisqu'il y a peu de corrélation dans la dimension z en comparaison avec les autres dimensions.

Il existe d'autres techniques de compression d'images médicales 4D, elles se distinguent par l'utilisation de la compensation de mouvement ce qui permet d'exploiter efficacement les redondances spatiales et temporelles existantes entre les images volumiques constituant la séquence 4D [3], [4], [5].

Le reste de l'article est organisé comme suit. La section 2 expose notre approche de compression des images médicales 4D. La section 3 rapport les résultats obtenus. Enfin une conclusion est présentée dans la section 4.

2 La méthode proposée

Nous proposons une méthode de compression d'images médicales 4D qui se base sur la transformée en ondelettes 3D et le codeur SPIHT3D, son principe se résume en trois étapes :

Etape 1 :

La première étape consiste à représenter chaque volume par N sous ensembles appelés GOS (Group of Slices) ou chaque GOS est composé de M coupes 2D d'un même volume. Ensuite, chaque GOS est codé séparément (Fig.1).

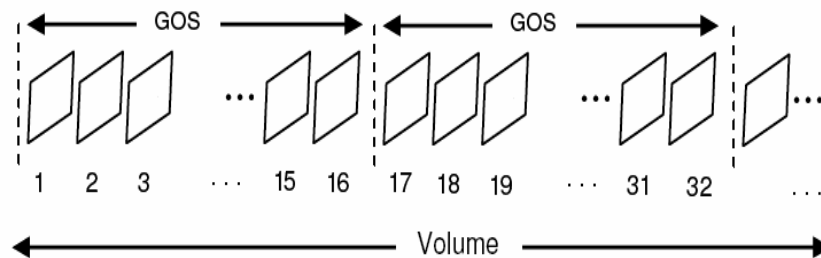


Fig. 1. Séquences obtenues après réorganisation des coupes 2D en GOS (*Group of Slices*) avec $M=16$ coupes.

Etape 2 :

Dans cette étape les GOS sont transformées en utilisant la transformée en ondelettes 3D. En compression d'image les ondelettes sont appréciées pour leur capacité à obtenir une représentation creuse de l'image en concentrant dans peu de coefficients

l'énergie du signal, permettant ainsi un codage efficace de l'image. Il existe deux manières pour calculer les coefficients d'ondelettes, par convolution ou par lifting.

La technique par convolution peut être vue comme une décomposition du signal à différents niveaux de résolution (analyse multi-résolution) et peut être implantée en utilisant un couple de filtres discrets à travers un schéma classique d'analyse/synthèse. La technique par lifting est une amélioration du schéma par convolution [6]. Elle consiste dans un premier temps à séparer les échantillons pairs (x_e) et impairs (x_0) dans 2 tableaux différents. Si $x(n)$ est le signal d'entrée, ce découpage se note :

$$x_e = x(2n) \quad (1)$$

$$x_0 = x(2n+1) \quad (2)$$

On applique aux échantillons pairs un opérateur de prédiction et on soustrait le résultat aux échantillons impairs. Après cette opération élémentaire, que l'on appelle 'pas primal' ou Predict, on obtient donc une erreur de prédiction :

$$d = x_0 - P(x_e) \quad (3)$$

Le pas dual du schéma lifting, ou Update, consiste à effectuer une mise à jour du signal pair avec le signal différentiel calculé. Le signal mis à jour s'exprime alors :

$$s = x_e + U(d) \quad (4)$$

Plusieurs enchaînements de pas primaux et duaux P_i et U_i peuvent être nécessaires. On obtient alors des signaux intermédiaires d_i et s_i . Au final, on obtient un signal filtré passe-bas s et un signal filtré passe-haut d . Le grand intérêt pratique de cette méthode réside dans la simplicité de l'inversion de la transformée qui consiste à changer les additions en soustractions :

$$x_e = s - U(d) \quad (5)$$

$$x_0 = d + P(x_e) \quad (6)$$

Le schéma lifting permet de réduire le coût calculatoire par rapport à l'implantation par convolution. Son second avantage est sa capacité à produire des coefficients d'ondelettes entiers [6]. Cette fonctionnalité est essentielle pour pouvoir réaliser des

méthodes de compression sans perte ou allant d'un schéma avec pertes vers du sans perte. La méthode présentée ci-dessus permet de calculer les coefficients d'une transformée en ondelettes 1D à valeurs entières. La transformée en ondelettes 1D a été successivement étendue aux images 2D, 3D et 4D en l'appliquant dans chaque direction.

Nous utilisons une transformée en ondelettes dyadique 3D réalisée par le schéma lifting sur deux niveaux de résolution. Celle-ci consiste à appliquée une transformée en ondelettes 1D sur chacun des trois axes x, y, z, (colonnes, lignes, inter coupes). La réalisation d'une transformée en ondelettes 3D à l'avantage de prendre en compte la corrélation entre les coupes, ce qui la rend plus efficace que d'effectuer une transformée en ondelettes 2D sur chaque image du volume. A la fin de la décomposition, 15 sous-bandes volumiques de résolution inférieure sont obtenus : l'image 3D basse-fréquence LLL et les sous-bandes volumiques 3D de détails (Fig. 2). Les sous-bandes obtenues sont codées à l'aide du SPIHT3D.

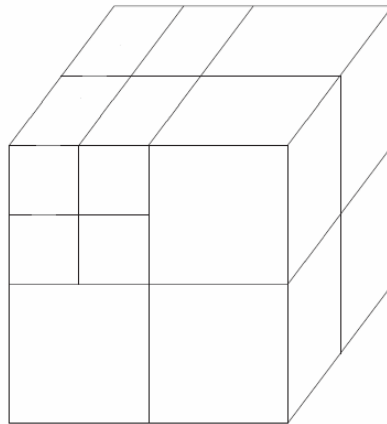


Fig. 2. Transformée en ondelettes 3D dyadique sur 2 niveaux

Etape 3 :

L'application de la transformée en ondelettes fait apparaître à différents niveaux d'échelles des dépendances entre les représentations de l'image (Fig. 3), qui peuvent être exploiter en compression. C'est cette propriété qui est à la base de l'algorithme SPIHT. L'algorithme SPIHT 2D (Set Partitioning in Hierarchical Trees) utilisé pour la compression des images 2D, exploite les corrélations entre sous-bandes aussi bien entre deux niveaux consécutifs de résolution (relations père-fils) qu'entre sous-bandes plus éloignées (relations de petits-enfants, arrières petits-enfants...). On isole ainsi des zones vastes de coefficients non significatifs, c'est ce qui permet d'atteindre de bonnes performances dans la compression. L'algorithme SPIHT a été étendu au cas 3D pour la vidéo [7] et pour la compression d'images volumiques en utilisant une transformée en ondelettes 3D. Les principes de base de SPIHT 3D sont identiques à sa

version 2D (exploite les dépendances inter-bandes dans trois dimensions au lieu de deux).

Les arbres utilisés par le SPIHT 3D considèrent que chaque nœud ne possède aucun descendant (les feuilles) ou bien huit descendants qui forment un groupe de $2 \times 2 \times 2$. Les coefficients de la sous-bande basse fréquence sont les racines des arbres et ils sont aussi regroupés par huit cependant un des huit coefficients n'a pas de descendants alors que les sept autres possèdent chacun huit descendants.

Les coefficients qui ne possèdent pas de descendants sont identifiés par leurs 3 coordonnées impaires. Aussi deux types d'arbres de zéros sont considérés, le premier (type A) consiste en une racine ayant tous ses descendants à 0. Le second type d'arbre (type B) contient uniquement les petits-enfants, arrière petits-enfants ... de la racine.

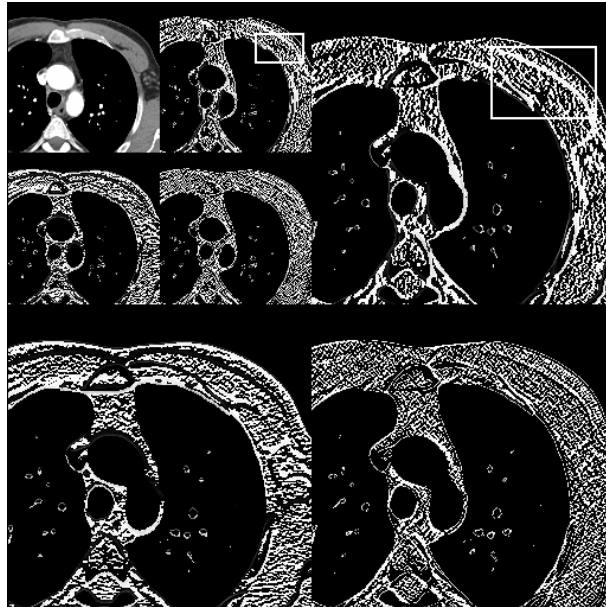


Fig. 3. Illustration des dépendances inter-échelles

Pour le codage SPIHT 3D utilise 3 ensembles de coordonnées :

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

Les règles de partitions sont les suivantes :

1. La partition initiale est formée des ensembles $\{(i, j, k)\}$ et $D(i, j, k)$, pour tous les nœuds de la basse fréquence (i, j, k) qui ont un descendant.
2. Si $D(i, j, k)$ est significatif alors il est partitionné en $L(i, j, k)$ plus 8 ensembles composé d'un seul élément avec $(l, m, n) \in O(i, j, k)$.

3. Si $L(i, j, k)$ est significatif alors il est partitionné en 8 sous-ensembles $D(l, m, n)$ avec $(l, m, n) \in O(i, j, k)$.

Trois listes sont utilisées pour stocker les informations de signification :

La Liste des Coefficients Significatifs (LCS),

La Liste des Coefficients Non significatifs (LCN),

La Liste des Ensembles Non significatifs (LEN).

Dans toutes les listes chaque entrée est identifiée par les coordonnées (i, j, k) . Ces derniers représentent dans les listes LCS et LCN des coefficients individuels et dans LEN soit l'ensemble $D(i, j, k)$ ou $L(i, j, k)$.

Pendant la passe de signification, les coefficients dans LCN, qui étaient non significatifs dans la passe précédente sont testés. Ceux qui deviennent significatifs sont mis dans LCS. Similairement, les ensembles de LEN sont évalués dans leur ordre d'entrée, et quand un ensemble est trouvé significatif il est supprimé de cette liste puis est partitionné. Les nouveaux ensembles avec plus d'un élément sont ajoutés à la fin de LEN avec le type (A ou B), alors que les simples coefficients sont ajoutés à la fin de LCS ou LCN suivant leur signification. La liste LCS contient les coordonnées des coefficients qui seront visités dans la prochaine passe de raffinement.

L'algorithme de codage est défini comme suit :

1. Initialisation

Sortie $n = \lceil \log_2(C_{\max}) \rceil$ où C_{\max} désigne le maximum de la valeur absolue des coefficients, le seuil $T_n = 2^{\lceil \log_2(C_{\max}) \rceil}$
 LCS est vide, $LCN = \{(i, j, k)\}$ cet ensemble représente les nœuds de la basse fréquence. LEN contient les mêmes coefficients que LCN excepté ceux qui n'ont pas de descendants.

Pour déterminer la signification d'un sous-ensemble E pour un seuil donné T on utilise l'opérateur de signification s_{T_n} tel que

$$s_{T_n}(E) = \begin{cases} 1 & \text{si } \exists c \in E : |c| \geq T_n \\ 0 & \text{si } \forall c \in E : |c| < T_n \end{cases}$$

2. Passe de signification

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

2.1.1 Sortie $s_{T_n}(i, j, k)$

2.1.2 Si $s_{T_n}(i, j, k) = 1$ alors mettre (i, j, k) dans LCS et coder le signe de $c(i, j, k)$

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

2.2.1 Si l'entrée est de type A

a Sortie $s_{T_n}(D(i, j, k))$

b si $s_{T_n}(D(i, j, k)) = 1$ alors

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

- Sortie $s_{T_n}(l, m, n)$
- Si $s_{T_n}(l, m, n) = 1$ alors mettre (l, m, n) dans LCS et coder le signe de $c(l, m, n)$
- Si $s_{T_n}(l, m, n) = 0$ alors mettre (l, m, n) à la fin de LCN
- Si $L(i, j, k) \neq \emptyset$ alors mettre (l, m, n) à la fin de LEN comme une entrée de type B

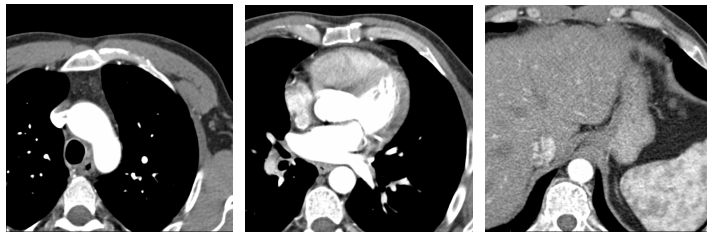
2.2.2 Si l'entrée est de type B

- a Sortie $s_{T_n}(L(i, j, k))$
- b Si $s_{T_n}(L(i, j, k)) = 1$ alors
 - mettre (l, m, n) à la fin de LEN comme une entrée de type A
 - supprimer (i, j, k) de LEN

3. Passe de raffinement : Pour chaque coefficient $(i, j, k) \in \text{LCS}$ à l'exception de ceux incluse dans la même passe de signification (c.a.d pour le même n). Sortie le n -ième bit significatif de $|c(i, j, k)|$

4. Modification du pas de quantification : $T_n \cdot T_n / 2$ et aller à l'étape 2

Pour obtenir, l'algorithme de décodage, il suffit simplement de remplacer le mot Sortie par Entrée dans l'algorithme précédent. Le codeur SPIHT 3D utilise un codage par plan de bits, chaque plan de bits est composé des bits de signification s_{T_n} de tous les coefficients pour un seuil donné. Aussi, il produit directement des symboles binaires, qui sont codés à l'aide d'un codeur arithmétique afin d'améliorer les performances de codage.



Coupe 1

Coupe 38

Coupe 76

Fig. 4. Exemple de coupes 2D issues de l'image médicale 4D du teste

3 Résultats

Les tableaux 1, 2 et 3 résument les résultats du codage sans perte en utilisant la méthode proposée sur une image médicale 4D CT d'un cœur battant composée de 10 volumes, chaque volume contient 76 images et la taille de chaque image est de 512×512 codée sur 8bits par pixels. Pour le calcul du PSNR3D de chaque volume et le PSNR2D des coupes 2D nous avons utilisé les formules suivantes :

Le PSNR3D est défini comme suit :

$$\text{PSNR3D} = 10 \log_{10} \frac{255^2 \times M \times N \times P}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \sum_{k=0}^{P-1} (f(i, j, k) - \hat{f}(i, j, k))^2} \quad (7)$$

$$\text{PSNR2D} = 10 \log_{10} \frac{255^2 \times M \times N}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (f(i, j, k) - \hat{f}(i, j, k))^2} \quad (8)$$

Où f et \hat{f} représentent l'image originale et l'image reconstruite respectivement de taille égale à $M \times N \times P$ pour le PSNR3D et de taille égale à $M \times N$ pour le PSNR2D.

Nous avons réalisé la compression en utilisant trois filtres différents. Les résultats obtenus montrent que c'est l'utilisation du filtre de Haar qui donne la meilleure qualité d'image mais en terme de taux de compression c'est le filtre 5/3 qui donne le meilleur taux 5,7 mais la qualité de l'image est moins bonne que les autres filtres. Avec le filtre 9/7 nous obtenons une image avec une bonne qualité (meilleure que 5/3) et avec un taux 5,44. Les techniques classiques de compression des images médicales 4D tel que JPEG2000 JPEG-LS et JPEG2000-3D atteignent des taux de compression variants entre 2 et 3. Avec notre technique nous avons atteint un taux de 5,7.

4 Conclusion

Nous venons de décrire une technique de compression d'images médicales 4D, cette technique est sans perte et permet d'atteindre un taux de compression allant jusqu'à 5,7. Les techniques de compression avec compensation de mouvement donnent de meilleurs résultats par rapport aux techniques classiques car elles exploitent les redondances dans les quatre dimensions donc l'intégration de la compensation de mouvement nous permettra d'atteindre de meilleurs taux.

Tableau 1. PSNR2D (dB) obtenu par la coupe 76 de chaque'un des dix volumes en utilisant les différents filtres ondelettes

| N°Volume | Haar | 9/7 | 5/3 |
|----------|-------|-------|-------|
| 01 | 46,70 | 29,01 | 18,65 |
| 02 | 46,74 | 29,02 | 18,62 |
| 03 | 46,68 | 29,02 | 18,63 |
| 04 | 46,69 | 29,01 | 18,65 |
| 05 | 46,71 | 29,03 | 18,66 |
| 06 | 46,72 | 29,09 | 18,67 |
| 07 | 46,71 | 29,12 | 18,68 |
| 08 | 46,73 | 29,03 | 18,66 |
| 09 | 46,75 | 28,99 | 18,67 |
| 10 | 46,71 | 29,01 | 18,64 |

Tableau 2. PSNR3D (dB) des dix volumes obtenu en utilisant les différents filtres ondelettes

| N°Volume | Haar | 9/7 | 5/3 |
|----------|-------|-------|-------|
| 01 | 47,85 | 31,75 | 22,13 |
| 02 | 47,87 | 31,77 | 22,18 |
| 03 | 47,85 | 31,78 | 22,17 |
| 04 | 47,87 | 31,78 | 22,19 |
| 05 | 47,84 | 31,77 | 22,16 |
| 06 | 47,86 | 31,77 | 22,15 |
| 07 | 47,86 | 31,75 | 22,14 |
| 08 | 47,85 | 31,74 | 22,13 |
| 09 | 47,84 | 31,75 | 22,13 |
| 10 | 47,86 | 31,75 | 22,14 |

Tableau 3. Taux de compression obtenu de chaque volume par les différents filtres ondelettes

| N°Volume | Haar | 9/7 | 5/3 |
|----------|------|------|------|
| 01 | 2,95 | 5,42 | 5,68 |
| 02 | 2,96 | 5,41 | 5,67 |
| 03 | 2,96 | 5,41 | 5,67 |
| 04 | 2,96 | 5,39 | 5,67 |
| 05 | 2,95 | 5,41 | 5,67 |
| 06 | 2,96 | 5,44 | 5,70 |
| 07 | 2,96 | 5,44 | 5,70 |
| 08 | 2,95 | 5,42 | 5,67 |
| 09 | 2,95 | 5,42 | 5,67 |
| 10 | 2,96 | 5,42 | 5,68 |

Références

1. H.G. Lalgudi, A. Bilgin, M.W. Marcellin, A. Tabesh, M.S. Nadar, and T.P. Trouard, "Four-dimensional compression of fMRI using JPEG2000," in *Proc. SPIE International Symposium on Medical Imaging*, Feb. 2005.
2. E. Siegel, K. Siddiqui and J. Johnson, "Compression of multislice CT: 2D vs. 3D JPEG2000 and effects of slice thickness", *Proceedings of SPIE Int. Soc. Opt. Eng.*, vol. 5748, pp. 162-170, 2005.
3. Kassim, A. A., Yan, P., Lee, W. S., Sengupta K.: Motion Compensated Lossy-to-Lossless Compression of 4D Medical Images Using Integer Wavelet Transforms. *IEEE Trans. Information Technology In Biomedicine*, Vol. 9, no. 1 (2005) 132–138
4. Yan, P., Kassim, A. A.: Lossless and near-lossless motion compensated 4D medical images compression. In *Proceedings of IEEE International Workshop on BioMedical Circuits & Systems*, Singapore (2004)
5. Sanchez, V., Nasiopoulos, P. Abugharbieh, R.: Lossless Compression of 4D Medical Images using H.264/AVC. *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Toulouse France (2006) II 1116-1119
6. I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 1998.
7. B.J. Kim and W.A. Pearlman. An Embedded Wavelet Video Coder Using Three-Dimensional Set Partitioning in Hierarchical Trees (SPIHT). In *IEEE Data Compression Conference DCC'97*, pages 221-260, 1997.