

## Compression de séquences d'images basée sur la région d'intérêt

D. Samai<sup>(1)</sup>, N. Doghmane<sup>(2)</sup>, M. Bedda<sup>(3)</sup> et L. Lucas<sup>(4)</sup>

<sup>(1)</sup> *Département de Génie Electrique, Université Kasdi Merbah, Ouargla, Algérie*

<sup>(2)</sup> *Département d'Electronique, Université Badji Mokhtar, Annaba, Algérie*

<sup>(3)</sup> *Faculté d'Engineering, Université d'Al-Jouf, Arabie-Saoudite*

<sup>(4)</sup> *Université Champagne-Ardenne, CReSTIC-SIC, Reims, France*

[Samai.djamel@gmail.com](mailto:Samai.djamel@gmail.com), [ndoghmane@univ-ananba.org](mailto:ndoghmane@univ-ananba.org), [mouldi\\_bedda@yahoo.fr](mailto:mouldi_bedda@yahoo.fr), [laurent.lucas@univ-reims.fr](mailto:laurent.lucas@univ-reims.fr)

**Résumé**— Ce travail propose un schéma de compression progressive par région d'intérêt pour les séquences d'images. Il combine entre l'algorithme SPIHT-3D et la SA-DCT-3D au lieu de la SA-DWT-3D pour réaliser cette tâche. L'ensemble du codeur allie également une faible complexité avec de très bonnes performances.

**Mots clefs**— Codage progressif, région d'intérêt, Shape-adaptive, SPIHT-3D, DCT-3D.

### I. INTRODUCTION

LA compression visuelle a attiré considérablement l'attention de la communauté de traitement d'images. C'est dû à la liste croissante d'applications telles que la vidéoconférence, les multimédias, la vidéo au-dessus des lignes téléphoniques, les communications sans fil, et la vidéo au-dessus de l'Internet. Elle s'impose comme une étape incontournable pour optimiser l'augmentation accrue de ces données. Une série de méthodes de compression existent actuellement dans le domaine visuel de technologie, telles que l'EZW (Embedded Zerotree of Wavelet) [1], le SPIHT (Set Partitioning In Hierarchical Trees) [2] et le JPEG2000 [3]-[5]. Ensuite ces méthodes ont été étendues au 3D [6]-[10] et ont donné des résultats comparables mêmes meilleurs que des algorithmes utilisant la compensation de mouvement tel que le MPEG2 [11] [12] avec une modeste complexité. Parallèlement, de nouvelles applications surgissent le besoin des fonctionnalités supplémentaire et des exigences pour les systèmes de compression d'images et de vidéo. La plupart d'entre elles sont étroitement liées à des manipulations orientées-objet. L'une de ces fonctionnalités est la compression ou le codage de séquences d'images par région d'intérêt (ROI coding : Region Of Interest coding) [13]-[17]. Une solution efficace à la compression de séquences d'images par région d'intérêt a été proposée par W. Pearlman [13]. Elle se base sur la transformation en ondelettes discrète de forme ou de contexte adaptable 3D (Shape-Adaptive Discrete Wavelet Transform 3D : SA-DWT-3D) appliquée à l'algorithme SPIHT-3D. Dans cette optique, nous avons introduit la SA-DCT-3D (Shape-Adaptive Discrete Cosine Transform 3D) dans l'algorithme de compression progressive (SPIHT-3D) ainsi que d'utiliser la fonctionnalité de compression par région d'intérêt pour

cette transformation. Notre objectif est l'obtention d'un codeur progressif de séquences d'images avec une complexité réduite et une efficacité en termes de débit-distorsion. Les résultats de simulation montrent que notre méthode a bien permis la combinaison de l'algorithme pyramidal SPIHT-3D avec la SA-DCT-3D tout en produisant une technique de compression de séquences d'images par région d'intérêt efficace.

### II. SHAPE-ADAPTIVE DCT-3D

La SA-DCT-3D se calcule sur les objets cubiques de volume  $M \times N \times N$  pixels où on divise chaque image (frame) de la séquence en blocs de  $N \times N$  pixels (figure 1).

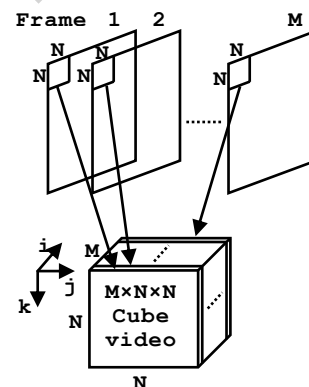


Figure 1 : Formation des cubes  $M \times N \times N$ .

La SA-DCT-3D peut être décomposée en une SA-DCT-2D spatiale et une SA-DCT-1D temporelle. Donc, chaque image de la séquence subit une SA-DCT-2D en premier lieu. Ensuite chaque coefficient obtenu dans une image devait subir une SA-DCT-1D avec ces homologues dans les autres images. Le concept de base de la SA-DCT-2D est illustré dans l'exemple de la figure 2. La figure 2-a montre un objet (noir) arbitrairement réparti sur un arrière-plan (blanc) en forme du segment d'image dans un bloc de référence  $8 \times 8$ . La première démarche consiste à effectuer la SA-DCT-1D horizontale de l'objet, où la longueur de chaque ligne  $i$  ( $0 \leq i \leq 7$ ) du segment de l'avant-plan est calculée. Les lignes sont décalées et alignées sur la bordure gauche du bloc de référence  $8 \times 8$  (Figure 2-b). Dépendant de la taille de chaque segment, les coefficients de la DCT-1D sont déterminés par l'équation suivante :

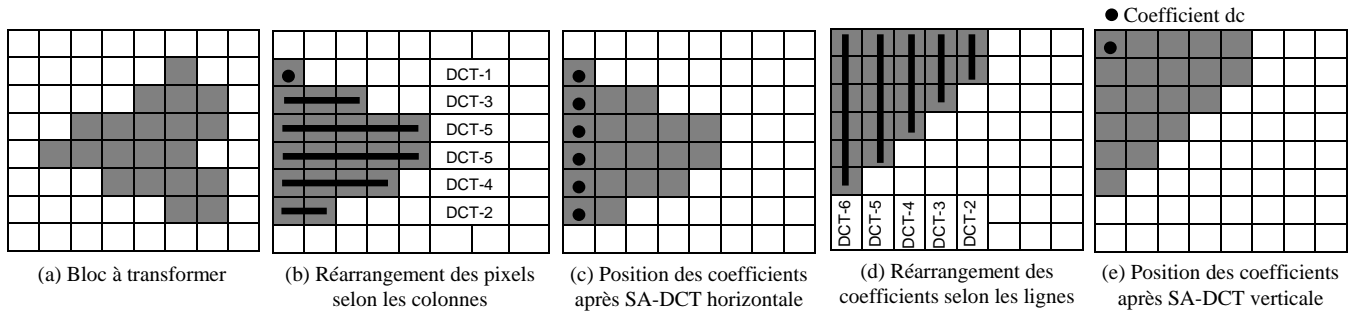


Figure 2 : Etapes successive pour le calcul d'une SA-DCT-2D.

$$X(u) = \sqrt{2/N} \cdot C(u) \cdot \sum_{i=0}^{N-1} x(i) \cdot \cos((2i+1)u\pi)/2N \quad (1)$$

Où  $C(i) = 1/\sqrt{2}$  pour  $i = 0$  et  $C(i) = 1$  pour  $i > 0$

$x(i)$  : Valeur d'un pixel,  $X(u)$  : Coefficient de la DCT

La composante continue ou la valeur moyenne de chaque ligne est située à l'extrémité gauche de la ligne (Figure 2-c). La deuxième démarche effectuée la SA-DCT-1D verticale (Figure 2-d). Le calcul des coefficients de la DCT-1D verticale adaptée à la taille de chaque colonne se fait toujours par l'équation (1) après un décalage vers le haut de ces coefficients. La figure 1-e montre les positions finales des coefficients de la DCT du bloc image  $8 \times 8$ . Le nombre des coefficients de la DCT est identique au nombre des pixels du segment d'image. Ainsi, la valeur moyenne de l'objet est située dans le coin gauche en haut du bloc  $8 \times 8$  de la DCT. Le reste des coefficients sont concentrés autour de la valeur moyenne. Une fois la SA-DCT-2D est terminée, une SA-DCT-1D est effectuée dans l'axe temporel. Pour cela, un décalage des coefficients se fait pour chaque image vers le début de l'axe temporel suivi par un calcul de la DCT-1D adapté à la taille de chaque segment.

La SA-DCT-3D n'a pas l'aspect pyramidal pour qu'elle soit appliquée directement à l'algorithme SPIHT-3D. En conséquence, il devait être nécessaire de modifier l'organisation des coefficients de la DCT-3D dans le but d'obtenir la propriété de la multi-résolution afin de l'introduire dans un algorithme de compression progressif. Pour cela, nous proposons d'ajouter trois étapes de décalage à la SA-DCT-3D. Un décalage temporel dans le sens contraire de décalage de la SA-DCT-1D temporelle suivi par un décalage des coefficients de chaque image vers le bas et un autre vers la droite. La figure 3 illustre les deux dernières étapes pour un bloc  $8 \times 8$  coefficients. Cette opération de décalage est nécessaire pour avoir les mêmes positions des pixels originaux où elle nous aide par la suite de faire une redistribution des coefficients de la DCT-3D pour avoir la structure des objets transformés en multi-résolution. En effet, les coefficients résultants sont redistribués par la même manière que celle proposée dans [18]. Elle nous permet de manipuler la DCT-3D comme la DWT-3D.

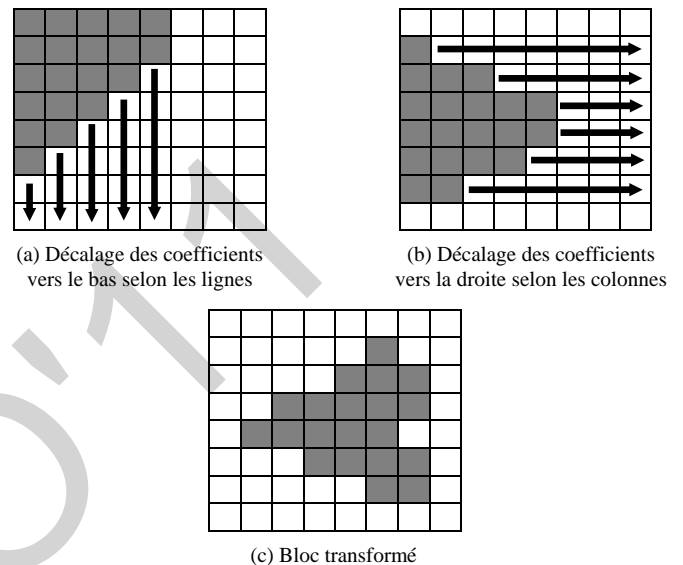


Fig. 3 Décalages lignes/colonnes des coefficients de la SA-DCT-2D.

La figure 4 montre un exemple du premier cube DCT-3D  $8 \times 8 \times 8$  d'une séquence d'images où ses coefficients sont redistribués dans le cube global de la DCT-3D de même volume que la séquence à coder. Les autres cubes seront traités de la même manière en respectant leurs rangs dans le cube global. De même, les masques des objets des différentes images de la séquence subiront la même redistribution précédente.

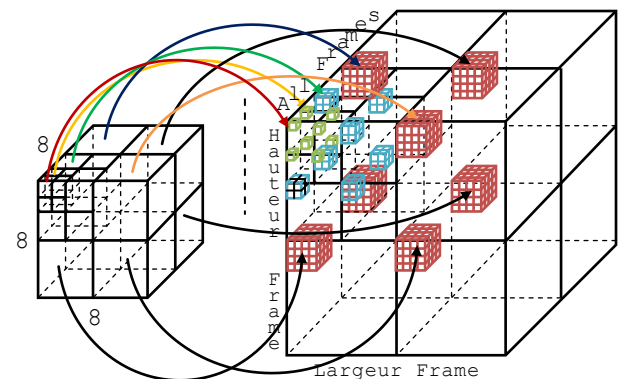


Figure 4 : Redistribution des coefficients d'un cube de la DCT-3D dans le cube global.

### III. CODAGE SPIHT-DCT-3D PAR REGION D'INTERET

La méthode de compression de séquences d'images par région d'intérêt utilisée dans cet article combine l'algorithme SPIHT-3D avec la SA-DCT-3D. Pour réaliser la tâche de compression par région d'intérêt, notre méthode utilise une des deux méthodes de décalage des coefficients (Max-Shift et mise à l'échelle) utilisées par l'algorithme JPEG2000 [19]-[21] dans le cas 2D. La figure 5 illustre le processus de codage d'une séquence d'images par SPIHT-ROI-3D en utilisant la compression de forme adaptable.

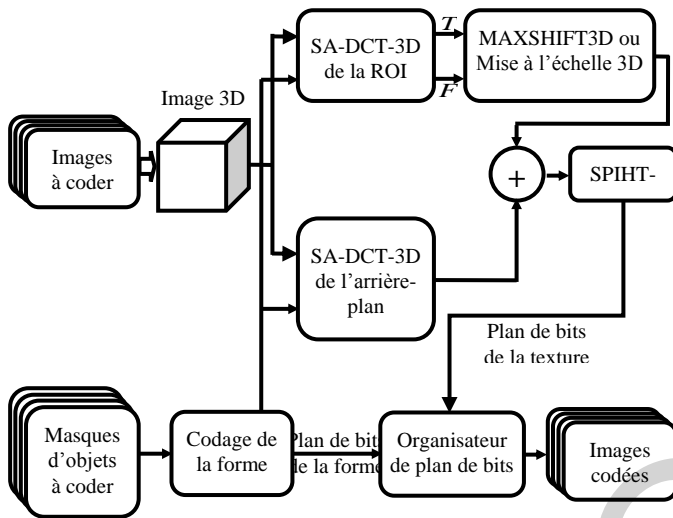
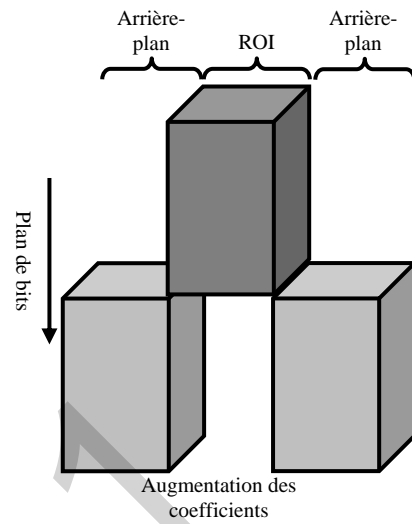


Figure 5 : Schéma de principe du codage d'une séquence d'images par SPIHT-ROI-3D

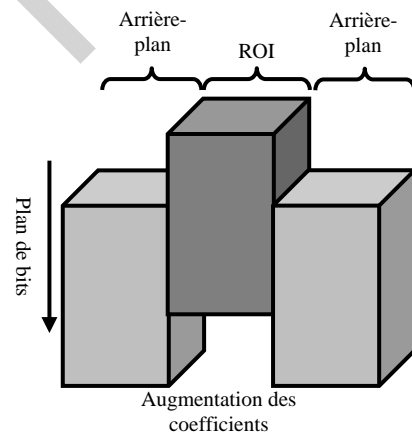
Premièrement, les coefficients appartenant à l'objet et à l'arrière-plan sont transformés avec la SA-DCT-3D en fonction de l'information de la forme (masque) séparément. Ensuite, les coefficients de la région d'intérêt sont pondérés ou multipliés par un facteur de  $2^5$  par l'une des deux méthodes, soit par la méthode Max-Shift, où tous les coefficients pondérés sont supérieur aux coefficients de l'arrière-plan, soit par la méthode de mise à l'échelle, où la pondération des coefficients est contrôlée (figure 6). Donc, les plans de bits des coefficients sont codés plan par plan afin d'avoir une qualité plus élevée à la région d'intérêt que l'arrière-plan. Dans la méthode de mise à l'échelle, l'information de la forme doit être transmise au décodeur contrairement à la méthode de Max-Shift.

La procédure de la compression s'achève par l'application de l'algorithme SPIHT-3D [7] [8] à l'ensemble des coefficients (région d'intérêt et arrière-plan). Cet ensemble est représenté comme une collection d'arbres d'orientation spatio-temporelle où chaque arbre est constitué des coefficients de toutes les sous-bandes qui correspondent à la même orientation spatiale. Un des coefficients (marqué par "\*" ) ne possède pas de descendants, les sept autres possèdent chacun 8 descendants ( $2 \times 2 \times 2$  coefficients) à des emplacements correspondants au même niveau. Chaque

coefficient de ces 8 descendants possède aussi 8 descendants dans le niveau inférieur de la pyramide. La figure 7 représente les arbres d'orientation spatiale, dans le cas d'une décomposition 3D dyadique à deux niveaux.



(a) Méthode de Max-Shift



(a) Méthode de mise à l'échelle

Fig. 6 Pondération des coefficients transformés.

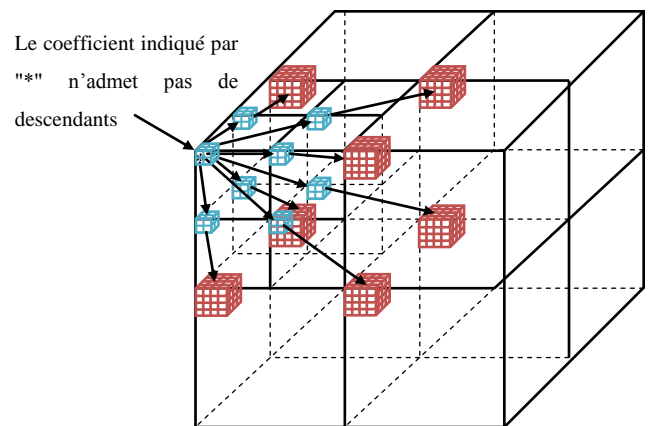


Fig. 7 Arbres d'orientation spatio-temporelle.

On peut également mentionner l'utilisation des arbres à orientation spatio-temporelle non équilibrée ou non dyadique. Cela permet d'appliquer facilement un nombre différent de décompositions entre les dimensions spatiales et temporelles, qui est une fonctionnalité souhaitable quand on code un ensemble d'images limité en taille.

Le SPIHT-3D utilise deux types d'arbres de zéros. Le premier type d'arbre (type A) consiste en une simple racine ayant tous ses descendants à 0 pour un plan de bits donné. En fait, bien que l'arbre de zéros soit spécifié par les coordonnées de la racine, la racine n'est pas incluse dans l'arbre. Le second type d'arbre (type B) est similaire, mais exclut les huit descendants de la racine. Les arbres de type B contiennent uniquement les petits descendants, arrières petits descendants de la racine. Il utilise aussi les ensembles suivants :

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

Les règles de partition sont les suivantes :

- La partition initiale est formée des ensembles  $\{(i,j,k)\}$  et  $D(i,j,k)$ , pour tous  $(i,j,k) \in LLL$  (sous-bande basse fréquences du niveau le plus haut) qui ont un descendant.
- Si  $D(i,j,k)$  est significatif alors il est découpé en  $L(l,m,n)$  plus 8 ensembles d'un seul élément avec  $(l,m,n) \in O(i,j,k)$ .
- Si  $L(i,j,k)$  est significatif alors il est partitionné en 8 sous-ensembles  $D(l,m,n)$  avec  $(l,m,n) \in O(i,j,k)$ .

L'algorithme SPIHT-3D stocke l'information de signification dans trois listes en trois dimensions ordonnées, on trouve la liste des coefficients significatifs (LSP : List of Significant Pixels), la liste des coefficients non significatifs (LIP : List of Insignificant Pixels) et la liste des ensembles non significatifs (LIS : List of Insignificant Sets). Dans chaque liste, l'entrée est représentée par un triplé de coordonnées  $(i,j,k)$ . Il est représenté par des coefficients individuels dans la LIP et dans la LSP et par des ensembles dans la LIS soit  $D(i,j,k)$  (type A), soit  $L(i,j,k)$  (type B). L'algorithme SPIHT-3D opère sur deux étapes ou deux passes, la passe de signification et la passe de raffinement. Pendant la passe de signification, les coefficients dans la LIP, qui étaient non significatifs dans la passe précédente sont testés. Ceux qui deviennent significatifs sont mis dans la LSP. Similairement, les ensembles de la LIS sont évalués dans leur ordre d'entrée, et quand un ensemble est trouvé significatif il est supprimé de cette liste puis il est partitionné. Les nouveaux ensembles avec plus d'un élément sont ajoutés à la fin de la LIS avec le type (A ou B), alors que les simples coefficients sont ajoutés à la fin de la LSP ou la LIP suivant leur signification. La passe de raffinement consiste à coder les bits les plus significatifs des coefficients significatifs de la LSP.

#### IV. RESULTATS ET DISCUSSION

Pour réaliser la compression de séquences d'images par région d'intérêt, nous avons implémenté un programme basé sur la compression de forme adaptable 3D en utilisant la DCT. L'application de la compression a été faite sur deux séquences de 16 images ("Pingpong" et "Walter") et de taille  $256 \times 256$  pixels pour chaque image. La segmentation des masques a été réalisée manuellement par un logiciel de retouche d'images et ils sont codés par un codage RLE [22]. La qualité des images reconstruites est mesurée par le rapport signal sur bruit crête (PSNR en dB) défini comme suit :

$$PSNR(dB) = 10 \log_{10} 255^2 / MSE \quad (2)$$

Avec

$$MSE = 1/M \times N \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [\hat{x}(m,n) - x(m,n)]^2 \quad (3)$$

MSE : représente l'erreur quadratique moyenne entre l'image originale et l'image reconstruite (décodée).

Ainsi le PSNR global pour une séquence est la moyenne des PSNR obtenus pour toutes les images reconstruites. Cette mesure est prise qu'à l'intérieur des objets et cela pour bien montrer l'efficacité de la compression par région d'intérêt.

Le tableau 1 ainsi que la figure 8 résument les résultats objectifs obtenus par notre méthode de compression pour une décomposition non dyadique (niveau de décomposition spatiale égale à 5 et niveau de décomposition temporelle égale à 3). Les résultats subjectifs sont représentés par les figures 9 et 10 pour le même cas non dyadique avec un taux de bits de 0,5 bpp.

Tableau 1 : Résultats distorsion-débit du SPIHT-ROI-SA-DCT-3D et SPIHT-DCT-3D pour deux séquences d'images.

|                       |                    | PSNR moyen (dB)     |       |              |       |
|-----------------------|--------------------|---------------------|-------|--------------|-------|
|                       |                    | SPIHT-ROI-SA-DCT-3D |       | SPIHT-DCT-3D |       |
| Nbre de decalage (ND) |                    | 4                   | MaxS  |              |       |
| Pingpong              | Taux de bits (bpp) | 0,125               | 30,58 | 31,42        | 24,36 |
|                       |                    | 0,25                | 36,51 | 37,53        | 26,30 |
|                       |                    | 0,50                | 44,08 | 48,85        | 29,07 |
|                       |                    | 0,75                | 48,33 | 71,35        | 31,97 |
|                       |                    | 1,00                | 52,29 | $\infty$     | 34,34 |
| Walter                | Taux de bits (bpp) | 0,125               | 33,12 | 36,67        | 35,33 |
|                       |                    | 0,25                | 38,79 | 40,67        | 39,18 |
|                       |                    | 0,5                 | 44,47 | 45,74        | 43,37 |
|                       |                    | 0,75                | 48,26 | 49,15        | 45,97 |
|                       |                    | 1,00                | 50,97 | 52,04        | 47,99 |

D'après les résultats obtenus, nous remarquons que notre méthode de codage utilisée a permis d'augmenter le PSNR dans les régions d'intérêt et spécialement lorsqu'on utilise la méthode Max-shift qui n'a pas besoins de coder les masques. Ainsi, pour la méthode de mise à l'échelle, le cout de bits de codage des masques est 0,0373 bpp pour la séquence "Pingpong" et 0,0639 bpp pour la séquence "Walter". Ce cout influe sur l'opération de la compression pour les taux de bits faibles.

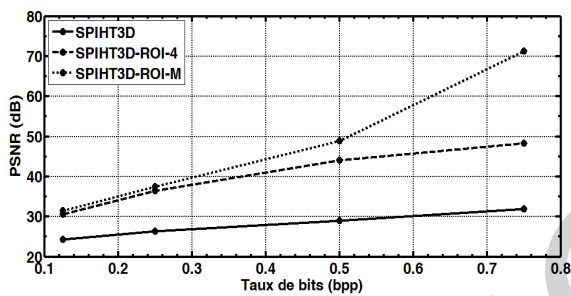
La redistribution des coefficients de la DCT-3D en plusieurs niveaux nous a permis aussi d'obtenir une architecture pyramidale similaire à celle obtenue par les ondelettes avec une complexité réduite de point de vue implémentation afin de compresser progressivement les séquences d'images en utilisant l'algorithme emboîté SPIHT-3D.

L'efficacité de cette fonctionnalité de compression par région d'intérêt est en fonction de plusieurs paramètres, parmi lesquels, on trouve : la taille des objets, le taux de bits

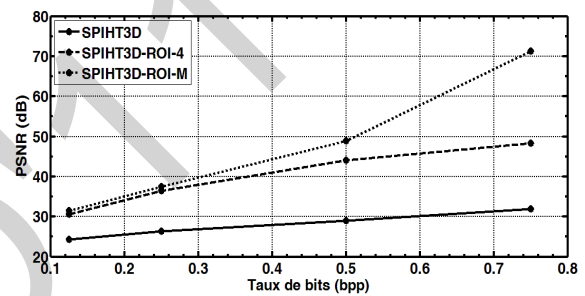
et la nature des séquences. En effet, le codage des objets pourrait se faire sans perte si ces derniers ont des tailles réduites avec un taux de bits assez élevé. Ainsi, les séquences d'images médicales s'avèrent parmi les applications les plus pertinentes pour ce type de codage.

### V.CONCLUSION

Dans ce travail, nous avons adapté l'algorithme SPIHT-3D avec la DCT-3D pour une technique de codage par région d'intérêt utilisant la compression de contexte adaptable. La méthode de compression utilisée permet d'avoir de très bons résultats de point de vue qualité visuelle des images traités d'une part et taux de bits d'autre part. Les résultats obtenus dépendent aussi de la nature des séquences d'images ainsi que le taux de bits. Ces résultats révèlent que la compression de forme adaptable ou shape-adaptive peut être insérée dans les codeurs à région d'intérêt.



(a) Pingpong



(b) Walter

Fig. 8 Evolution de la distorsion en fonction de débit.

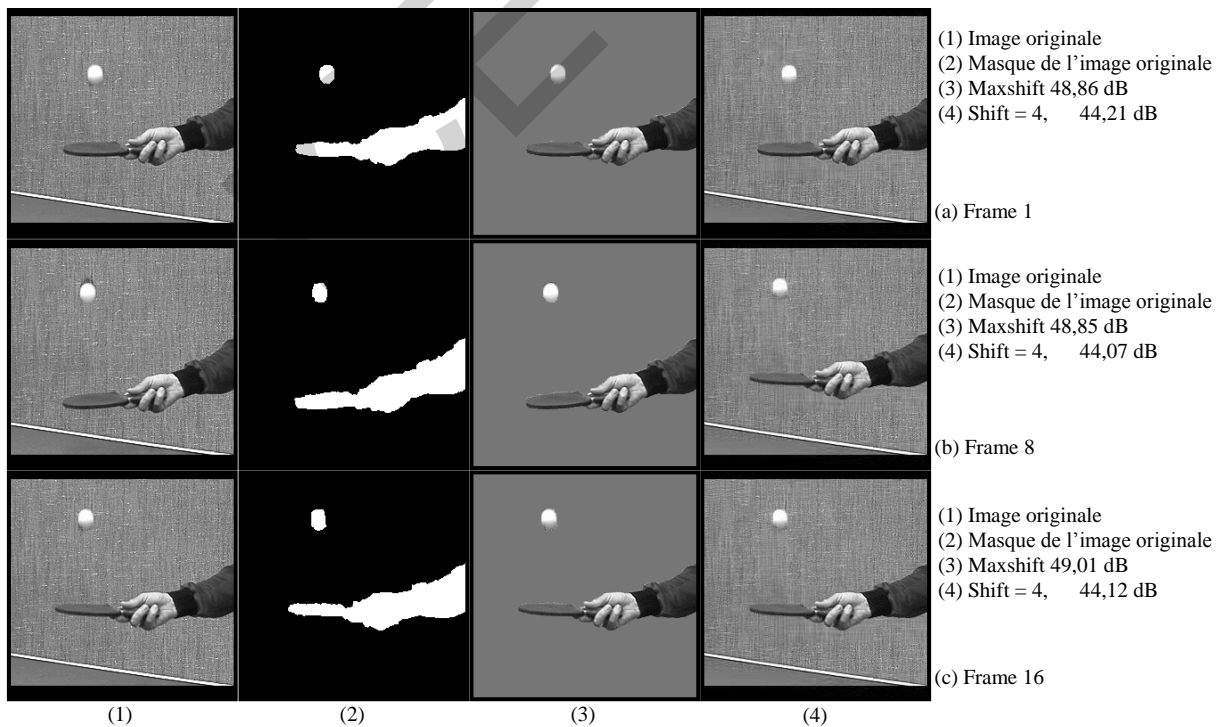


Fig. 9 Comparaison subjective de l'algorithme SPIHT-ROI-SA-DCT-3D pour la séquence d'images "Pingpong"

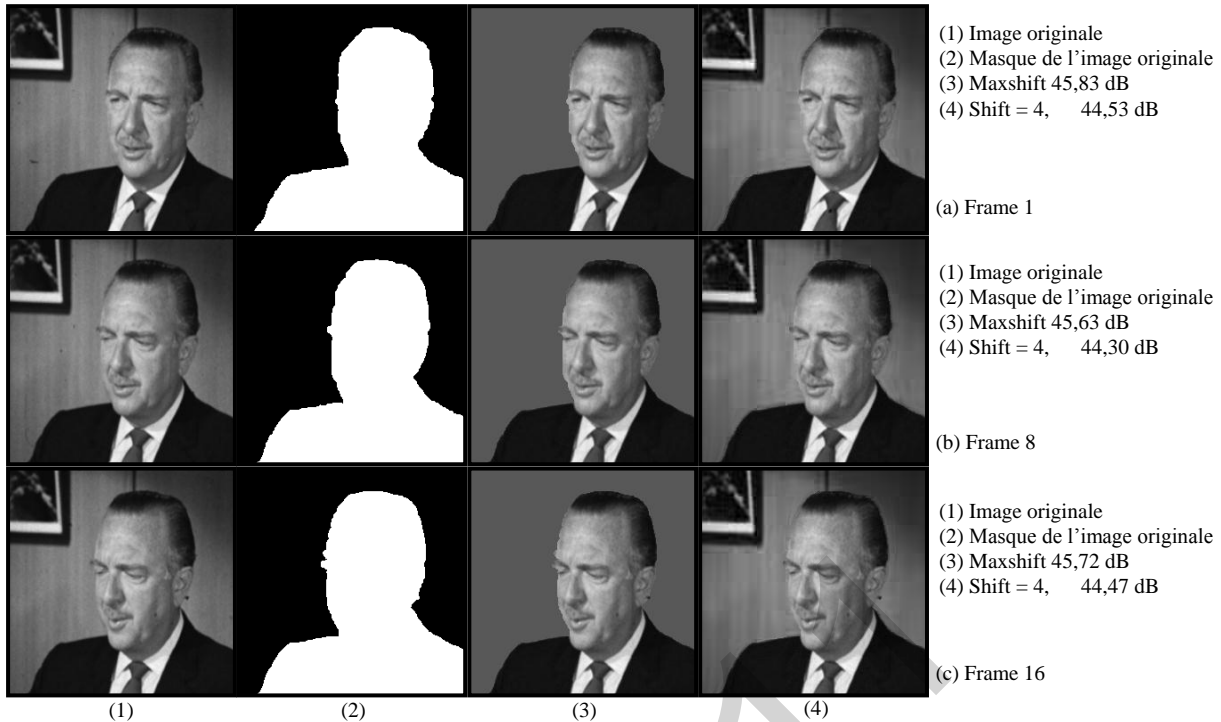


Fig. 10 Comparaison subjective de l'algorithme SPIHT-ROI-SA-DCT-3D pour la séquence d'images "Walter"

#### REFERENCES

- [1] J. M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445-3462, Dec. 1993.
- [2] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. On Circuits and Systems for Video Technology*, Vol. 6, pp. 243-250, 1996.
- [3] C. A. Christopoulos, A. Skodras, T. Ebrahimi, "The JPEG 2000 Still Image Coding System: An Overview," *IEEE Trans. on Consumer Electronics*, Vol. 46, No. 4, 1103-1127, Nov. 2000.
- [4] D. Taubman and M. Marcelin, "JPEG2000: Image compression fundamentals, standards and practice," Kluwer Academic Publishers, 795 pages, Boston November 2001.
- [5] ISO/IEC 15444-1:2004 "Information technology – jpeg 2000 image coding system : Core coding system," Published 15-04-2008.
- [6] Y. Chen and W.A. Pearlman, "Three-Dimensional Subband Coding of Video Using the Zerotree Method," *Visual Communications and Image Processing '96, Proc. SPIE 2727*, pp. 1302-1312, Mar. 1996.
- [7] B. J. Kim, and W. A. Pearlman, "An Embedded Wavelet Video Coder Using Three-Dimensional Set Partitioning in Hierarchical Trees (SPIHT)," in *Data Compression Conference, 1997. DCC '97. Proceedings*, pp. 251-260, 25-27 Mar 1997.
- [8] B. J. Kim, K. Z. Xiong and W. A. Pearlman, "Low bit-rate scalable video coding with 3D set partitioning in hierarchical trees (3D-SPIHT)," *IEEE Transactions on Circuits and Systems for Video Technology*, 10(8) :1374-1387, December 2000.
- [9] X. Tang, S. Cho, and W. A. Pearlman, "3D set partitioning coding methods in hyperspectral image compression," in *the Proceedings of IEEE International Conference on Image Processing*, pp. 239-242, September 2003.
- [10] J. J. Wu, Z. S. Wu, and C. K. Wu, "Lossy to lossless compressions of hyperspectral images using three-dimensional set partitioning algorithm," *SPIE Optical Engineering*, Vol. 45, No. 2, pp. 0270051-0270058, 2006.
- [11] ISO/IEC JTC1/SC29/WG11 N0702 Rev, "Information Technology – Generic Coding of Moving Pictures and Associated Audio, Recommendation H.262," Draft International Standard, Paris, 25 March 1994.
- [12] ISO/IEC 13818, "Information Technology – Generic Coding of Moving Pictures and Associated Audio," Second edition, 2000.
- [13] I. Ueno and W.A. Pearlman, "Region of interest coding in volumetric images with shape-adaptive wavelet transform," In *Proceedings of the SPIE/IS&T Electronic Imaging 2003*, 5022, pp. 1048-1055, 2003.
- [14] X. Tang and W.A. Pearlman, "Progressive resolution coding of hyperspectral images featuring region of interest access," *SPIE Defense & Security 2005, Visual Information Processing XIV, Proc. SPIE 5817*, pp.570-580, April 2005.
- [15] X. Tang and W.A. Pearlman, "Lossy-To-Lossless block-based compression of hyperspectral/volumetric data," in *the Proceedings of IEEE International Conference on Image Proc.*, pp. 1133-1136, 2006.
- [16] M. Jing, W. Chengke, C. Dong and Z. Youxi, "Embedded coding of medical images with regions of interest based on 3-D zerotree," *Higher Education Press, co-published with Springer-Verlag GmbH*, Volume 2, number 1, pp. 8-12, march 2007.
- [17] G. R. Suresh, S. Sudha and R. Sukanesh, "Region Based Compression of MRI Sequence Using Shape Adaptive Wavelet Transform Coding," *International Journal of Soft Computing*, Volume 3, Issue 1, pp. 33-37, 2008.
- [18] D. Samai, N. Doghmane and M. Bedda, "Set Partitioning Method Based 3D- DCT," *Séminaire sur les systèmes Numériques Embarqués (SSNE '08)*, École Militaire polytechnique (EMP), Bordj El Bahri-Algérie, 05-06-mai 2008.
- [19] D. Nister and C. Christopoulos, "Lossless Region of Interest with Embedded Wavelet Image Coding," *Signal Processing*, Vol. 78, No. 1, pp. 1-17, Oct. 1999.
- [20] C. A. Christopoulos, J. Askelof, M. Larsson, "Efficient Methods for Encoding Regions of Interest in the Upcoming JPEG 2000 Still Image Coding Standard," *IEEE Signal Processing Letters*. Vol. 7, No. 9, 247-249, Sept. 2000.
- [21] M. Rabbani and D. Santa Cruz, "JPEG2000 image coding standard," *In Proc. IEEE International Conference on Image Processing. ICIP '01*, Thessaloniki, Greece, October 2001.
- [22] D. Salomon, *Data Compression: The Complete Reference*. Springer-Verlag, London, Fourth Edition, 2007, pp. 22-32.