A New Application of MSPIHT for Medical Imaging

[1]A. ZITOUNI, [2]Z. BAARIR, [3]A. OUAFI, [4]Abdlmalik TALEB AHMED
[1 2 3] LESIA Laboratory of Research, Electronic Department University of Biskra, Biskra, Algeria
[4] University of Valenciennes et du HainautCambrésis FRE CNRS 3304UVHC LAMIH le mont houy 59313 Valenciennes Cedex 9 ,France
[1]a.zitouni@univ-biskra.dz, [2]Baarir_Z@yahoo.fr, [3]ou_karim@yahoo.fr, [4]AbdlmalikTaleb-Ahmed@univ-valenciennes.fr

Abstract-**In this paper, we propose a new application for medical imaging to image compression based on the principle of Set Partitioning In Hierarchical Tree algorithm (SPIHT). Our approach called , the modified SPIHT (MSPIHT), distributes entropy differently than SPIHT and also optimizes the coding. This approach can produce results that are a significant improvement on the Peak Signal-to-Noise Ratio (PSNR) and compression ratio obtained by SPIHT algorithm, without affecting the computing time. These results are also comparable with those obtained using the Set Partitioning In Hierarchical Tree (SPIHT) and Joint Photographic Experts Group 2000 (JPG2) algorithms.**

Keywords: **Image compression, SPIHT, MSPIHT, Entropy, Coding, PSNR, Compression ratio.**

I. INTRODUCTION

The SPIHT algorithm is a fast and efficient technique for image compression . Like EZW and other embedded wavelet compression schemes, SPIHT generally operates on an entire image at once. The whole image is added and transformed, and then the algorithm requires repeated access to all coefficient values. There is no structure to the order in which the coefficient values are accessed. The SPIHT method is not a simple extension of traditional methods for image compression, but represents an important advance in the field. The method deserves special attention because it provides the following: image quality, progressive image transmission, optimized embedded coding and lossless compression rate or distortion specification.

SPIHT is a method of coding and decoding the wavelet transform of an image. By coding and transmitting information about the wavelet coefficients, it is possible for a decoder to perform an inverse transformation on the wavelet and reconstruct the original image. The entire wavelet transform does not need to be transmitted in order to recover the image. Instead, as the decoder receives more information about the original wavelet transform, the inverse-transformation will yield a better quality reconstruction (i.e. higher peak signal to noise ratio) of the original image. SPIHT generates excellent image quality and performance due to several properties of the coding algorithm. A lot of algorithms

were proposed in the literature like Block-Based Modified SPIHT (BMSPIHT) algorithm that combines both the features of zerotree and zeroblock algorithms into a single algorithm [1]. The Wavelet Packet SPIHT (WP-SPIHT) algorithm introduces a new implementation of wavelet packet decomposition which is combined with SPIHT [2]. The Contourlet transform SPIHT (CT-SPIHT) focuses mainly on the new fingerprint compression using Contourlet transform (CT), which includes elaborated repositioning algorithm for the CT coefficients, and modified set partitioning in hierarchical trees (SPIHT) which is applied to get better quality [3].

In this paper, we propose a modification of the SPIHT [4] coding algorithm for coding wavelet coefficients. Our modification is called the Modified SPIHT (MSPIHT) and has two specificities: it distributes entropy differently than the original SPIHT algorithm and it optimizes the coding. In addition, the robustness of the MSPIHT compares favorably with the original SPIHT algorithm and both the JPEG [5,6] and JPG2 [7] algorithms.

Following this introduction, the principle of the Set Portioning In Hierarchical Tree algorithm is reviewed (sect. 2). In sect.3, the proposed MSPIHT algorithm is described in detail. In sect.4, the results obtained with the MSPIHT algorithm are analyzed and compared with results from SPIHT algorithm and the JPEG and JPG2 algorithms for several test images.

II. SPIHT ALGORITHM

The SPIHT algorithm is unique in that it does not directly transmit the contents of the sets, the pixel values, or the pixel coordinates. What it does transmit is the decisions made in each step of the progression of the trees that define the structure of the image. Because only decisions are being transmitted, the pixel value is defined by what points the decisions are made and their outcomes, while the coordinates of the pixels are defined by which tree and what part of that tree the decision is being made on. The advantage to this is that the decoder can have an identical algorithm to be able to identify with each of the decisions and create identical sets along with the encoder.

The part of the SIPHT that designates the pixel values is the comparison of each pixel value to $2^n \leq | c_{ij} | < 2^{n+1}$ with each pass of the algorithm having a decreasing value of *n* *(threshold* : $n = \left|\log_2(\max_{(i,j)\in image})\left| coeff_{(i,j)}\right|)\right|$).

In this way, the decoding algorithm will not need be passed the pixel values of the sets but can get that bit value from a single value of *n* per bit depth level. This is also the way in which the magnitude of the compression can be controlled. By having an adequate number for *n*, there will be many loops of information being passed but the error will be small, and likewise if *n* is small, the more variation in pixel value will be tolerated for a given final pixel value. A pixel value that is $2^n \le |\ c_{ij}\ |$ is said to be significant for that pass.

By sorting through the pixel values, certain coordinates can be tagged at "significant" or "insignificant" and then set into partitions of sets. The trouble with traversing through all pixel values multiple times to decide on the contents of each set is an idea that is inefficient and would take a large amount of time. Therefore the SPIHT algorithm is able to make judgments by simulating a tree sort and by being able to only traverse into the tree as much as needed on each pass. This works exceptionally well because the wavelet transform produces an image with properties that this algorithm can take advantage of. This "tree" can be defined as having the root at the very upper left most pixel values and extending down into the image with each node having four (2 x 2 pixel group) offspring nodes (cf. Fig.1)

The wavelet transformed image [8,9,10,11,12] has the desired property that the offspring of a node will have a smaller pixel magnitude value than the parent node. By exploiting this concept, the SPIHT algorithm will not have to progress through all the pixels in a given pass if it need not go past a certain node in the tree. This means that the combined lists do not need to contain all the coordinates of every pixel, just those that will show the adequate comparison information.

Unfortunately, using tree traversal algorithms would slow down the performance of the system and create unnecessary complex data structures. Instead of tree traversal, the SPIHT algorithm uses sets of points to be able to hold the minimal amount of values and still make comparisons to other lists instead of many bulky tree structures.

III. Proposed Algorithm (Modified MSPIHT)

The following sets can represent the corresponding tree representations:

- O(i,j) is the set of coordinates of all offspring of node (i,j),
- D(i,j) is the set of all coordinates that are descendants (all nodes that are below) of the node (i,j),
- L(i,j) is the set of all coordinates that are descendants but not offspring of node (i,j).

The lists that will be used to keep track of important pixels are:

- LIS: List of Insignificant Sets, this list is one that shows us
- that we are saving work by not accounting for all coordinates but just the relative ones.
- LIP: List of Insignificant Pixels, this list keeps track of pixels to be evaluated
- LSP: List of Significant Pixels, this list keeps track of pixels already evaluated and need not be evaluated again.

A general procedure for the code is as follows (cf. Fig 2):

3.1 Initialization: output *n*, *n* can be chosen by user or predefined for maximum efficiency

$n = \left|\log_2(\max_{(i,j)\in image})\left| coeff_{(i,j)}\right|)\right|$.

LSP is empty, add starting root coordinates to LIP and LIS.

3.2 Sorting pass: (new *n* value)

3.2.1 for entries in LIP: (stop if the rest are all going to be insignificant)

- decide if it is significant and output the decision result

If it is significant, move the coordinate to LSP and output the sign of the coordinate

3.2.2 for entries in LIS: (stop if the rest are all going to be insignificant)

-if the entry in LIS represents D(i,j) (every thing below node on tree)

- decide if there will be any more significant pixels further down the tree and output the decision result

- if it is significant, decide if all of its four children (O(i,j)) are significant and output decision results (cf. fig 4)

-if significant, add it to LSP, and output sign

-if insignificant, add it to LIP

-if the entry in LIS represent L(i,j) (not children but all others)

-if the four children (O(i,j)) are insignificant, their coding is performed by one bit "0" in the outbit set instead of four bits "0000" encoding in the SPIHT algorithm.

- decide if there will be any more significant pixels in L(i,j) further down the tree and output the decision result

- if there will be one, add each child to LIS of type D(i,j) and remove it from LIS

3.2.3 Refinement Pass: all values in LSP are now $2^n \le |\ c_{ij}\ |$

For all pixels in LSP, output the nth most significant bit (cf. fig 3)

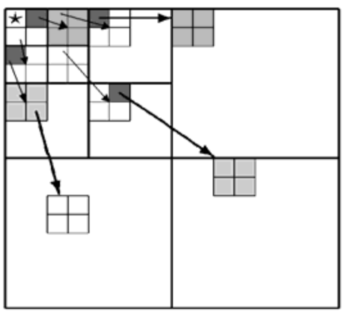

Fig. 1: Spatial orientation trees

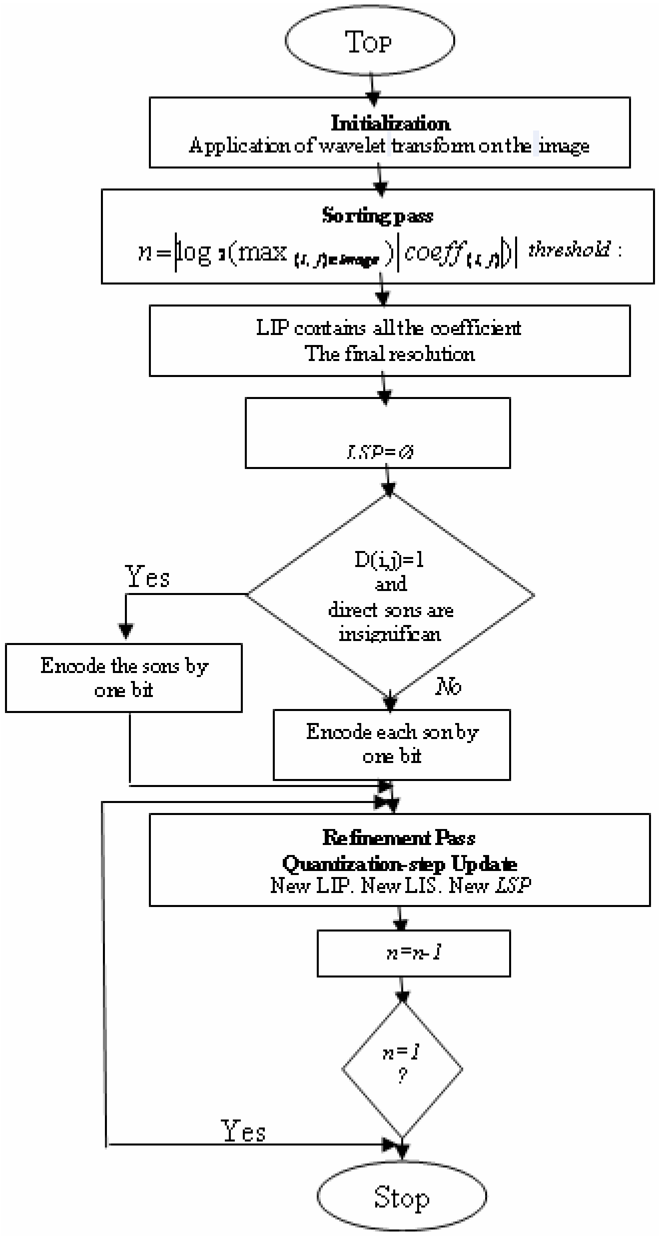

Fig. 2: Structure of the MSPIHT algorithm

3.2.4 Quantization-*step Update:*

A bit corresponding to 2^{j-1} is emitted for all the significant values in the list LSP in order to increase the precision of those values transmitted [5, 13]. The significant values {63, -45, 61 and 49} from the matrix test (cf. fig 5) are quantified respectively by the bits " 1 0 1 1 " [1]. Then, step B of the algorithm is repeated on the image residue by incrementing j by one.

This process is reiterated until the desired quality of the reconstructed image is reached or until the number of transferable bits required is exceeded.

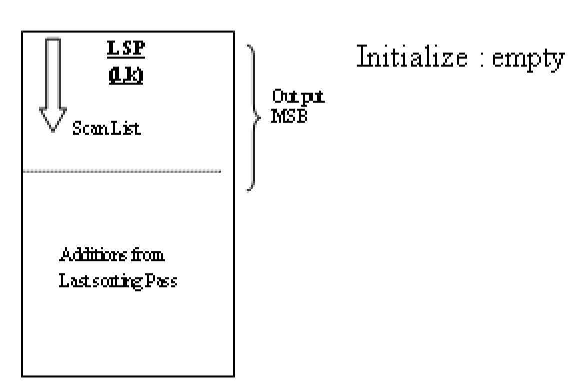

Fig. 3 : SPIHT refinement pass

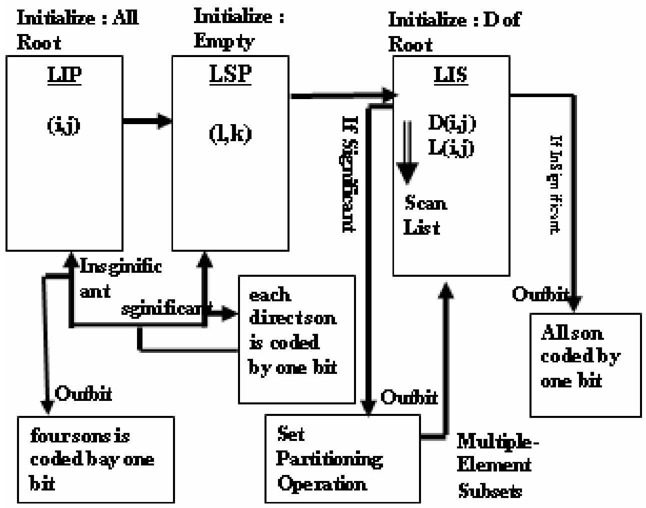

Fig. 4 : Processus de regroupement des coefficients

IV. DIFFERENCE BETWEEN THE MSPIHT

The difference between the MSPIHT algorithm that we propose and the SPIHT algorithm lies in the insignificance test process used for the set of coordinates of all offspring of node (i,j) and the coding procedure used for the outbit symbols.

	0	1	2	3	4	5	6	7
0	63	-34	**49**	10	7	-13	12	7
1	-31	23	-14	-13	3	4	6	1
2	**15**	**14**	3	-12	5	-7	3	9
3	**-9**	**-7**	-14	8	4	-2	3	9
4	-5	9	-1	**47**	4	-6	-2	2
5	3	0	-3	2	2	-2	0	4
6	2	-3	6	4	3	6	3	6
7	5	11	5	6	0	3	-4	4

Fig. 5: Example 1 of decomposition to three resolutions for an 8x8 matrix.

4.1 Insignificance test process

Let us consider the matrix test (cf. fig 5) for a first iteration (initial threshold $T_0 = 32$).

Result of the algorithms SPIHT and MSPIHT applied to example1 matrices (cf. fig 5)

Example 1

SPIHT

OUTBIT : +1 -1 0 0 1 +1 0 0 0 1 **0 0 0 0** 0 0 1 0 1 0 +1 0 0 0 0

LSP : 1 0 1 0

MSPIHT

OUTBIT :+1 -1 0 0 1 1 +1 0 0 0 10 **0** 0 0 1 0 1 0 +1 0 0 0 0

LSP : 1 0 1 0

As in Example 1 matrices (cf. fig 5) the offspring of (-31) are encoded in MSPIHT with one single Symbol "0" in set outbit instead of four Symbol "0000" in the SPIHT algorithm.

These are the initial MSPIHT settings. The initial threshold is set to 32 Example 2 matrices (cf. fig 6) The notation (i,j)A or (i,j)B, indicates that an LIS entry is of type 'A' or 'B', respectively. Note the duplication of co-ordinates in the lists, as the sets in the LIS are trees without the roots. The coefficient (0,0) is not considered a root.

MSPIHT begins coding the significance of the individual pixels in the LIP. When a coefficient is found to be significant it is moved to the LSP, and its sign is also coded. We used the notation +1 and -1 to indicate when a bit 1 is immediately followed by a sign bit.

After testing pixels it begins to test sets, following the entries in the LIS (active entry indicated by bold letters). In this example D(1, 0) is the set of 20 coefficients {(2,0), (3,0), (2,1), (3,1), (4,0),(5,0), (6,0), (7,0), (4,1), (5,1), (6,1), (7,1), (4,2), (5,2), (6,2), (7,2), (4,3), (5,3), (6,3), (7,3)}.

Because D(1, 0) is significant MSPIHT next tests the insignificance of the four offspring {2,0), (3,0),(2,1), (3,1)}.

After all offspring are tested they are coded by only one bit zero in the outbit set using MSPIHT algorithm insted of four bits zero when SPIHT algorithm is employed, (1, 0) is moved to the end of the LIS, and its type changes from 'A' to 'B', meaning that the new LIS entry meaning changed from D(1, 0) to L(1, 0) (i.e., from set of all descendants to set of all descendants minus offspring). Same procedure as in comments and applies to set D (0, 1) and D (1, 1).

4.2 Coding the outbit symbols

- If D(i,j)=0 the sons and their sons are all coded bay one bit zero.
- If D(i,j)=1 and one or more sons is significant each direct son is coded by one bit .
- If D(i,j)=1 and direct sons are insignificant the set of the four sons is coded bay one bit zero .

	0	1	2	3	4	5	6	7
0	63	-30	**9**	**10**	7	-13	12	7
1	-31	23	**-14**	**-13**	3	4	6	1
2	**15**	**14**	**3**	**-12**	5	-7	3	9
3	**-9**	**-7**	**-14**	**8**	4	-2	3	**45**
4	-5	9	-1	7	4	-6	-2	2
5	3	0	-3	2	2	-2	0	4
6	2	-3	6	4	3	6	3	6
7	5	11	5	**61**	0	3	-4	**49**

Fig. 6: Example 2 of decomposition to three resolutions for an 8x8 matrix.

SPIHT : Out Bit : +1 0 0 0 1 **0 0 0 0** 1 **0 0 0 0** 1 **0 0 0 0** 1 0 0 0 1 0 0 0 +1 1 0 0 0 1 0 0 0 +1 1 0 0 0 1 0 0 0 +1

Lsp : 1 0 1 1

MSPIHT : Out Bit +1 0 0 0 10 **0** 1 0 **0** 10 **0** 1 0 0 0 1 0 0 0 +1 1 0 0 0 1 0 0 0 +1 1 0 0 0 1 0 0 0 +1

Lsp : 1 0 1 1

The code of insignificance of the four offspring for coefficient (0,1){**9,10,-14,-13**},(1,0){**15,14,-9,-7**}and (1,1){**3, -12,-14,8**} its one zero in set out bit in bold in figure 5 is the symbol '*0*', rather than the "*0000*" used in SPIHT algorithm.

Number of symbols to calculate the Outbit list coefficients for both Example 1(fig.5) and Example 2 (cf. fig.6).

Without using any other form of entropy coding the SPIHT algorithm used 48 bits in this first pass but in the MSPIHT entropy coding used 39 bits in this first pass (exemple2) and in (exemple1) the SPIHT algorithm used 29 bits in this pass but in the MSPIHT entropy coding used 26 bits in this first pass.

V. Results and Discussion

The MSPIHT algorithm was performed using Matlab on an INTEL Pentium Core duo (1.8 Ghz, RAM 2G).

We tested our algorithm on two different still images (RMN1 200 x200 bbp and RMN2 216 x216 bbp), according to a two-level wavelet decomposition using biorthogonal filters 9/7 [2, 9].

	Example 1		Examlpe 2	
	SPIHT	**MSPIHT**	SPIHT	**MSPIHT**
Number of Symbols	29	**28**	50	**44**

The PSNR (dB) performance and compression ratio CR (bpp) of our MSPIHT algorithm were compared to those for the SPIHT algorithm as well as to those obtained with the JPG2 and JPEG algorithms. These parameters are expressed by the following relations [4,14]:

$$PSNR(db) = 10\log_{10}\left[\frac{(255)^2}{MSE}\right]$$

$$MSE = \frac{1}{N \times M}\sum_{i=1}^{N}\sum_{j=1}^{M}(x_i - y_j)^2$$

$$CR(bpp) = \frac{number \text{ of coded bits}}{number \text{ of initial bits}}.$$

Where N, M is the image size, x_i the initial image and y_j the reconstructed image.

Referring to Tables 1-2, it is clearly seen that, regardless the threshold value, the total information required for the coding of the two medical images using the MSPIHT is much smaller than the one required using the SPIHT algorithm.

In the majority of cases, the results obtained by the MSPIHT are better than those obtained by JPEG (Table3). For rates higher than 0.50bpp, the MSPIHT performs better than the SPIHT and JPG2 algorithms (Table3) . Even for lower rates, the MSPIHT performance is still very close to that of the SPIHT and JPG2 algorithms, and the results start getting better around 0.25bpp for the RMN1 (200×200) and RMN2 (216×216) images (cf. Fig 7-Fig 8).

The MSPIHT method is not a simple extension of traditional methods for image compression, and represents an important advance in the field. The method deserves special attention because it provides the following:

Image quality, progressive, image transmission, optimized embedded coding, lossless compression and rate or distortion specification.

TABLE 1 OUTBIT AND LSP OF THE MSPIHT AND SPIHT ALGORITHMS APPLIED TO THE RMN1 IMAGE 200×200 FOR THRESHOLD TH =32.

MSPIHT				**SPIHT**		
Th	**OUT BIT**	**LSP**	**TOTAL INFORMATION(bits)**	**OUT BIT**	**LSP**	**TOTAL INFORMATION(bits)**
Th/8	6695	983	7678	7049	983	8032
Th/16	13238	3220	16458	14396	3220	17616
Th32	23496	8069	31565	25060	8069	33129
Th/64	31732	14358	46090	33037	14358	47395

TABLE 2 OUTBIT AND LSP OF THE MSPIHT AND SPIHT ALGORITHMS APPLIED TO THE RMN2 IMAGE 216×216 FOR THRESHOLD TH=32.

MSPIHT				**SPIHT**		
Th	**OUT BIT**	**LSP**	**TOTAL INFORMATION(bits)**	**OUT BIT**	**LSP**	**TOTAL INFORMATION(bits**
Th/8	6952	905	7857	7240	905	8115
Th/16	12181	2558	14739	13087	2558	15645
Th32	21666	6257	27923	23327	6257	29584
Th/64	31388	12250	43638	33086	12250	45336

TABLE 3 RESULTS OF THE VARIOUS ALGORITHMS APPLIED TO TEST IMAGES (RMN1216×216 AND RMN2 216×216).

Image	Coding algorithm	PSNR (db)				
		0.25 bpp	0.5 bpp	0.75 bpp	1bpp	1.5 bpp
RMN1 (200*200)	MSPIHT	**26.39**	**29.61**	**34.00**	**39.45**	**45.92**
	SPIHT	26.10	28.90	31.40	33.10	36.40
	JPG2	27.01	30.40	33.10	35.13	39.30
	JPEG	22.60	28.60	31.50	34.10	45.50
RMN2 (216*216)	MSPIHT	**29.14**	**32.33**	**36.72**	**42.47**	**49.43**
	SPIHT	27.80	31.30	34.10	36.20	40.60
	JPG2	28.20	31.90	34.60	36.80	41.20
	JPEG	26.8	29.60	31.70	34.10	46.20

VI. CONCLUSION

In this paper, we developed an image compression algorithm (MSPIHT) based on the same principle as Set Partitioning In Hierarchical Tree algorithm. This algorithm is able to improve the performance of the SPIHT algorithm because 1) using one insignificance symbols instead of four better optimizes the entropy and 2) the binary regrouping of these symbols on outbit set better optimizes the coding.

The proposed algorithm is able to accomplish this without increasing computation time. In addition, this algorithm performed comparably with the JPG2, SPIHT and JPEG algorithms, which could be very interesting for the field of hierarchical coding.

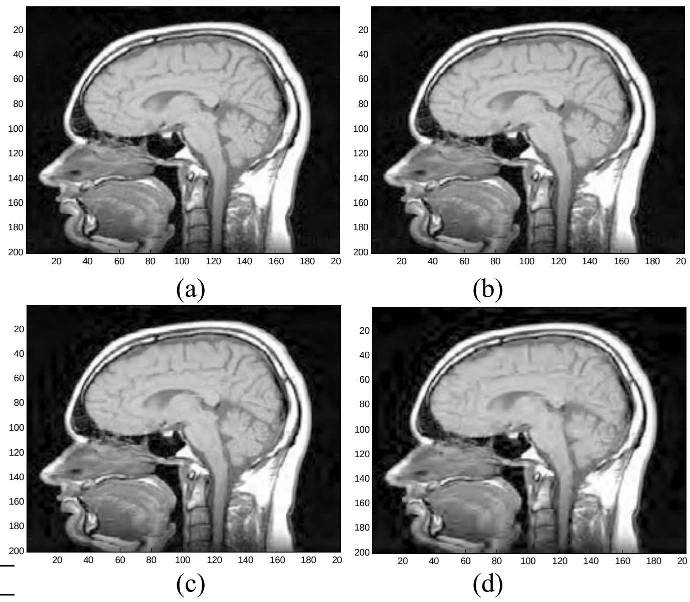

Fig.7: Results for RMN1 image reconstructed using the MSPIHT algorithm.(a) the original RMN1 image 200× 200.(b) The RMN1image reconstructed With PSNR=45,92 dB and CR=1.5 bpp. (c) The RMN1 image reconstructed With PSNR=39,45 dB and CR=1 bpp. (d) The RMN1 image reconstructed With PSNR=34,00dB and CR=0.75 bpp.

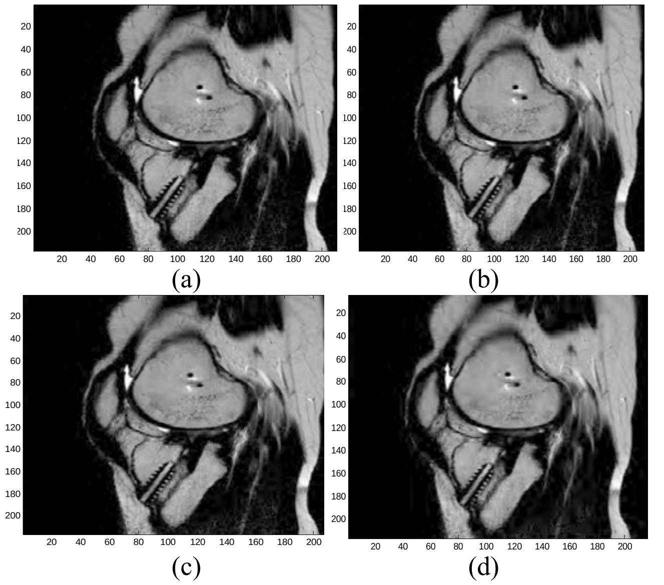

Fig.8: Results for the RMN2 image reconstructed using the MSPIHT algorithm.(a) the original RMN2 image 216× 216.(b) The RMN2 image reconstructed With PSNR=49.,43 dB and CR=1.5 bpp. (c) The RMN2 image reconstructed With PSNR=42,47 dB and CR=1 bpp. (d) The RMN2 image reconstructed With PSNR=36.72 dB and CR=0.75 bpp.

REFERENCE

[1] R.Viswanath, Reddy1, T.Sreenivasulu Reddy2, and Dr.Govind Sharma31,3 IIT Kanpur, Dept. of EEE, Kanpur, Indi, “Efficient Coding of Image Subbands using Blockbased Modified SPIHT”, International Journal of Recent Trends in Engineering, Vol 2, N° 5, November 2009.

[2] Nicola Sprljana,Sonja Grgicb,Mislay Grgicb, “Modified Spiht algorithm for wevlet packet image coding”, real-time Imaging 11 (2005) 378-388.

[3] R.Suudhakar, R.Karthing, and S.Jayaraman, “Fingerprint Compression Using Contourlet Transform with Modified SPIHT Algorithm”. Iranian journal of electrical and Computer Engineering Vol 5,NO,1,Winter-spring 2006.

[4] A.Said and W.A.Pearlman,”A new,fast and efficient image cdec based on set partitioning in hierarchical trees”,IEEE trans.circuits syst.video technol.6,243-250 (jun 1996)

[5] J.M. Shapiro, “Embedded image coding using zerotres of wavelet coefficients,” IEEE Trans. on Signal Processing, vol. 41, pp. 3445–3462, Dec. 1993.

[6] M. D. Adams, A. Antoniou, “Reversible EZW-Based Image Compression Using Best-Transform Selection and Selective Partial Embedding”. IEEE Transactions on circuits and systems-II: Analog and digital processing, Vol. 47, NO. 10, pp. 1119-1122, Oct. 2000.

[7] ISO/ IEC CD 15444-1,ITU-T Rec T,800 CD VI .O (Committee Draft International Standard),”Information technology-JPEG2000 Image Coding System”,(December 1999).

[8] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. “Image coding using wavelet transform”. IEEE Transaction, Image Processing, 1(2):205–220, April 1992.

[9] A.Cohen,I.Daubechies,et J.feauveau. ”Biorthogonal bases of compactly supported wavelets”.communications on Pure and Applied Mathematics,45/485-560,1992.

[10] M. Angeles Losada, Gulay Tohumoglu, David Fraile, Antonio Artès, “Multi-itération wavelet zerotree coding for image compression”. Elsevier Science, Signal Processing, vol. 80, pp. 1281-1287, 2000.

[11] G. Donovan, J. Geronimo, D. Hardin, P. Massopust, “Construction of orthogonal wavelets using fractal interpolation functions”, SIAM J. Math. Anal., vol. 27, N_4, pp. 1158-1192, 1996.

[12] B. Jawerth, W. Sweldens, “An overview of wavelet based multiresolution analyses”, web site, www.mathsoft.com/wavelets.html, pp. 1-39, 1997.

[13] R. Thomas, “Algorithmes Rapides de Reconstruction en Tomographie par Compression des Calculs. Application à la Tomofluoroscopie 3D”. Thèse de Doctorat, Institut national polytechnique de GRENOBLE, Octobre 2002.

[14] Y.Low, , R.Besa, ”Wavelet-based medical image compression using EZW”. 4th National Conference on Telecommunication Technology Proceedings, Malaysia (2003).

Athmane Zitouni recieved his B.eng. degree in electronic in 1997 and his Master in 2003 from Biskra University (Algeria). Presently he is an associate professor at the Engineering Science Faculty of Biskra University (Algeria) since 2004. His research interest includes signal processing and image coding.

Zine–Eddine BAARIR received the B.Sc. degree in electrical and electronic engineering, option communication Systems from Swansea University , Wales in 1981. He obtained a diploma of higher studies (DEA) in Electronic Engineering from Orsay. University, Paris11 in 1983. The PhD was received in Electronic engineering – Signal processing from Ecole Supérieure d’électricité (SUPELEC), Paris in 1986. He joined with Electronic Eng at Mohamed Khider University of Biskra, Algeria, where is actually associate professor, since 1990.

Abdelkrim Ouafi received his B.eng degree in electronic in 1997 and his Master in 2001 from Biskra University (Algeria). The PhD was received in department of Electrical Engineering at Biskra University, Algeria in 2010. Presently he is an associate professor at the Engineering Science Faculty of Biskra University (Algeria) since 2002. He is member of LESIA laboratory of research “Experts Systems, Imaging and their applications in Engineering” since 2000. His research interest includes signal processing and image coding.

Abdelmalik Taleb-Ahmed received a post graduate degree and a Ph. D. in Electronics and Microwaves from the Université des Sciences et Technologies de Lille 1 in 1988 and 1992. From 1992 to 2004, He was an Associate Professor at the Université du Littoral Cote d’Opale de Calais. Since 2004, He is currently a Professor at the Université de Valenciennes et du Hainaut Cambrésis, and does his research at the LAMIH URM CNRS 8530, His research interests includes signal and image processing.