Image coding of 3D volume using wavelet transform for fast retrieval of 2D images

T. Vijayaraghavan and K. Rajan

Abstract: An encoder/decoder system for 3D volumetric medical data is proposed. The system allows fast access to any 2D image by decoding only the relevant information from each subband image and thus provides minimum decoding time. This will be of immense use for the medical community because most of the computerised tomography, magnetic resonance imaging and positron emission tomography modalities produce volumetric data. As a fully-fledged 3D wavelet transform is used for compression, the advantage of good compression ratio is preserved. Preprocessing is carried out prior to wavelet transform, to enable easier identification of coefficients from each subband image. Inclusion of special characters in the bit stream (markers) facilitates access to corresponding information from the encoded data. Experiments are carried out by performing Daub4 filter along x (row) and y (column) directions and Haar filter along the z (slice) direction to account for the difference between interslice and intraslice resolution. The performance of the system has been evaluated on four sets of volumetric data and the results are compared to other 3D encoding/2D decoding schemes. Results show that for slice spacing of $3-10$ mm, there is substantial improvement in decoding time. The speedup is found to be ~ 2.

1 Introduction

Most of the current imaging modalities such as magnetic resonance imaging (MRI), computerised tomography (CT) and positron emission tomography (PET) produce 3D volumetric data. Fig. 1 shows a typical 3D MRI. 3D image volume is assumed to be a stack of 2D frames along the z direction. Compression is indispensable for transmission and storage of such a large set of data. The basic idea is to take advantage of the correlation among the data samples in the 3D space to improve compression efficiency. A few studies have been reported in the literature on application of 3D wavelet transform for medical image compression [1–3]. The basic idea behind those methods is to apply separable 3D wavelet transform to remove interslice and intraslice redundancy. Wang and Huang [4] employed different sets of wavelet filters along x, y and z directions to account for the differences between interslice and intraslice resolutions. The methods in the literature [1–4] are for 3D compression and 3D decompression, that is, the entire volume is reconstructed as a whole. However, in practice, fast access to 2D image is beneficial for medical practitioners. In this context, random access to 2D images with multidimensional layered zero coding and arithmetic coding [5] has been demonstrated in Menegaz and Thiran [6] and Vijayaraghavan et al. [7], respectively. These techniques do not provide good decode time for random access to a single 2D image. Performing separable

3D wavelet transform (WT) on a 3D volumetric data results in eight 3D subbands; each subband image has a contribution from a set of 2D images. To reconstruct a 2D image, information has to be retrieved from multiple 3D subbands. For example, to reconstruct one plane of the original image, we need to retrieve the corresponding plane from all 3D subbands. The decoding has to be carried out on all these coefficients [6, 7]. The data sets so retrieved from all 3D subbands are sufficient to reconstruct multiple 2D images. A substantial speedup in decoding time is possible only if the relevant coefficients from these 3D subbands are clubbed together. In the proposed approach, a preprocessing technique is performed prior to WT for easier identification of the relevant coefficients from each 2D subband to decode a single 2D plane.

2 Proposed scheme

We follow the conventional 3D compression scheme with an additional preprocessing stage before carrying out WT at each level. Before performing WT at one level, original 3D image data is reorganised. The resulting 3D-wavelet coefficients are again rearranged before taking WT for subsequent levels. In the proposed approach, Haar [8, 9] filter is performed along the z direction and Daub4 [10] filter along the x and y directions. Daub4 filter is used because it is a representative short filter bank that preserves edge information and also offers lower computational complexity. In most of the MRI/CT volumetric images, the slice spacing varies between $3-10$ mm along the z direction. It is found that the Haar wavelet performs better for image sets with slice distances greater than 3 mm [4]. So Haar wavelet has been used along z direction. The next step is the quantisation of the wavelet coefficients followed by arithmetic coding to achieve compression. We have set the quantisation step size as unity to achieve nearly lossless compression. We say nearly lossless because the compression may not be reversible as the WT filters have
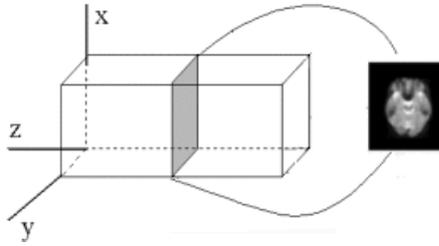
Fig. 1 *3D MRI image data*

non-integer tap weights and WT produces non-integer transform coefficients that are truncated to finite precision. Encoded data is obtained by scanning the subbands from the coarsest resolution LLL to the finest resolution HHH. Corresponding coefficients required to reconstruct a single 2D image are identified and encoded. This constitutes one scanning pass. After each scanning pass, a special character (marker) is introduced into the bitstream. Thus, all the wavelet coefficients are encoded in several scanning passes. The bitstream syntax is shown in Fig. 2. The reverse operation is carried out on the receiving side. Once the user specifies the image index of 2D along z direction, the relevant information is retrieved and decoded. Postprocessing is carried out at every level after inverse wavelet transform (IWT), which is exactly the reverse of preprocessing.

3 Theory

3.1 Preprocessing

Preprocessing refers to rearrangement of data between planes. Let G_{pmn} be a 3D data set of size $S \times R \times C$, where $p = 1, 2, \ldots, S$, $m = 1, 2, \ldots, R$, $n = 1, 2, \ldots, C$ and S, R, C represent the number of slices, rows and columns, respectively. Let us start our discussion by considering only two planes from the 3D data and we will generalise this concept at the end of this subsection. Let $p = 1$ and $p = 2$ be the first two 2D images from the original 3D data, as shown in Fig. 3a. Each plane is divided into 2^y equal parts with dimensions $R/2^{y-1} \times C/2^{2y-1}$ called blocks, whose indices are given as $q = 1$ and $q = 2$ (Fig. 3b), where y refers to the level to be applied. Let a_{pq} denote the data in each block; subscript p and q uniquely identify the blocks and q refers to the block number in the pth plane. Reorganised 3D data is placed in b_{rt}, where t refers to the block number in the rth plane (t and r take the same range of values as that of p and q). The data in a_{11} (block $q = 1$ in the plane $p = 1$) are copied to b_{11} (block $t = 1$ in the plane; $r = 1$ in the resultant). This is shown as $b_{11} = a_{11}$. The block b_{12} gets a flipped copy a_{21} represented here as $b_{12} = \text{flip}(a_{21})$. In most of the MRI/CT images, the region of interest is towards the centre. So flipping is carried out on a_{21} to ensure continuity along the z direction, which is needed only for $y = 1$. So the following operations are carried out to complete the preprocessing operation for the
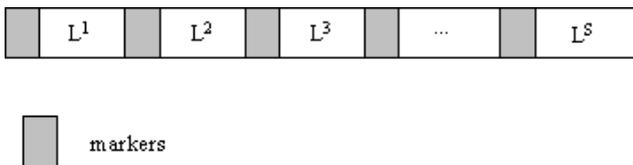

Fig. 2 *Bitstream syntax*
L^j represents corresponding information needed to reconstruct jth image
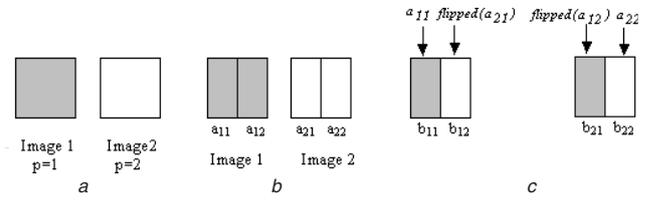

Fig. 3 *Preprocessing of 3D original image to 2D images*
a Original image slices, each of size $R \times C$. Darkened portion corresponds to image 1 and bright portion corresponds to image 2
b Each image is split into two halves, each of size $R \times C/2$
c The data are rearranged. Image 1 data is restricted to the left half in each plane and image 2 in the right half

pairs $b_{11} = a_{11}$, $b_{12} = \text{flip}(a_{21})$ and $b_{21} = \text{flip}(a_{12})$, $b_{22} = a_{22}$. The resultant b_{rt} looks as shown in Fig. 3c. In both planes, $r = 1$ and $r = 2$, image 1 and image 2 data are strictly restricted to the left and right half of the planes, respectively. Before going to the subsequent levels ($y > 1$), we generalise this process. In the volumetric 3D data, adjacent two-image planes are paired, that is, $p = 1$ and $p = 2$ form the first pair, $p = 3$ and $p = 4$ form the second pair and $p = S - 1$ and $p = S$ form the $S/2$th pair. Preprocessing is now carried out for each pair and the resultant b_{rt} blocks are generated. WT is performed for all blocks independently along x, y and z directions. This completes the first-level WT.

The subsequent-level WT is performed only to LLL subband. So it is sufficient to carry out the reorganisation step only to the LLL subband. Now we generalise the concept of preprocessing for $y > 1$. Each 2D subband image is divided into 2^y equal parts, such that the size of each block is $R/2^{y-1} \times C/2^{2y-1}$ and indices for each block are given by $q = 1, 2, \ldots, 2^y$. The number of 2D subband images at any level is $S/2^{y-1}$ and the number of pairs is $S/2^y$. Preprocessing is carried out on each pair as follows: in each pair p and $p + 1$ (where p is an odd number), odd indexed blocks $q = 1, 3, 5, \ldots, 2^y - 1$ are separated out. The number of odd indexed blocks in each plane p is 2^{y-1} and in each pair it is 2^y. Now all the odd indexed blocks are transferred to the plane $r = p$ in the resultant b_{rt} as follows: $b_{r1} = a_{p1}$, $b_{r2} = a_{p3}$, $\ldots$, $b_{r(2^{y-1})} = a_{p(2^y-1)}$, $\ldots$, $b_{r(2^{(y-1)}+1)} = a_{(p+1)1}$, $\ldots$, and $b_{r2^y} = a_{(p+1)(2^y-1)}$. Similarly, all the even indexed blocks $q = 2, 4, 6, \ldots, 2^y$ are transferred to plane $r = p + 1$ in b_{rt}. The same procedure is carried out for each and every pair. This completes the preprocessing at any level y. The preprocessed data replaces LLL subband and WT is performed to each and every block independently along x, y directions and Haar along z direction.

3.2 Haar wavelets

Like all wavelet transforms, Haar transform decomposes a discrete signal into two subsignals, namely, trend and fluctuation. Let us start our discussion by applying Haar wavelet to 1D signal and extend the same logic to 3D. Let $f = f_1, f_2, \ldots, f_N$ be an even point sequence of length N. Let a_m^1 and d_m^1 (superscript denotes level) represent the trend and fluctuation of the original signal f, respectively, where a_m^1, d_m^1 are given by

$$a_m^1 = \frac{(f_{2m-1} + f_{2m})}{\sqrt{2}} \tag{1}$$

$$d_m^1 = \frac{(f_{2m-1} - f_{2m})}{\sqrt{2}} \tag{2}$$

for $m = 1, 2, \ldots, N/2$.

Equations (1) and (2) reveal that only two nearby data have contributed to each a_m^1 and d_m^1. Applying IWT to each a_m^1 and d_m^1, reconstruct two nearby data, namely, f_j and f_{j+1} for $j = (2m - 1)$. Applying Haar wavelet to the 3D data G_{pmn} along z direction transforms both G_{pmn} and $G_{(p+1)mn}$ for $p = 1, 3, 5, \ldots, S-1$ to the $((p + 1)/2)$th image of 3D subband at level one. So access to two 2D images, namely, G_{pmn} and $G_{(p+1)mn}$ is feasible by applying IWT only for the $(p + 1)/2$th image from each subband. Now we generalise this concept to any level x. At level x and 2^x, 2D images can be reconstructed by applying IWT only to the involved set of images, that is, corresponding information at each level. The slices p to $(p + 2^x - 1)$ of the original 3D data (where $(p - 1) = K \times 2^x$ for $K = 0$, 1, 2, ...) map to the $((p + 2^x - 1)/2^x)$th image of subband at level x, that is, at coarsest resolution. The IWT is an iterative process starting at the coarsest scale; the approximation coefficients at the finer level $x - 1$ are reconstructed from the coefficients at the coarser level x. So, performing IWT only to the $(p + 2^x - 1/2^x)$th image from all subbands at the coarser level x reconstructs the corresponding approximation subbands at the next finer level $x - 1$. Instead of one image, it reconstructs two images at the next finer level $x - 1$, whose indices (we refer to the plane number) are $(2\alpha - 1)$ and 2α, where $\alpha = (p + 2^x - 1)/2^x$. The same discussion applies to the next finer level $x - 2$. In this level, there will be four images whose indices are, namely, 4α, $4\alpha - 1$, $4\alpha - 2$ and $4\alpha - 3$. Finally, at level 1 there will be $2^{(x-1)}$ images, whose indices are $2^{(x-1)}\alpha$ and $(2^{(x-1)}\alpha) - 1, \ldots, (2^{(x-1)}\alpha) - (2^{(x-1)} - 1)$.

3.3 Identification of corresponding coefficients

From the previous subsection it is clear that each subband image at level x (i.e. coarsest resolution) has a contribution from 2^x 2D images. Each subband image in the next finer level $x - 1$ has a contribution from 2^{x-1} 2D images and so on. Preprocessing involves the identification of corresponding coefficients from each subband image to enable single 2D image decoding. This is explained as follows.

Before performing one-level WT, the original 3D data is preprocessed. After preprocessing, any two adjacent planes r and $r + 1$ (where r must be an odd number) have data corresponding to one 2D image in the left half (darkened portion in Fig. 4c) and another 2D image in the right half. One-level WT ensures that each subband image has a
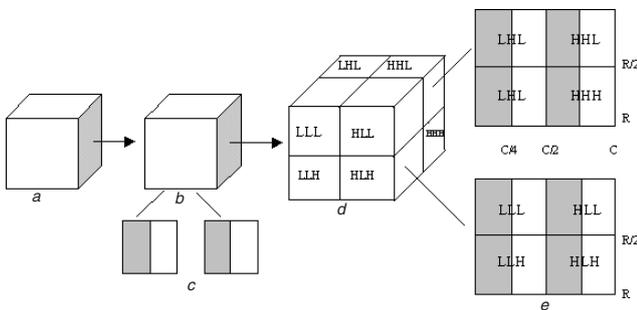
Table 1: Comparison of the number of the coefficients to be decoded with and without preprocessing

Level	With preprocessing	Without preprocessing
1	$R/2 \times C/4$	$R/2 \times C/2$
2	$R/4 \times C/16$	$R/4 \times C/4$
3	$R/8 \times C/64$	$R/8 \times C/8$

contribution from two adjacent planes of the preprocessed image data. As a result of preprocessing, each subband image has coefficients corresponding to one 2D image in one half and another 2D image in the other half; that is, a subband image of dimension $R/2 \times C/4$ has a contribution from only one 2D image. The corresponding coefficient for odd indexed image p_o is in the first half of the $((p_o + 1)/2)$th image (darkened portion in Fig. 4e) and for even indexed image p_e it is in the second half of the $p_e/2$th image of subbands. So, the number of coefficients needed from each subband image to reconstruct a single 2D image is $R/2 \times C/4$. Identification of corresponding coefficients to reconstruct a 2D image p for $p = 1, 2, \ldots, S$ from 3D data set G_{pmn} involves the following steps:

1. Identification of plane index k, given as

$$k = \begin{cases} p/2^x & \text{if } p = a \times 2^x \quad \text{for } a = 1, 2, \ldots \\ \lfloor p/2^x \rfloor + 1 & \text{otherwise} \end{cases}$$

2. Identification of the block number c in the kth plane. We divide kth subband image into 2^x equal parts, such that each has dimension $R/2^x \times C/4^x$ and the index to each one is given as $c = 1, 2, \ldots, 2^x$. Identification of corresponding coefficients involves matching index c with z, that is, $c = z$, where z is defined as follows

$$z = \begin{cases} 2^x & \text{if } p = a \times 2^x \quad \text{for } a = 1, 2, \ldots \\ p \bmod 2^x & \text{otherwise} \end{cases}$$

The number of coefficients needed from each subband image in the coarser level to reconstruct the corresponding coefficients in the next finer level (only the approximation coefficients in the next finer resolution will be reconstructed) with and without preprocessing are tabulated in Table 1 for various levels. Table 1 indicates that the number of coefficients needed to be decoded from each subband image is reduced by a factor of two at level 1, by a factor of four at level 2 and by a factor of eight at level 3. Thus, preprocessing enables easier identification of relevant coefficients from each subband image and provides faster decoding.

4 Results

The implementation of the system has been carried out on four sets of MRI images with various slice spacings. The dimensions and the slice distances for the data sets are tabulated in Table 2. The data sets used in our implementation are gray-scale images with eight bits per pixel. We have compared the decoding time, peak signal-to-noise ratio (PSNR) and compression ratio (CR) of the decoded image with preprocessing (WP) and without preprocessing (WOP). WOP techniques are demonstrated in Menegaz and Thiran [6] and Vijayaraghavan [7], where random access to a set of 2D images is accomplished by independently encoding each subband image and making the corresponding information accessible through introduction

Fig. 4 *Identification of corresponding coefficients*

a Original 3D image data
b Preprocessed image data
c r and $(r + 1)$th plane of the preprocessed image data. Darkened portion represents data from plane r. Bright portion represents data from plane $r + 1$
d 3D subbands as a result of one level WT
e A typical plane from each subband. Darkened portion represents the coefficients to reconstruct plane r and bright portion for plane $r + 1$. The dimensions are represented for the subband image

Table 2: Specifications of 3D volume image sets

Image set no.	Dimensions	Slice distance, mm
1	64 × 128 × 128	2
2	64 × 64 × 64	3
3	24 × 64 × 64	5
4	22 × 64 × 64	6

of special characters into the bitstream. The results are obtained by performing Daub4 transform along *x, y* directions and Haar transform along *z* direction for both WP and WOP cases.

### 4.1 Comparison of decoding time

Preprocessing enables easier identification of corresponding coefficients from each subband image and also reduces the number of coefficients needed to reconstruct the image. This facilitates faster decoding of desired slices. A single 2D image is reconstructed from each set and the decoding time is tabulated in Table 3. The first and second columns give the image set and the reconstructed slice number, respectively. Decoding time for each WP and WOP slice is presented in the third and fourth column, respectively. As a result of preprocessing, the speedup is improved to 1.98. Thus, fast retrieval of any single 2D slice is possible with minimum decoding time.

4.2 Comparison of PSNR and CR

PSNR is taken as the appropriate error metric to analyse the image quality. Figs. 5–8 show the image index versus PSNR for the set 1, 2, 3 and 4, respectively. The horizontal axis and vertical axis represent image (slice) index and

Table 3: Comparision of decoding time for various 2D slices

Set no.	Slice no.	T_{WP}, s	T_{WOP}, s	T_{WOP}/T_{WP}
1	31	0.54	1.07	1.98
2	37	0.418	0.83	1.985
3	15	0.412	0.819	1.987
4	14	0.414	0.822	1.985
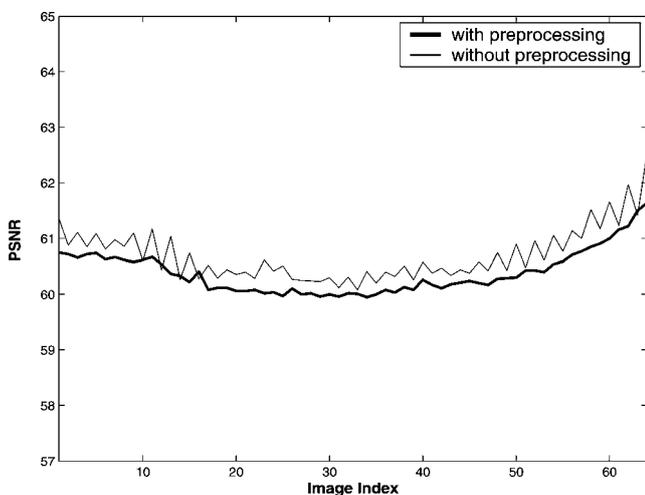
Fig. 5 *PSNR results for image set 1 with and without preprocessing*

PSNR, respectively. It is found that PSNR, for WP is almost the same as that obtained by WOP. To compare the performance of both techniques, we define the percentage discrimination in PSNR as

$$\%\text{Discrimination} = \frac{\text{PSNR}(WP) - \text{PSNR}(WOP)}{\text{PSNR}(WOP)} \quad (3)$$
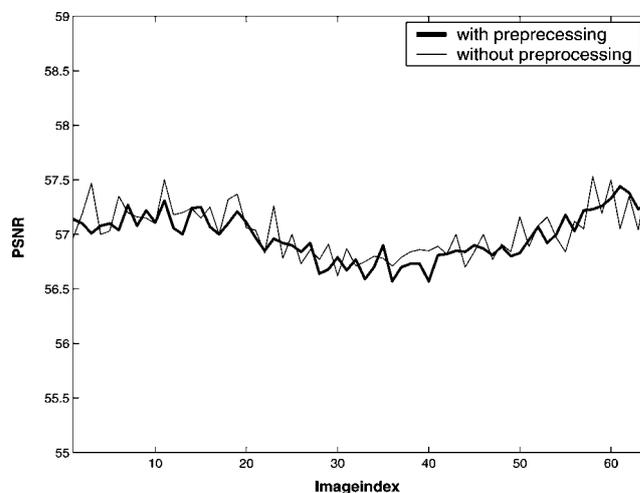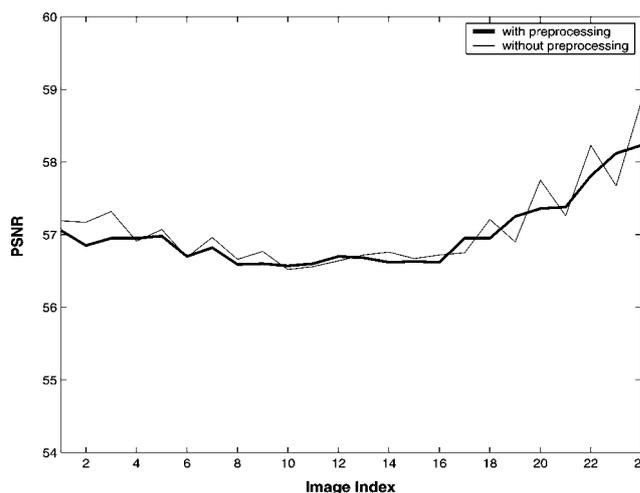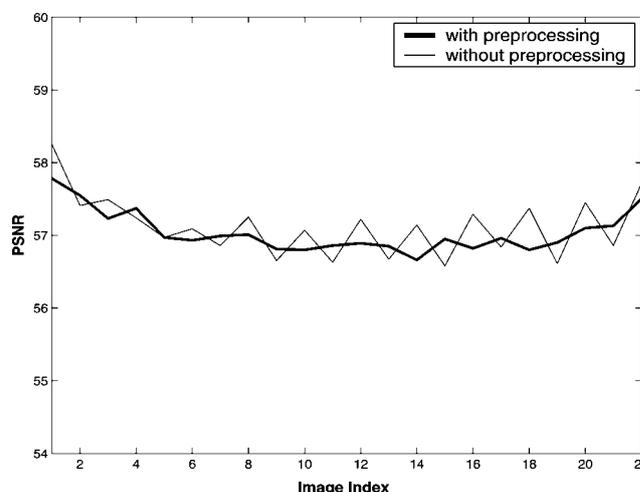

Fig. 6 *PSNR results for image set 2 with and without preprocessing*

Fig. 7 *PSNR results for image set 3 with and without preprocessing*

Fig. 8 *PSNR results for image set 4 with and without preprocessing*

Table 4: Comparison of PSNR and CR

Set no.	PSNR		%Discrimination	CR	
	WP	WOP		WP	WOP
1	60.385	60.69	−0.5	4.45	4.876
2	56.98	57.02	−0.07	1.268	1.368
3	56.98	57.08	−0.17	1.38	1.44
4	57.04	57.12	−0.14	1.54	1.64

Table 4 shows the overall PSNR for both techniques. Percentage discrimination for all image sets is tabulated in Table 4. Percentage discriminations for sets 1, 2, 3 and 4 are about −0.5%, −0.07%, −0.175% and −0.14%, respectively. A negative sign indicates that PSNR is slightly better for WOP. Overall compression ratio is calculated for both techniques and the results are tabulated in Table 4. The reason for discrimination in PSNR and CR is explained as follows: original 3D data has well-defined correlation between the slices, which depend on slice distances. The general trend is that the smaller the slice distance, the better the correlation. As a result of preprocessing, the correlation between the slices is affected. It is found from Table 4 for set 2, set 3 and set 4 that the percentage discrimination is less when compared to set 1 because of the larger slice spacing. So correlation between the slices is less prominent when compared to image set 1. So, for a slice spacing of 3–10 mm, this system provides minimum decoding time at comparable rates.

5 Conclusion

We have presented a system featuring 3D encoding and 2D decoding. 3D encoding guarantees the best compression ratio. The proposed method allows fast access to any 2D image by decoding the relevant coefficients from each subband image. A new preprocessing scheme is proposed for faster retrieval of a single 2D image. Markers included in the bitstream facilitate random access to the corresponding information required to reconstruct any 2D image. Comparison with other schemes shows that, for a slice spacing of 3–10 mm, this method uses the least decoding time. The results are very encouraging and uses will include embedded applications where decoding time is critical.

References

1 Wang, J., and Huang, H.K.: 'Three-dimensional medical image compression using wavelet transform with parallel computing', *Proc. SPIE*, 1995, **2431**, pp. 162–172
2 Baskuet, A., Benoit-Cattin, H., and Odet, C.: 'On a 3D medical coding method using a separable 3D wavelet transform', *Proc. SPIE*, 1995, **2431**, pp. 173–183
3 Wei, J., Saipetch, P., Panwar, R., Chen, D., and Ho, B.K.T.: 'Volumetric image compression by 3D discrete wavelet transform', *Proc. SPIE*, 1995, **2431**, pp. 184–194
4 Wang, J., and Huang, H.K.: 'Medical image compression using 3D wavelet transform', *IEEE Trans. Med. Imag.*, 1996, **15**, (4), pp. 547–554
5 Witten, I.H., Neal, R.M., and Cleary, J.G.: 'Arithmetic coding for data compression', *Commun. ACM*, 1987, **30**, pp. 520–540
6 Menegaz, G., and Thiran, J.P.: '3D encoding/2D decoding of medical data', *IEEE Trans. Med. Imag.*, 2003, **22**, (3), pp. 424–440
7 Vijayaraghavan, T., Biju, M., and Rajan, K.: 'Medical image compression using three-dimensional encoding and two-dimensional decoding'. Proc. 12th Int. Conf. on Advanced Computing and Communication (ADCOM 2004), Ahmedabad, December 2004, pp. 600–606
8 Albanesi, M.G., and De Lotte, I.: 'Image compression by the wavelet decomposition', *Signal Process.*, 1992, **3**, (3), pp. 265–274
9 Walker, J.S.: 'A primer on wavelets and their scientific applications' (CRC Press LLC, 1999)
10 Daubechies, I.: 'Orthonormal bases of compactly supported wavelets', *Comm. Pure Appl. Math.*, 1988, **41**, pp. 909–996