The Journal of Systems and Software 82 (2009) 526–534

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

A lossy 3D wavelet transform for high-quality compression of medical video

Gregorio Bernabé [a,*], Jose M. García [a], José González [b]

[a] Departamento de Ingeniería y Tecnología de Computadores, Universidad de Murcia, Campus de Espinardo, 30071 Murcia, Spain
[b] Intel Barcelona Research Center, Intel Labs, Barcelona, 08034 Barcelona, Spain

ARTICLE INFO

Article history:
Received 20 December 2006
Received in revised form 23 September 2008
Accepted 28 September 2008
Available online 17 October 2008

Keywords:
Telemedicine
Medical video
High-quality lossy compression
3D fast wavelet transform

ABSTRACT

In this paper, we present a lossy compression scheme based on the application of the 3D fast wavelet transform to code medical video. This type of video has special features, such as its representation in gray scale, its very few interframe variations, and the quality requirements of the reconstructed images. These characteristics as well as the social impact of the desired applications demand a design and implementation of coding schemes especially oriented to exploit them. We analyze different parameters of the codification process, such as the utilization of different wavelets functions, the number of steps the wavelet function is applied to, the way the thresholds are chosen, and the selected methods in the quantization and entropy encoder. In order to enhance our original encoder, we propose several improvements in the entropy encoder: 3D-conscious run-length, hexadecimal coding and the application of arithmetic coding instead of Huffman. Our coder achieves a good trade-off between compression ratio and quality of the reconstructed video. We have also compared our scheme with MPEG-2 and EZW, obtaining better compression ratios up to 119% and 46%, respectively for the same PSNR.

1. Introduction

In the last few years, there has been a considerable increase in the volume of medical images and video generated in hospitals. In a typical hospital, vast numbers tera-bytes (10^{15}) of medical data are generated every year. The medical multimedia information is different from other multimedia data because of its particular properties. There are legal and strict regulations applied to medical multimedia information, since the health of a patient depends on the correctness and accuracy of this information. Moreover, the integrity, confidentiality and security of medical data is crucial to protect it from accidental or malicious alteration during interchange and storage. Another critical property is that the information related to a patient must be available in a short period of time, whenever or wherever it is required, and especially so in the case of emergencies.

Most of the medical history of a patient must be kept and stored since legislation requires all recorded healthcare information to be preserved for a certain period of time (typically 5–10 years) before it can be deleted. Thus, hospitals must deal with very high storage requirements. Tele-diagnosis is also becoming a popular technique in hospitals. A doctor may ask for advice of a colleague who works in another hospital, and even another country, by means of real-time transmission of medical images and video. Due to the huge amount of data transmitted, high-bandwidth networks are needed in order to maintain the quality of the video and allow a correct diagnosis. In both cases (storage and transmission), compression techniques are used to drastically reduce the amount of information that needs to be handled. Finally, the quality of the compressed data must be good enough to allow for a correct diagnosis when it is reconstructed.

Nowadays, higher compression ratios can be obtained by means of lossy compression techniques (JPEG Wallace, 1991, MPEG Sikora, 1997), but radiologists are very reluctant to use them as they might introduce compression artifacts, which could complicate diagnosis. Normally, doctors prefer to use lossless compression techniques (JPEG-LS JPEG/JBIG, 1992) so that the quality is preserved. However, lossless compression leads to compression ratios which are significantly lower than those achieved by lossy techniques. Thus, compliance with legal demands and keeping medical video for ten years may become prohibitive for most hospitals on account of storage requirements. In addition, the constant increase of the network traffic may hamper the use of tele-diagnosis if images are not sufficiently compressed. All of this makes the research on new lossy compression techniques attractive, especially that oriented to exploiting the behavior of medical video, which is usually encoded in gray scale, using just 1 byte per pixel, and which presents very small interframe variations.

Many video compression schemes have been proposed, but the scheme developed by the moving pictures experts group (MPEG) has emerged as a widely accepted industry standard. However,

* Corresponding author. Tel.: +34 968 36 75 70; fax: +34 968 36 41 51.
E-mail addresses: gbernabe@ditec.um.es (G. Bernabé), jmgarcia@ditec.um.es (J.M. García), pepe.gonzalez@intel.com (J. González).

0164-1212/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2008.09.034

MPEG presents some problems when it is applied to compressing medical video, since it cannot fully exploit the aforementioned characteristics. MPEG-2 applies the discrete cosine transform (DCT) by dividing the image into blocks of 8×8 pixels. This causes the presence of visible artifacts in the border of blocks, which influences the quality of the reconstructed medical video. Also, the Gibbs effect Basith and Done, 1996 is one of the most common artifacts and it also afflicts MPEG. It is caused by the DCT used to digitize chrominance and luminance information. Nowadays, a very attractive area of research involves the proposal and evaluation of different transform functions that may overcome the limitations that the DCT used by MPEG-2 presents for some particular types of video. The Wavelet Transform has emerged as an attractive alternative to the DCT. Wavelet techniques have recently generated much interest in applied areas such as FBI fingerprint compression Brislawn, 1994, modeling the joint statistics of images Simoncelli, 1999, denoising noisy data Donoho, 1993, etc. Also, the wavelet transform has been mainly applied to image compression. Several coders have been developed using the 2D wavelet transform Lewis and Knowles, 1992; Antonini and Barlaud, 1992; Shapiro, 1993. Moreover, the latest image compression standard, JPEG-2000 Marcellin et al., 2000; Santa-Cruz and Ebrahimi, 2000, is also based on the 2D discrete wavelet transform with a dyadic mother wavelet transform. The 2D wavelet transform has also been used for compressing video Hilton et al., 1994. However, three-dimensional (3D) compression techniques seem to offer better results than two dimensional (2D) ones that operate in each frame independently. In Muraki, 1992 and Muraki, 1995, Muraki introduced the idea of using 3D wavelet transform to approximate 3D volumetric data efficiently. This idea was developed by Ihm and Park Ihm and Park, 1998 to achieve a good trade-off among compression ratio, quality and fast random access. Since one of the three spatial dimensions can be considered similar to time, Chen and Pearlman developed a three-dimensional subband coding to code video sequences Chen and Pearlman, 1996, posteriorly improved with an embedded wavelet video coder using 3D set partitioning in hierarchical trees (SPHIT) Kim and Pearlman, 1997; Kim and Pearlman, 2000. Today, the standard MPEG-4 Battista et al., 1999; Battista et al., 2000 supports an ad-hoc tool for encoding textures and still images, based on a wavelet algorithm. In this work, we present and evaluate a new lossy video compression scheme, based on the use of the fast wavelet transform (FWT) Graps, 1995; Stollnitz et al., 1995; Strang and Nguyen, 1997. The work presented in this paper is a major revision and an extension of two previous papers published by the authors in Bernabé et al., 2000 and Bernabé et al., 2001. This new technique achieves high compression ratios with an excellent quality, so that medical doctors cannot distinguish between the original and the reconstructed video. The main differences between our work and previous ones are the following:

- Our proposals are mainly focused on medical video, exploiting its particular properties.
- We propose a new coding scheme for three-dimensional wavelet which exploits both the spatial and the temporal redundancies.
- We evaluate the performance achieved by different wavelet mother functions and the number of steps that this function is applied to, in order to find which provides the best trade-off between quality and compression ratio.
- We propose and evaluate two ways of thresholding: the percentile policy and the discarding of the less significant bits of all wavelet coefficients.
- We propose a quantizer where the number of bits needed by each pixel coefficient to be encoded depends on the layer that the pixel belongs to or the number of wavelet transforms applied on the layer.
- We use the run-length code and the Huffman code in the entropy encoder.
- We propose several improvements to the quantizer and the entropy encoder in order to increase the compression ratio without affecting video quality.

The rest of this paper is organized as follows. Section 2 summarizes the background to wavelets. In Section 3 we describe our proposed coding method using the 3D wavelet transform. We present the main details of each component of the method. Section 4 explains the improvements to enhance our original encoder. Experimental results with some test medical video are analyzed in Section 5. Moreover, this section includes a comparison of our encoder with MPEG-2 and EZW in the coding of two medical video sequences. Section 6 summarizes the work and concludes the paper.

2. The wavelet transform foundations

The basic idea of the wavelet transform is to represent any arbitrary function f as a weighted sum of functions, referred to as wavelets. Each wavelet is obtained from a mother wavelet function by conveniently scaling and translating it. The result is equivalent to decomposing f into different scale levels (or layers), where each level is then further decomposed with a resolution adapted to that level.

In a multiresolution analysis, there are two functions: the mother wavelet and its associated scaling function. Therefore, the wavelet transform can be implemented by quadrature mirror filters (QMF), $G = g(n)$ and $H = h(n)$ $n \epsilon Z$. H corresponds to a low-pass filter, and G is a high-pass filter. The reconstruction filters have impulse response $h^*(n) = h(1-n)$, and $g^*(n) = g(1-n)$. For a more detailed analysis of the relationship between wavelets and QMF see Mallat, 1989.

The filters H and G correspond to one step in the wavelet decomposition. Given a discrete signal, s, with a length of 2^n, at each stage of the wavelet transformation the G and H filters are applied to the signal, and the filter output downsampled by two, thus generating two bands, G and H. The process is then repeated on the H band to generate the next level of decomposition, and so on. This procedure is referred to as the 1D fast wavelet transform (1D-FWT).

It is not difficult to generalize the one-dimensional wavelet transform to the multi-dimensional case Mallat, 1989. The wavelet representation of an image, $f(x, y)$, can be obtained with a pyramid algorithm. It can be achieved by first applying the 1D-FWT to each row of the image and then to each column, that is, the G and H filters are applied to the image in both the horizontal and vertical directions. The process is repeated several times, as in the one-dimensional case. This procedure is referred to as the 2D fast wavelet transform (2D-FWT).

3. The original 3D-FWT encoder

Fig. 1 shows the key processing steps involved in the 3D-FWT-based lossy compression method. We describe each in some detail below.

3.1. Direct 3D-FWT

As in 2D, we can generalize the one-dimensional wavelet transform for the three-dimensional case. Instead of one image, there is now a sequence of images. Thus, a new dimension has emerged, the time (t). The 3D-FWT can be computed by successively applying the 1D wavelet transform to the value of the pixels in each dimension.

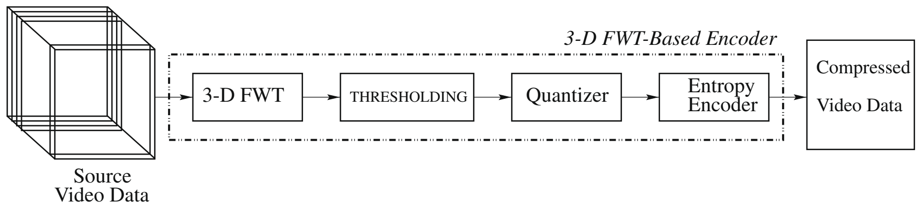

Fig. 1. 3D-FWT-based encoder processing steps.

It is important to note the potential of such a transform for medical video. As pointed out before, medical video sequences have little movement. Consequently, applying the 1D-FWT in the time axis to a sequence divides the original signal into two equal sets of frames, where the first one includes the average signal and the second one represents the details. Due to the lack of movement, the coefficients of the second set of frames are very good candidates to be neglected in the thresholding phase (since their values will be close to zero), thus allowing higher compression ratios.

It is common in wavelet compression to transform the average signal recursively. The number of iterations performed in each dimension depends on several factors, such as the amount of compression desired, the size of the original video and the mother wavelet function. In general, the higher the desired compression ratio, the more times the transform is performed. Note that applying the wavelet transform too many times may have a significant impact on quality. Hence, this parameter must be carefully chosen.

In this paper, we firstly consider Daub-4 Daubechies, 1992 as the mother wavelet function. We have chosen this function because some previous works have proved its effectiveness. However, we have also tried other mother wavelet functions such as Haar, Daub-8, a bathlet wavelet Monro et al., 1996 and a hybrid between Daub-4 (for the x and y dimensions) and Daub-8 (for the time dimension), in order to evaluate the impact of the mother wavelet function on final performance.

3.2. Thresholding

After decorrelating the original frames by applying the 3D-FWT, the next step compresses frames by discarding those coefficients whose value does not provide enough information. In order to do this, a threshold (p) is chosen and the next function ($f_{x,y,t}$) is applied to the decorrelated coefficients ($d_{x,y,t}$):

$$f_{x,y,t} = \begin{cases} 0 & \text{if } d_{x,y,t} < p \\ d_{x,y,t} & \text{if } d_{x,y,t} \geqslant p \end{cases}$$

where p is a x-percentile of the set of all wavelet coefficients. For instance, if we want to apply the 90-percentile, p must be chosen so that 90% of the wavelet values are under p, and thus, according to the previous function, they will set to zero. We will refer to p as the threshold. The choice of a suitable threshold has a significant impact on final compression ratio and quality.

3.3. Quantization and entropy encoding

We need to quantize the non-zero wavelet coefficients before encoding them. Basically, the task performed in this step involves the transformation of the floating point coefficients into unsigned integer coefficients. To carry out this conversion, it is necessary to specify how many bits are needed to encode each of the wavelet layers.

Each frame is independently processed. The number of required bits depends on the number of wavelet transforms applied. The more times the wavelet is performed the higher the coefficients are and the more bits are needed. Furthermore, the number of bits needed by each pixel coefficient to be encoded also depends on the layer that this pixel belongs to. In other words, we assign more bits to the low-pass layers since coefficients have a value greater than those belonging to the high-pass layers.

On the other hand, we have carried out another improvement after the quantization step. We have combined the previous thresholding policy with another one that discards the less significant bits of all wavelet coefficients after the quantization step. This yields a higher compression ratio, and only slightly decreases quality.

Finally, an entropy encoding step is performed: a run-length compression is applied to the binary representation of the integer coefficients, exploiting the presence of a large number of zeros present in the video. After applying the binary run-length, another entropy encoder is applied, the Huffman encoding with a 128-symbol alphabet, in order to increase further the compression ratio without affecting the video quality.

4. The enhanced 3D-FWT encoder

In order to increase the compression ratio, while maintaining the video quality, we have developed several improvements in the quantization and the entropy encoder.

4.1. 3D-conscious run-length

In order for the binary run-length be more effective, the highest possible number of zeros from the decorrelated image must be placed in a row order. In the previous approach, the coefficients of each independent frame are processed in a similar way to JPEG aligns coefficients before applying the entropy encoder. However, since the 3D-FWT is applied, the original sequence is also decorrelated in the time axis. Therefore, there is a higher likelihood of having longer chains of zeros if coefficients are processed taking into account the time dimension, particularly in the case of the H bands. We refer to this technique as 3D-conscious run-length.

This enhancement takes into account the different sub-cubes achieved in the application of the 3D-FWT in order to apply the binary run-length following the time dimension order.

Moreover, the quantizer also takes into account the number of times that the 3D-FWT has been applied and assigns different number of bits depending on the sub-cube.

In general, the number of required bits depends on the number of the wavelet transformations performed to each sub-cube. For instance, for a video of 64 frames of 512×512 pixels, the first 3D-FWT is applied over the whole sequence. The second application is just performed on the sub-cube consisting of 32 frames of 256×256 pixels (video reference). This sub-cube will generate higher wavelets coefficients, which need more bits in the quantization process.

4.2. Hexadecimal coding

In the former approach, the run-length compressed chains of up to 128 zeros, because the Huffman code used 128 symbols. This limits the enhanced approach, since chains of thousands of zeros may appear after traversing the video, given the time dimension. We propose to represent the chains of zeros by using the hexadecimal representation of their length in the following way: A first symbol (hexadecimal number) indicates the number of hexadecimal symbols used to encode the length of the chain. Next, the following hexadecimal symbols code this length. Thus, chains of millions of zeros can be represented, and the Huffman code will be reduced to 16 symbols. For example, if the decorrelated video has 15,503 zeros in a row, the former run-length code needs 366 bytes to encode it. This length is now represented as 43C8F ($0 \times 3C8F = 15{,}503$), drastically reducing the number of bytes needed to encode it (2.5 bytes). This hexadecimal coding can be improved with some extensions.

Firstly, chains of up to seven zeros in a row need one byte to be coded: the first 4-bit symbol is always 1 and the second 4-bit symbol ranges from 1 to 7. We have verified that the largest chain of zeros does not need more than seven hexadecimal symbols. If we fix this bound, chains of more than 268 millions of zeros can be compressed in 4 bytes. The improvement is based on representing with just one symbol (1/2 byte) the chains of zeros ranging from one to seven. If the first bit is zero, the symbol indicates the number of hexadecimal symbols that code the chain of zeros, whereas if the first bit is one, the following three bits directly indicate a length between 1 and 7. Secondly, the free 8 hexadecimal symbol appears to indicate a chain of ones, so a hexadecimal symbol is used to determine the number of hexadecimal symbols that follow and, finally, the symbols representing the number of ones in a similar way as performed with the chains of zeros.

4.3. Applying arithmetic coding

In the previous approach, we have used the Huffman code in the entropy encoder after the binary run-length coding. Arithmetic coding appears to be much more effective to complete our encoder, since it bypasses the idea of replacing an input symbol with a specific code. Instead, it replaces a stream of input symbols with a single floating point output numbers Nelson and Gailly, 1996. In our encoder, the arithmetic coding has been applied after the 3D-conscious run-length and the hexadecimal coding. Therefore, it has been implemented to work with only sixteen symbols in hexadecimal.

5. Performance evaluation

In this section we evaluate the performance of our proposed compression scheme for different medical scenarios.

5.1. Measuring the quality of compressed video

We have used the following measurements to evaluate the quality of the compression process using the 3D-FWT:

(a) Compression ratio.
(b) Quality of the decoded video sequence.

The compression ratio is computed as the ratio between the original and the compressed data sizes.

The numerical evaluation of the quality of the compressor is achieved by computing the peak signal to noise ratio (PSNR) in the reconstructed video. The PSNR of a reconstructed video has been calculated by computing the arithmetic mean of the PSNR for all the frames of the video. In addition to the PSNR value, we have visually tested the quality of reconstructed video.

5.2. Environment workbench

The evaluation has been carried out on an Intel Pentium-III 450 MHz bi-processor with 256 Mbytes of RAM, but we have only used one processor. The operating system was Linux 2.2.12–20 smp. The entire video compressor and decompressor have been written in the C programming language.

As input tests, we have compressed and decompressed three medical video sequences coded in gray scale (8 bit per pixel), *heart* and *catheter* video medical sequences of 256 frames of 512×512 pixels and a *hand* video medical sequence of 256 frames of 256×256 pixels. These videos have been divided into four blocks of 64 frames.

The *heart* sequence shows the movement of the heart and surrounding regions while the *catheter* sequence shows the introduction of a catheter. Moreover, the *hand* sequence shows a human hand with slight movement of the fingers.

5.3. Evaluation of the original 3D-FWT encoder

We have proved that the order to apply the wavelet transform in different dimensions has no affect on the trade-off between compression ratio and quality. In order to exploit the scarce presence of movement, we propose to apply the 3D-FWT by performing the wavelet transform first in the time dimension and then in the space dimensions. In this way, the first frames accumulate the average signal whereas the rest represent the detail (which will probably be discarded in the following steps of the coder process).

Then, we compare the performance of the different mother wavelet functions evaluated in this work: Haar, Daub-4, Daub-8, bathlet wavelet Monro et al., 1996, and a hybrid between Daub-4 and Daub8 for the *heart*, *catheter* and *hand* sequences on the original 3D-FWT Encoder. Tables 1–3 show the results for these evaluations. Each cell in the table shows the compression ratio (top value) and the PSNR (bottom value). These results are presented for different configurations with the following parameters:

- Parameter b represents the number of discarded bits in the quantization process.
- Parameter per-X indicates the X-percentile that has been used for the thresholding step.

Table 1
Experimental results with Haar, Daub-4 and Daub-8 for heart.

Heart	Haar		Daub-4		Daub-8	
	2-*t*	3-*t*	2-*t*	3-*t*	2-*t*	3-*t*
b = 2-per93	7,53	7,68	7,76	8,00	8,14	8,20
	39,77	34,60	39,97	36,63	38,95	39,15
b = 2-per95	9,49	9,76	9,68	9,78	9,87	9,84
	38,17	34,06	37,83	35,77	37,28	37,68
b = 2-per96	10,93	10,76	11,6	11,08	11,71	11,48
	37,14	33,75	36,61	35,20	36,05	36,72
b = 2-per97	14,40	13,20	14,2	13,73	14,10	13,96
	35,25	33,21	34,78	34,30	34,65	35,47
b = 3-per93	9,33	9,61	9,49	9,98	9,89	10,17
	39,31	34,41	39,15	36,34	38,52	38,69
b = 3-per95	11,72	12,12	11,78	12,10	11,91	12,24
	37,90	33,93	37,52	35,59	37,02	37,38
b = 3-per96	13,31	13,48	13,81	13,53	13,82	13,95
	36,95	33,63	36,42	35,06	35,89	36,52
b = 3-per97	16,76	16,08	16,57	16,39	16,64	16,54
	35,18	33,13	34,68	34,22	34,09	35,37

Table 2
Experimental results with Haar, Daub-4 and Daub-8 for catheter.

Catheter	Haar		Daub-4		Daub-8	
	2-*t*	3-*t*	2-*t*	3-*t*	2-*t*	3-*t*
b = 2-per93	9,55	9,51	9,69	10,07	9,62	10,01
	40,01	40,53	38,88	39,41	38,87	39,43
b = 2-per95	11,74	12,06	11,95	12,09	11,84	11,96
	37,95	38,83	36,58	37,65	36,80	37,84
b = 2-per96	14,13	13,95	14,15	14,04	14,17	13,94
	36,10	37,76	35,22	36,46	35,28	36,71
b = 2-per97	16,58	16,86	16,68	17,18	16,71	17,38
	34,27	36,22	33,66	35,14	33,49	35,25
b = 3-per93	11,58	11,75	11,66	12,23	11,61	12,13
	39,58	39,96	38,45	38,98	38,46	39,02
b = 3-per95	14,29	14,94	14,47	14,87	14,33	14,69
	37,72	38,50	36,34	37,39	36,57	37,57
b = 3-per96	16,60	17,20	16,68	17,30	16,60	17,21
	36,01	37,52	35,09	36,27	35,17	36,52
b = 3-per97	19,33	20,11	19,49	20,53	19,42	20,60
	34,23	36,11	33,48	35,05	33,44	35,17

Table 3
Experimental results with Haar, Daub-4 and Daub-8 for hand.

Hand	Haar		Daub-4		Daub-8	
	2-*t*	3-*t*	2-*t*	3-*t*	2-*t*	3-*t*
b = 2-per95	9,46	10,05	9,46	9,67	9,68	10,06
	44,03	44,74	44,54	45,29	44,36	44,78
b = 2-per96	10,58	10,84	10,64	11,01	10,98	11,57
	42,78	44,08	43,48	44,37	43,32	43,93
b = 2-per97	12,73	12,39	13,05	12,43	13,46	13,04
	41,08	42,75	41,86	43,38	41,73	42,99
b = 2-per98	17,75	15,39	18,36	15,99	18,72	16,56
	37,63	41,02	38,73	41,68	38,58	41,44
b = 3-per95	11,44	12,87	11,65	13,23	11,93	13,58
	43,28	43,55	43,47	43,75	43,30	43,44
b = 3-per96	12,76	13,33	12,80	13,75	13,17	14,15
	42,31	43,29	42,79	43,53	42,66	43,23
b = 3-per97	14,92	15,21	15,44	15,22	15,87	15,98
	40,87	42,26	41,46	42,86	41,35	42,50
b = 3-per98	20,29	18,35	20,97	19,25	21,34	19,90
	37,58	40,80	38,60	41,41	38,45	41,19

- Parameter 2-*t* or 3-*t* indicates whether the wavelet transform has been applied 2 or 3 times in each of the axis (time, x, y). We have discarded the results obtained with just one step, since the compression ratio obtained is lower than for the other configurations. Also, application of the wavelet transform four or more times considerably decreases the quality, and, therefore, these results are not considered in this work which are unacceptable from the point of medical view.

By comparing the number of times the 3D-FWT is applied for the *heart* sequence, we can observe that applying the transform 3 times increases the compression ratio but clearly affects the PSNR, leading to poor quality video sequences (PSNR decreases by 1 and 5 dB), so we consider that for *this medical video* more than 2 applications of the 3D-FWT are not worth performing. We have accepted the quality is *excellent* when PSNR is around 41 for all frames because this implies that there are no differences between the original and the reconstructed sequences. On the other hand, the quality is *good* or *acceptable* when PSNR is around 38 for all frames and the sequence could be accepted by the medical community.

For the *catheter* sequence, the application of the transform 3 times increases the compression ratio and the quality of the video sequences. This is due to the *catheter* sequence has less interframe variations than the *heart* sequence, and therefore our coding scheme achieves a better trade-off between compression ratio and quality.

The *hand* sequence has less interframe variations than the others ones, and therefore our coding scheme achieves an excellent compression rate and quality, which confirms the potential of the FWT for videos that barely contain movement. In this sequence, we have obtained that application of the transform 3 times can increase or decrease the trade-off between the compression ratio and the quality. It also depends on others factors, such as the percentile used or the number of bits discarded in the quantization phase.

For the *heart* and *hand* sequences, the mother wavelet function Daub-4 obtains the best trade-off between compression ratio and quality for all the analyzed configurations. For the *catheter* sequence, the Haar wavelet function slightly outperforms the Daub-4. However, the trade-off between the compression ratio and the quality of the Daub-4 is almost the same as the Haar. The quality of the Daub-8 is very similar to Daub-4 for the three sequences. However, taking into account the execution time, the Daub-8 implementation is much more costly (around twenty percent), since it uses more filters, which causes an increment in the number of floating point operations.

In three sequences, we have also evaluated the performance of the bathlet mother wavelet function and the hybrid wavelet between Daub-4 and Daub-8. Results for these particular implementations are not shown in this work since they are very similar to the results obtained by the Daub-4 function.

Through analyzing the results for the different percentiles and number of bits discarded we can also conclude that the more bits and coefficients that are discarded, the more compression ratio is achieved. However, as the percentile increases, the PSNR drops significantly, so an optimal configuration must be carefully chosen in order to obtain a reasonable compression ratio without influencing the quality. From our results, we can conclude that a per95 or per96 with $b = 3$, is the best choice, because 3 bits discarded improve the compression ratio without affecting the quality of the reconstructed video.

5.4. Evaluation of the enhanced 3D-FWT encoder

We have compared the original and the enhanced 3D-FWT encoders on the *heart*, *catheter* and *hand* video medical sequences. Figs. 2–4 show the compression ratio achieved by the two wavelet-based encoders. Results are presented for two and three applications of the 3D-FWT, and considering percentiles ranging from 93 to 98 in the thresholding phase. Peak signal to noise ratio (PSNR) is the same for both the original and the enhanced techniques for each configuration as the enhancements proposed in Section 4 optimize the entropy encoder and do not affect the quality of the reconstructed sequences. The mother wavelet function is the Daub-4.

We can observe that the enhanced method clearly improves the compression ratio for all configurations. Also, since the number of discarded coefficients and the percentile increase, the difference between the original and the enhanced method also increases significantly, because the improvements seek to exploit the high presence of zeros. For instance, in the *heart* sequence, choosing a 95-percentile, discarding 3 bits and applying the FWT twice times, means the compression ratio is increased by 57%, whereas for a 97-percentile, discarding 3 bits and applying the FWT three times, the compression ratio is increased by 79%. The enhanced method obtains better results for the *hand* sequence than the *heart* and *catheter* sequences because the *hand* sequence experiences less interframe variations, which confirms the potential of the enhanced method with sequences that hardly have any movement. For example, for a 97-percentile, discarding 3 bits and applying

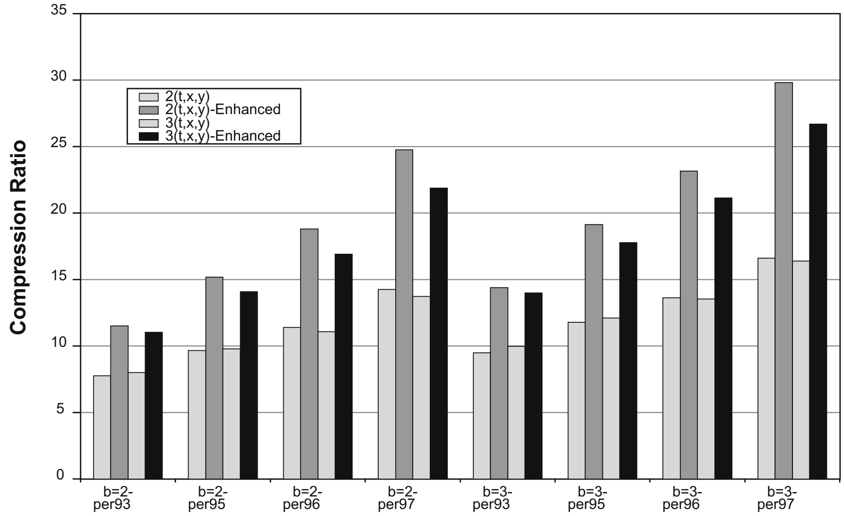

Fig. 2. Compression ratios for heart.

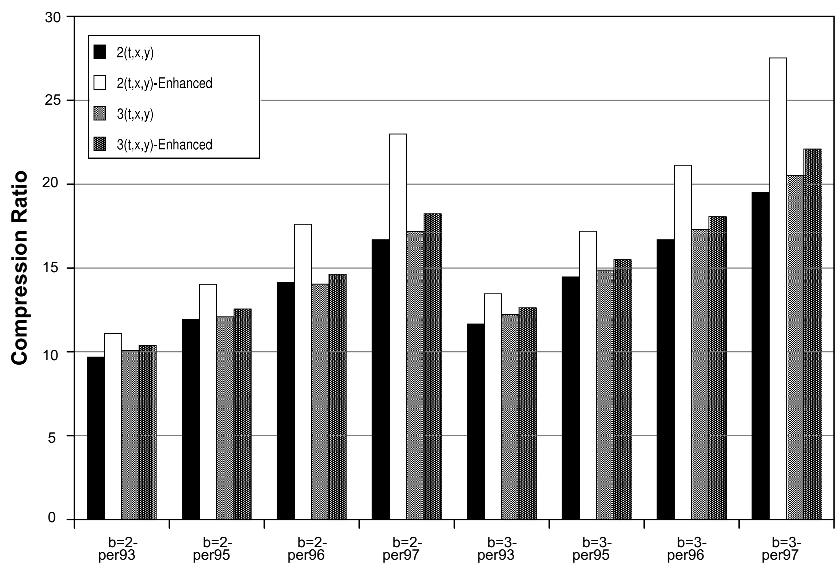

Fig. 3. Compression ratios for catheter.

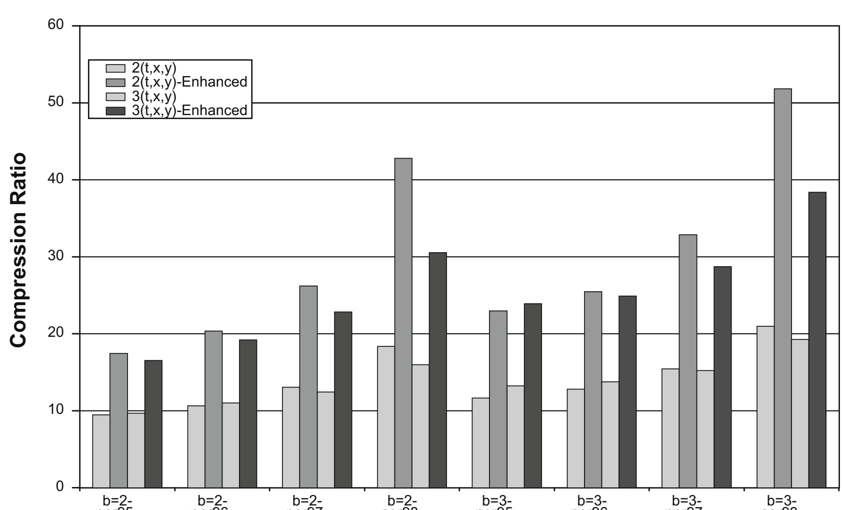

Fig. 4. Compression ratios for hand.

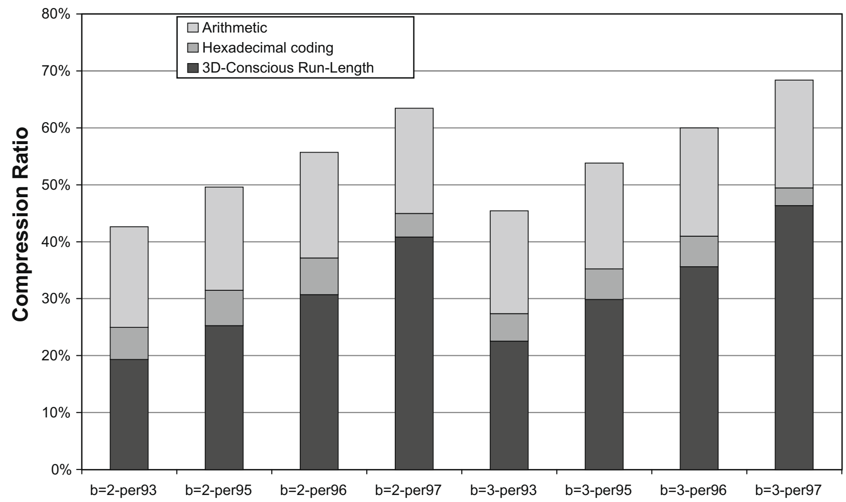

Fig. 5. Contribution of each enhancement.

the FWT twice times the compression ratio is increased by 41%, 79% and 113% in *catheter*, *heart* and *hand* sequences.

In Fig. 5, we present the contribution to the overall performance improvement of each of the enhancements presented in this work for the *heart* sequence. We can observe that the 3D-conscious run-length is responsible for most of the improvement achieved. In fact, by applying just this technique, the original compression ratio is increased between 20% and 40%. This supposes a contribution between 45% and 68% to the overall performance. This confirms that, if coefficients are processed taking into account the time dimension, there are many long-length chains of zeros that are effectively compressed, even though Huffman is used as entropy encoder. The Hexadecimal coding provides a contribution that ranges from 5% to 13% to the overall improvement. Despite the fact that this technique does not seem to provide significant benefits by itself, the absolute contribution (around 5%) is maintained in all configurations and does not depend on the length of the chains of zeros. In addition, this enhancement benefits the subsequent application of Arithmetic code instead of Huffman in the entropy encoder. Finally, the use of Arithmetic code provides an absolute improvement of around 20%, better than some previously reported results, which show improvements ranging from 5% to 10% when Huffman code is replaced by Arithmetic code Smith, 1997. This is due to the synergistic effect obtained by previously applying Hexadecimal coding.

5.5. Comparison with MPEG-2 and EZW

The video sequences have also been compressed using EZW Shapiro, 1993, in the quantization process, and MPEG-2. We force EZW, the original 3D-FWT coder and MPEG-2 to obtain the same quality as the proposed enhanced 3D-FWT.

Fig. 6 shows the compression ratio for the *heart* and *hand* sequences for the four methods. For each video, two set of bars are shown. The first set is obtained when the quality is forced to be excellent (PSNR around 41 for all frames). This implies that there are no differences between the original and the reconstructed sequences. On the other hand, the second set is forced to be of good quality (PSNR of 38).

First, we can observe that EZW outperforms both the original 3D-FWT coder and MPEG-2 for both sequences and for both qualities. This confirms some problems of the standard MPEG-2 in coding medical video sequences. If we configure MPEG-2 to obtain the top compression ratio irrespective of the quality, the compression ratios are significantly better than the wavelet-based encoders. However, the quality of the reconstructed sequences is very poor

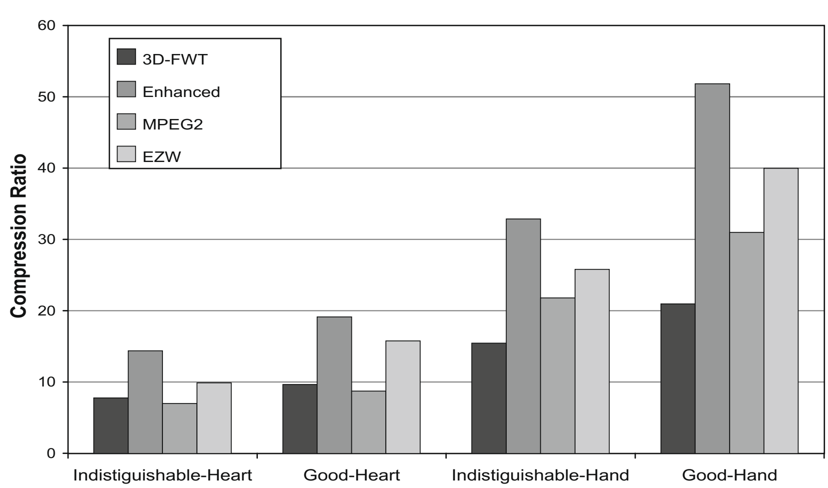

Fig. 6. Compression ratios of 3D-FWT, enhanced, MPEG2 and EZW for both sequences.

and we can observe the presence of the blockiness effect. Also, the MPEG-2 algorithm is much more complex since, as well as applying a transform (the DCT instead of the FWT), it also performs motion compensation, which slows down the coding process. The quantizer EZW is also much more complex, but it uses the FWT, performing the tree processing algorithm, which drops down the coding process. The enhanced 3D-FWT clearly outperforms the EZW, for all configurations as well, increases the compression ratio ranging from 21% to 54% in the *heart* sequence and 30% to 34% in the sequence *hand* for the *excellent* and *good* quality, respectively. These results confirm that the proposed improvements efficiently exploit the properties of the wavelet transform to code medical video sequences.

6. Conclusions

In this work, we have presented and evaluated a new compression scheme based on applying the 3D Fast Wavelet Transform to medical video, and based on the following proposals:

- In order to exploit the scarce presence of movement and the spatial and temporal redundancies, we have applied the 3D-FWT performing the wavelet transform first in the time dimension and then in the space dimensions.
- We have evaluated the performance achieved by different wavelet mother functions (Haar, Daub-4, Daub-8, Bathlet-4 and a hybrid between Daub-4 and Daub-8), and the number of steps that this function has been applied to, in order to find which provides the best trade-off between quality and compression ratio. The Daub-4 has obtained the best results.
- We have proposed and evaluated two ways of thresholding: the percentile policy and the discarding of the less significant bits of all wavelet coefficients.
- We have proposed a quantizer where the number of bits needed by each pixel coefficient to be encoded depends on the layer that this pixel belongs to.
- We have developed an entropy encoder based on a binary run-length code and a Huffman code. This entropy encoder increases the compression ratio without affecting the video quality.

We have analyzed the performance of our proposal in terms of both compression ratio and quality (PSNR), and we have visually confirmed these results.

We have presented and evaluated several improvements to a compression scheme based on applying the 3D-FWT, with the focus on coding medical video. These improvements achieve a better compression ratio, ranging from 40% to 70%, maintaining the good quality obtained in the original encoder. The trade-off achieved between compression ratio and quality is excellent, especially when compared with the compression ratio and quality achieved by the standard MPEG-2 and the quantizer EZW. There is no extra cost in computation time, and better compression ratios are obtained, up to 119% and 46%, respectively, for the same PSNR.

Acknowledgement

The authors would like to thank the anonymous reviewers for their detailed comments and valuable suggestions, which have been helped to improve the quality of the paper. This work has been jointly supported by Spanish Ministry of Ciencia e Innovación under grant "TIN2006-15516-C04-03" and European Commission FEDER funds under grant "Consolider Ingenio-2010 CSD2006-00046". The video sequences were donated by the Hospital Recoletas, Albacete, Spain.

References

Antonini, M., Barlaud, M., 1992. Image coding using wavelet transform. IEEE Transactions on Image Processing 1 (2), 205–220.
Basith, S.A., Done, S.A., 1996. Digital video, mpeg and associated artifacts. Technical Report, Department of Computing and Electrical Engineering, Imperial College.
Battista, S., Casalino, F., Lande, C., 1999. Mpeg-4: A multimedia standard for the third millennium, part 1. IEEE Multimedia 6 (4), 74–83.
Battista, S., Casalino, F., Lande, C., 2000. Mpeg-4: A multimedia standard for the third millennium, part 2. IEEE Multimedia 7 (1), 76–84.
Bernabé, G., González, J., García, J.M., Duato. J., 2000. A new lossy 3-D wavelet transform for high-quality compression of medical video. In: Proceedings of IEEE EMBS International Conference on Information Technology Applications in Biomedicine, pp. 226–231.
Bernabé, G., González, J., García, J.M., Duato, J., 2001. Enhancing the entropy encoder of a 3D-FWT for high-quality compression of medical video. In: Proceedings of IEEE International Symposium for Intelligent Signal Processing and Communication Systems.
Brislawn, C.M., 1994. Fingerprints go digital. Notices of the AMS 42 (11).
Chen, Y., Pearlman, W.A., 1996. Three-dimensional subband coding of video using the zero-tree method. Proceedings of SPIE-Visual Communications and Image Processing, 1302–1310.
Daubechies, I., 1992. Ten lectures on wavelets. Society for Industrial and Applied Mathematics.
Donoho, D., 1993. Nonlinear wavelet methods for recovery of signals, densities, and spectra from indirect and noisy data, Different Perspectives on Wavelets. In: Proceeding of Symposia in Applied Mathematics, 47, pp. 173–205.
Graps, A., 1995. An introduction to wavelets. IEEE Computational Science and Engineering 2 (2).
Hilton, M.L., Jawerth, B.D., Sengupta, A., 1994. Compressing still and moving images with wavelets. Multimedia Systems 2 (3).
Ihm, I., Park, S., 1998. Wavelet-based 3D compression scheme for very large volume data. Graphics Interface '98., 107–116.
JPEG/JBIG, I.J.W., 1992. Fcd 14495 lossless near-lossless coding of continuous tone still images (jpeg-ls). IEEE Transactions on Image Processing 1 (2), 244–256.
Kim, B.-J., Pearlman, W.A., 1997. An embedded wavelet video coder using three-dimensional set partitioning in hierarchical trees (spiht). Proceedings of Data Compression Conference.
Kim, Y., Pearlman, W.A., 2000. Stripe-based spiht lossy compression of volumetric medical images for low memory usage and uniform reconstruction quality. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing.
Lewis, A.S., Knowles, G., 1992. Image compression using the 2-D wavelet transform. IEEE Transactions on Image Processing 1 (2), 244–256.
Mallat, S., 1989. A theory for multiresolution signal descomposition: The wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (7), 674–693.
Marcellin, M.W., Gormish, M.J., Bilgin, A., Boliek, M.P., 2000. An overview of JPEG-2000. Data Compression Conference. March.
Monro, D.M., Bassil, B.E., Dickson, G.J., 1996. Orthonormal wavelets with balanced uncertainty. School of Electronic and Electrical Engineering. University of Bath, England (ICIP).
Muraki, S., 1992. Approximation and rendering of volume data using wavelet transforms. Proceedings of Visualization, 21–28. October.
Muraki, S., 1995. Multiscale volume representation by a dog wavelet. IEEE Transactions on Visualization and Computer Graphics 1 (2), 109–116.
Nelson, M., Gailly, J.-L., 1996. The Data Compression Book, M&T Books, 2 ed.
Santa-Cruz, D., Ebrahimi, T., 2000. A study of jpeg 2000 still image coding versus others standards. In: Proceedings of the X European Signal Processing Conference, September 2000.
Shapiro, J.M., 1993. Embedded image coding using zerotrees of wavelets coefficients. IEEE Transactions on Signal Processing 41 (12), 3445–3462.
Sikora, T., 1997. MPEG Digital Video Coding Standards. McGraw-Hill Company.
Simoncelli. E.P., 1999. Modeling the joint statistics of images in the wavelet domain. Spie 44th Annual Meeting, 3813.
Smith, S.W., 1997. The Scientist and Engineer's Guide to Digital Signal Processing. California Technical Publishing.
Stollnitz, E.J., DeRose, T.D., Salesin, D.H., 1995. Wavelets for computer graphics: A primer, part 1. IEEE Computer Graphics and Applications 15 (3), 76–84.
Strang, G., Nguyen, T., 1997. Wavelets and Filter Banks. Wellesley-Cambridge Press.
Wallace, G.K., 1991. The jpeg still picture compression standard. Communications of the ACM. April.

Gregorio Bernabé was born in Antibes (Alpes Maritimos, France) on 21 November 1974. He received the M.S. and Ph.D. degrees in Computer Science from the University of Murcia (Spain) in 1997 and 2004, respectively. In 1998, he joined the Computer Engineering Department of the University of Murcia, and became an Associate Professor in May 2004. From 2006 to 2008 he served as Director of the Computer Engineering Department. His current reserach interests include the development of optimizations to improve the performance of multimedia applications in chip-multiprocessor (CMP) architectures.

Jose M. Garcia was born in Valencia, Spain on 9 January, 1962. He received the M.S. and the Ph.D. degrees in Electrical Engineering from the Techical University of

Valencia (Valencia, Spain), in 1987 and 1991, respectively. He is currently serving as Dean of the school of Computer Science at the University of Murica (Spain). He is professor in the Department of Computer Engineering, and also Head of the Research Group on Parallel Computer Architecture. He was developed several courses on Computer Structure, Peripheral Devices, Computer Architecture and Multicomputer Design. His current research interests lie in high-performance coherence protocols for Chip Multiprocessors (CMPs) and shared-memory multiprocess or systems, and high-speed interconnection networks. He has published more than 110 refereed papers in different journals and conferences in these fields.

José Gonzalez received the M.S. and Ph.D. degrees from the Universitat Politecnica de Catalunya (UPC). In January 2000, he joined the Computer Engineering Department of the University of Murcia, Spain, and became an Associate professor in June 2001. In March 2002, he joined the Intel Barcelona Research Center.