Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 69384, 11 pages
doi:10.1155/2007/69384

Research Article

Development and Evaluation of High-Performance Decorrelation Algorithms for the Nonalternating 3D Wavelet Transform

E. Moyano-Ávila,[1] F. J. Quiles,[2] and L. Orozco-Barbosa[2]

[1] *Departamento de Tecnologías y Sistemas de Informaciòn, Universidad de Castilla-La Mancha, 45071 Toledo, Spain*
[2] *Departamento de Sistemas Tecnologías, Universidad de Castilla-La Mancha, 02071 Albacete, Spain*

Received 2 September 2006; Revised 30 December 2006; Accepted 27 March 2007

Recommended by Erwin De Kock

We introduce and evaluate the implementations of three parallel video-sequences decorrelation algorithms. The proposed algorithms are based on the nonalternating classic three-dimensional wavelet transform (3D-WT). The parallel implementations of the algorithms are developed and tested on a shared memory system, an SGI origin 3800 supercomputer making use of a message-passing paradigm. We evaluate and analyze the performance of the implementations in terms of the response time and speed-up factor by varying the number of processors and various video coding parameters. The key points enabling the development of highly efficient implementations rely on the partitioning of the video sequences into groups of frames and a workload distribution strategy supplemented by the use of parallel I/O primitives, for better exploiting the inherent features of the application and computing platform. We also evaluate the effectiveness of our algorithms in terms of the first-order entropy.

Copyright © 2007 E. Moyano-Ávila et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

In recent years, the rapid growth in the field of medical imaging has enabled the development of several new classes of digital images, videos, or image sequences. Visual data applications are characterized by their stringent requirements in terms of storage, processing, and transmission. The use of video compression techniques can significantly reduce the storage and communications requirements of video applications. However, the compression of visual data is a computationally intensive. In this context, the use of parallel high-performance computer architectures can prove to be an effective solution particularly when large volumes of visual data have to be processed [1, 2].

In this paper, we evaluate the parallel implementations of three image-sequences decorrelation algorithms having been recently introduced in one of our previous works [3]. The algorithms had been developed around the nonalternating classic wavelet-based transform algorithms versus the Classic method [4]. The development of these novel algorithms has been motivated by the need of coping with the stringent requirements of visual data: huge storage and high processing demands [3]. Due to the inherent characteristics of the video sequences of our interests, that is, angiograms, we have partitioned the video sequences into group of frames. This partitioning scheme has proven to be highly effective in reducing the memory resources required for evaluating the 3D-WT [5].

We experimentally evaluate the performance of our proposals in a shared-memory multiprocessor system making use of a message-passing programming paradigm, built around the standard message passing interface: MPI and its associated MPI-IO subsystem [6, 7]. In order to further improve the performance of the actual implementation, we have paid particular attention to the workload distribution allowing us to reduce the communication steps and number of I/O operations: two key issues when dealing with applications requiring the sharing and exchange of large volumes of data distributed among the multiple elements of a multiprocessor platform.

In the following, the paper is organized as follows. Section 2 reviews the principles of the 3D-WT. We briefly overview the fragmentation scheme and the principles of the classic and nonalternating classic 3D-WT: two relevant elements used in the design of our high-performance video decorrelation algorithms. Section 3 overviews the related work,

both in the areas of video processing and parallel processing of video sequences. Section 4 reviews the operation of our algorithms. Section 5 describes the parallel implementation of the decorrelation algorithms with particular emphasis on the workload distribution scheme employed. Section 6 reports the experimental results in terms of the processing time, speed-up, and quality of the signal achieved by making use of a parallel multiprocessor system. Finally, our conclusions are given in Section 7.

2. 3D WAVELET TRANSFORM

Over the past few years, there have been numerous reports on the use of wavelets in many fields. wavelet-based methods are well suited for a variety of data processing tasks, and especially for image and video compression [1, 5].

2.1. Wavelet analysis and the 3D wavelet transform

Wavelet analysis procedures involve the decomposition of a signal into scaled and shifted signals. In this way, the WT analyzes a signal at different frequency bands with different resolutions by decomposing it into a coarse approximation and detail information. An efficient way of implementing a discrete wavelet transform is possible by using filter banks. This scheme has been developed by Mallat [8]. The multiresolution decomposition of the signal into different frequency bands is obtained by successive highpass and lowpass filtering. The lowest resolution band includes the high-magnitude approximation coefficients of the signal. Most of the energy is captured by those coefficients while most of the detail coefficients have small or even null magnitudes. After filtering, the signal can be subsampled by two. This is called one level of decomposition. Every decomposition level halves the spatial/time resolution since only half the number of samples characterizes the complete signal. However, this operation doubles the frequency resolution. This procedure is also known as subband coding and can be successively repeated for further decomposition.

2.2. The nonalternating versus alternating classic 3D-WT

The 3D-WT can be performed by applying three separate 1D-WT along the three dimensions of a video sequence: the time dimension, the horizontal (rows), and the vertical (columns) dimensions of each frame. The temporal one-dimensional unitary filter is applied to the same pixel in every frame of the sequence [9]. The process of applying the 3D-WT to all the three dimensions defines a decomposition level. Thereafter, the same procedure can be applied to the resulting coarse scale approximation to further decompose the sequence. This procedure is carried out as many times as levels have been predefined.

The 3D-WT of a video sequence can be constructed using the 1D wavelet transform in two ways: the classic decomposition and the nonalternating classic decomposition. Figure 1 shows the classic (Figure 1(a)) and the nonalternating classic Figure 1(b) 3D-WT decomposition schemes [4].

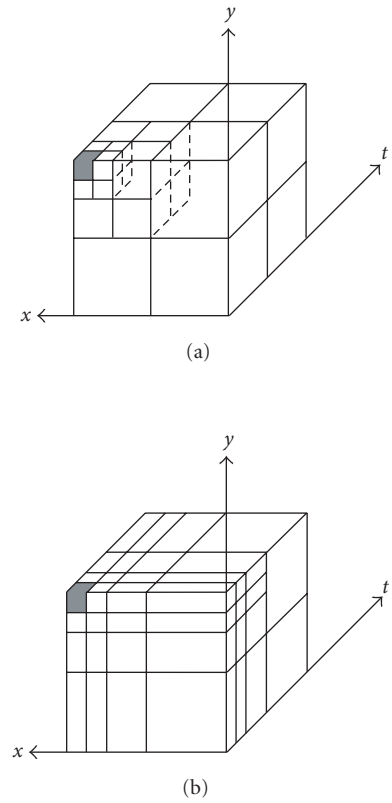

Figure 1: (a) Classic and (b) nonalternating classic 3D wavelet decomposition schemes.

The black cube in both schemes depicts the lowest frequencies in a group of consecutive frames, in the time domain, and the remaining cubes or bands show the details at 3D wavelet decomposition.

The classic decomposition is computed by applying the 1D-WT transform alternating between frames, rows and columns of the video sequence for each decomposition level (see Figure 1(a)). This way to compute the classic 3D-WT is not efficient overall since it imposes stringent requirement on the memory requirements. That is to say, all the frames in the sequence should be readily available to perform the temporal wavelet transform. In general, this is unfeasible to do when the amount of frames in a video sequence is large. Even if this requirement is lifted by temporally fragmenting the video sequence (see Section 2.3), the fact of alternating from the time to the horizontal and the vertical domains and back makes it unsuitable for a straightforward and efficient parallelization. In this case, the data to be processed at each change will have to be redistributed at the beginning of each level and change of domain, that is, from frames to rows, from rows to columns, from columns to frames, and so on.

The nonalternating classic algorithm is a different way to perform the wavelet transform, whose main feature is to carry out all decomposition levels over a data dimension before applying them on to the next data dimension (see Figure 1(b)). This means that the WT is applied by dimension but not by levels. That is to say, the nonalternating

classic method of the 2D-WT involves the data transformation on the row domain at all levels before proceeding with the data decomposition on the column domain.

The nonalternating classic WT takes advantage of having a particular dimension data in memory to perform all the decomposition levels. This avoids changing the context alternatively on each decomposition level. The disadvantage is the higher number of samples to process at each level (except the first level) because of the higher size of low frequency bands. However, we will show that the use of a proper workload mechanism may significantly contribute to improve the overall performance of the nonalternating classic algorithm, in terms of the processing time, speed-up, and the quality of the signal.

2.3. GOF-based 3D-WT

One of the major challenges to overcome when computing the 3D-WT has to do with the development of efficient algorithms capable of handling video sequences composed of a large number of video frames. In particular, computing the temporal WT requires the simultaneous access to all the frames in the sequence. In general, this is unfeasible when dealing with video sequences composed of a large number of images. Yet, the use of large virtual memory space may not provide a satisfactory solution due to the overhead introduced by the paging mechanisms used to make all the required information available to the processor. This problem can be solved by fragmenting the sequence into independent units for its temporal decomposition: group of frames (GOF).

The GOF-based 3D-WT overcomes the huge memory requirements required for processing the whole sequence. This grouping technique has proven to be particularly useful for the processing of video sequences characterized by low motion, that is, having few differences among consecutive frames [5]. This is particularly true in medical sequences, such as angiograms. In these cases, the grouping of a larger number of frames in a GOF allows to capture most temporal redundancies and increase the coding efficiency.

After carrying out the temporal decomposition over a GOF, a 2D wavelet transform can be applied to the relevant frames in the group, according to the decomposition level. This latter task is only performed to the temporal low frequency bands. All the decomposition levels must be applied to the current GOF. This process finishes when all GOFs in the original sequence are transformed by the predetermined number of levels.

3. RELATED WORK

In the past few years, there has been an increasing interest on the parallel computation of various signal processing transforms, such as the FFT, DCT, and wavelet transforms. Some efforts have focused on special architectures for the fast computation of some of these transforms. For instance, in [10] Modarressi and Sarbazi-Azad have proposed an algorithm for the calculation of the 3D DCT over a k-ary n-cube architecture enabling its computation on real time. However they do not present a numerical evaluation of the proposed scheme. They have also overlooked the I/O processing, an important issue when dealing with huge volumes of data. In fact, many studies have focused on reducing the impact of the I/O methods being used over the performance of parallel algorithms involving the processing of large data sets. Towards this end, Yu and Ma have studied various I/O methods to be integrated into a parallel visualization system [11]. They have focused their study on the MPI-IO parallel technology. They have found out that special attention has to be paid to the I/O system by dedicating special resources and by properly tuning the I/O mechanisms, such as a dedicated input processor and adaptive fetching mechanisms. Another relevant work on the design of parallel algorithms has been carried out by Wapperom et al. [12]. They have designed and evaluated a parallel algorithm for computing the three-dimensional FFT. They have also found out that the I/O mechanisms play a central role on the performance of the overall parallel systems. They have based their system on MPI using two different platforms: an origin2000 and an alphaserver cluster having paid particular attention to the impact of the I/O methods over the overall performance of the parallel processing algorithm.

Various relevant research efforts have been reported lately for the wavelet transform. Thulasiraman et al. have developed a multithreaded algorithm for computing the 2D wavelet transform [13]. The authors conduct a set of experimental trails over an emulated multithreaded platform and came to the conclusion that their system outperforms the MPI-based implementations having been reported in the literature. Even though their results are very promising, the unavailability of a complete (hardware/software) fine coarse system leaves open the debate on the overall performance gains of such systems.

Regarding the 3D wavelet transform, Kutil and Uhl have conducted a study of the software and hardware needs of the 3D wavelet transform [14]. They have centered their study on the classic wavelet transform and conducted a set of trials on an SGI PowerChallenge. They have not, however, considered the use of the GOF concept in order to minimize the memory requirements when computing the 3D-WT. In a recent study they have focused on the time taken by the wavelet transform decomposition making part of the JPEG 2000 and MPEG-4 VTC image compression standards [15]. They have used a shared memory programming paradigm based on OpenMP. Once again, they only considered the classic 2D-WT.

In a recent paper [16], Katona et al. have studied the use of field-programmable gate arrays as a means to implement a real-time wavelet-domain video denoising algorithm. The proposed architecture makes use of two FPGAs. While the first one is dedicated to performing the wavelet decomposition, the second one reads the wavelet coefficients. The reported results show the effectiveness of the proposed scheme for real-time video processing. Even though the results are encouraging, the experimental platform is still under development as the noise level estimation requires the user intervention.

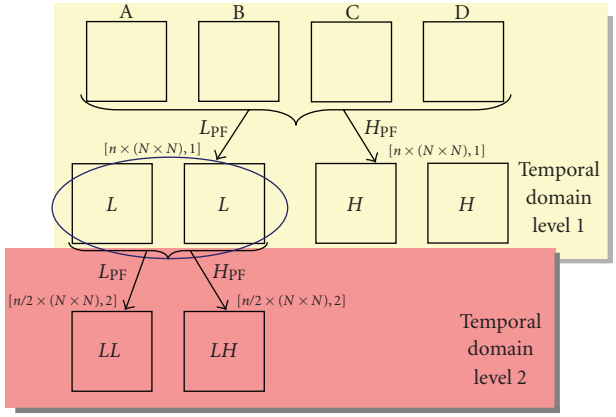

Figure 2: NoAl-classic algorithm: temporal decomposition.

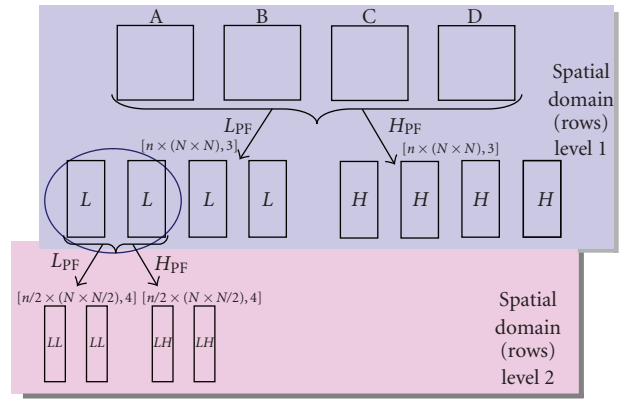

Figure 3: NoAl-classic algorithm: horizontal spatial decomposition (rows).

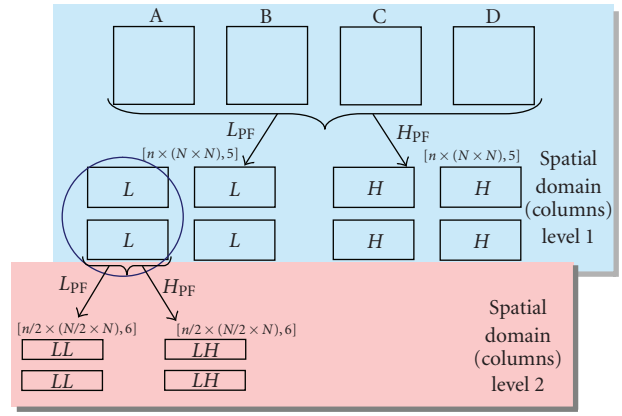

Figure 4: NoAl-classic algorithm: vertical spatial decomposition (columns).

Even though there have been many studies conducted on the parallel computation of various signal processing transforms, no in-depth studies have been reported on parallel algorithms to perform the spatial and temporal decompositions. Furthermore, they have not fully examined the use of workload strategies as a means to overcome the overhead of moving huge amounts of visual data. In this paper, we address these two issues, that is, alternative ways for computing the spatial and temporal decompositions and the use of workload strategies supplemented by state-of-the art parallel I/O mechanisms. Our algorithms aim at reducing the number of communication operations required to exchange data among the participating processors. We evaluate the overall performance of our proposals in an SGI origin 3800 making use of message passing paradigm based on MPI including the use of the MPI-IO library. The overall system performance is measured in terms of the response time and speed-up factor. We also evaluate the quality of the signal in terms of the first-order entropy.

4. NONALTERNATING WT ALGORITHMS

In this section, we describe the three algorithms whose parallel implementations and evaluation make the subject of this paper. All three algorithms are based on the nonalternating classic WT. In order to effectively reduce the huge memory requirements when applying the 1D-WT over the temporal domain, we propose the use of the GOF-based scheme introduced in Section 2.3.

According to the principles of operation of the nonalternating classic scheme, the 1D-WT is applied over a dimension as many times as having been predefined, before proceeding with the decomposition on the other dimension. That is to say, instead of alternatively transforming the temporal, horizontal (rows), and vertical (columns) domains, the algorithms presented herein completely apply the 1D-WT over a domain as many times as levels have been predefined. The following three figures show the details of the nonalternating classic algorithm.

Figure 2 explicitly depicts the case of a two-level temporal decomposition over a GOF consisting of four frames, named A, B, C, and D. As shown in the figure, the first decomposition is carried out over all the four frames resulting in two blocks containing the low-frequency bands labeled L and two blocks containing the high-frequency bands, labeled in turn H. Each one of these blocks is equivalent in size to an original frame. Figure 2 also shows that a second level of decomposition on the temporal domain is applied to the low-frequency bands which produces a block with the subband LL and a second one with the LH subband. In this figure, as well as in Figures 3 and 4, the notation $[n \times (N \times N), l]$ indicates the main features of the data being manipulated, that is to say, n indicates the number of frames per GOF. For instance, in the case of the decomposition carried out at level 2, this number will be halved in every dimension, indicating the number of frames in the GOF to be processed. $N \times N$ gives the size of the frames in pixels while the third parameter indicates the step number for the different levels of decomposition for the 3D-WT. For the sake of clarity, in the figures, we only represent two levels over each data dimension.

Figure 3 shows the two-level spatial (horizontal) decomposition. The output of this first level decomposition results

in four blocks containing the lowpass bands (two of them pertaining to the temporal low-frequency bands and the other two pertaining to the temporal high bands) and four other blocks containing the highpass bands. According to the classic algorithm, in the second level, only the low-pass band is processed for its further decomposition. As seen from the figure, the second level decomposition results in four blocks, containing the *LL*, *LL*, *LH,* and *LH* subbands. It is worth to notice that at each level of decomposition the amount of data to be processed is reduced. Furthermore, according to the classic algorithm, only the lowpass band is further processed at each level of decomposition.

In a similar way to the horizontal decomposition, the vertical decomposition is carried out over a GOF. As a result of the first level of decomposition, four blocks containing the *L* bands and four other blocks containing the *H* bands are obtained (see Figure 4). In the second level, and according to the classic algorithm, the decomposition is carried out only over the *L* bands, resulting in four blocks containing the *LL* and *LH* subbands.

The other two algorithms to be introduced aim to further eliminate time and space redundancies for a more efficient encoding of the resulting wavelet coefficients. The classic 3D-WT algorithm does not effectively reduce the spatial redundancies on every video frame. The reason is that it only applies all spatial decomposition levels over frames pertaining to the temporal low-frequency bands. As we will show, the removal of spatial redundancies from the temporal high-frequency bands implies a light increase on the computational complexity of the encoding process, but it will certainly result in a better coding efficiency.

The first of the algorithms, namely the NoAl-N0 algorithm, applies only one temporal decomposition level on a GOF. Furthermore, this algorithm only aims to reduce most spatial redundancies at every frame in a GOF, since it applies all the spatial decomposition levels over all the temporal subbands (*L* and *H*) and not only on those pertaining to the temporal low-frequency sub-bands; as done by the NoAl-classic (only *L* subbands in Figure 2) and classic algorithms.

The NoAl-N1 algorithm, similar to the NoAl-N0 algorithm, attempts to eliminate all the spatial redundancies, but it goes a step further by eliminating the temporal redundancies present in the image sequence. Then, it carries out the predetermined levels of temporal and spatial decompositions, but the spatial transform is applied over all the temporal sub-bands, as done by the NoAl-N0 algorithm.

The NoAl-classic, NoAl-N0, and NoAl-N1 algorithms require keeping in memory all the frames in a GOF to perform the temporal decomposition. In a sequential implementation, all the frames of a GOF have to be readily available to perform the temporal decomposition. This means that all frames in a GOF have to be in memory despite memory demands and cache misses. The latter can heavily impact the performance of the overall process. The problem is aggravated if a GOF consists of a large number of frames and large frame sizes. We then propose the use of parallel processing as an effective way to carry out the 3D-WT decomposition.

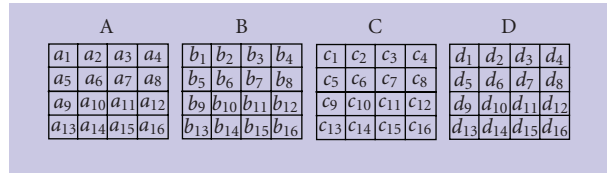

Figure 5: A GOF consisting of four frames.

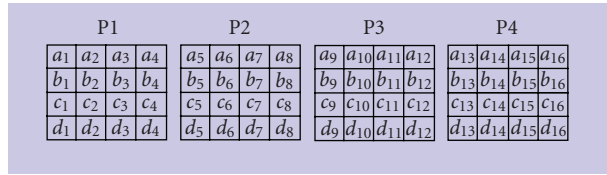

Figure 6: A distributed GOF (four frames) over four processors (P1, P2, P3, P4).

5. PARALLELIZATION SCHEME

The key points of the parallelization scheme proposed herein include a workload distribution strategy and the use of an efficient parallel I/O subsystem exploiting the hardware features.

5.1. Workload distribution strategy

The workload distribution strategy aims to evenly balance the workload among the processors and reduce the number of cache misses. The strategy has been designed by taking into account both the application and the multiprocessor system features. Towards this end, we have made use of the GOF-based scheme enabling the even distribution of the video sequence among the participating processors. Figure 5 depicts the case of four frames per GOF, where each frame has in turn been divided into four rows and each row into four blocks. For instance, frame A has been divided into four blocks per row, denoted from top to bottom by $A_1 = \{a_1, a_2, a_3, a_4\}$, $A_2 = \{a_5, a_6, a_7, a_8\}$, $A_3 = \{a_9, a_{10}, a_{11}, a_{12}\}$, and $A_4 = \{a_{13}, a_{14}, a_{15}, a_{16}\}$, where each row has been subdivided into four blocks. Frames B to D have been fragmented in the same way. Furthermore, the frames are divided into blocks of rows, X_i, with $i = 1, 2, \ldots, N$, where N denotes the number of processors. The ith processor receives the ith block of rows, within a GOF. Figure 6 illustrates the way the four frames depicted in Figure 5 are distributed among four processors, denoted by P1, P2, P3, and P4.

Under this workload distribution strategy, all frames in a GOF are distributed among N processors avoiding large memory demands or cache misses over a single processor, despite the GOF size. In this way, each active processor can perform the temporal decomposition, the most critical task, on a specific block of rows of all frames in a GOF. Furthermore, every processor can perform the spatial decomposition on its complete rows and its blocks of rows of each

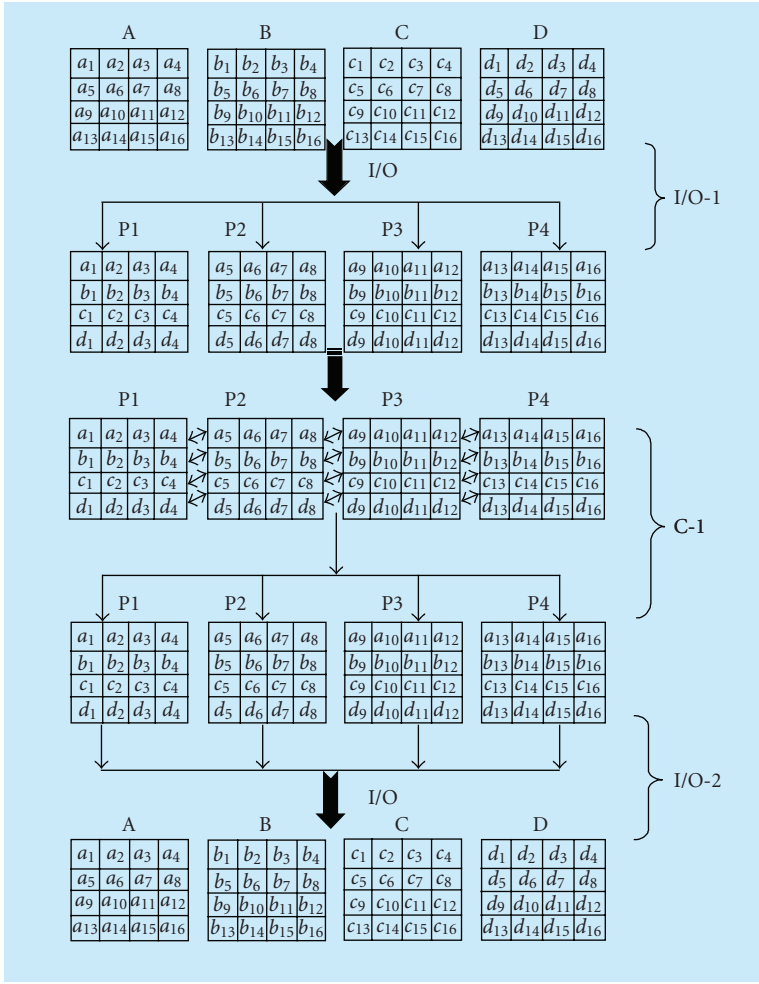

Figure 7: Parallel scheme: 4 frames per GOF and 4 processors (P: processor, C: communication, I/O: parallel input/output).

frame. Next, we undertake a more detailed description of our parallel algorithms and workload distribution implemented by our parallel scheme (see Figure 7).

5.2. Parallel scheme

Figure 7 shows the workload strategy used by the parallel 3D-WT decomposition algorithms proposed herein. First, the workload distribution is performed by a parallel I/O (denoted by step I/O-1 in Figure 7), taking advantage of the system hardware features. In this way, each processor reads the portion of data associated to the frames of the GOF stored in disk. This operation considerably reduces the data delivery versus a sequential I/O, where only one processor retrieves all data from the disk and sends the data to the corresponding processor.

Upon receiving the data to be processed, each processor carries out the temporal decorrelation on all the blocks of rows of the current GOF. Upon completing this phase, the processors can proceed to partially carry out the spatial decomposition of the rows. Both tasks are separately performed as many times as levels are predefined in the relevant algorithm. No communication among the processors is required since all the necessary data are available to the processors.

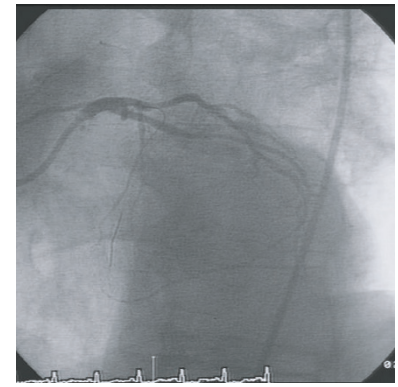

Figure 8: A typical frame from an angiography sequence.

The last stage of the nonalternating 3D-WT algorithms has to do with the spatial decomposition on a column-by-column basis. In order to be able to carry out this operation,

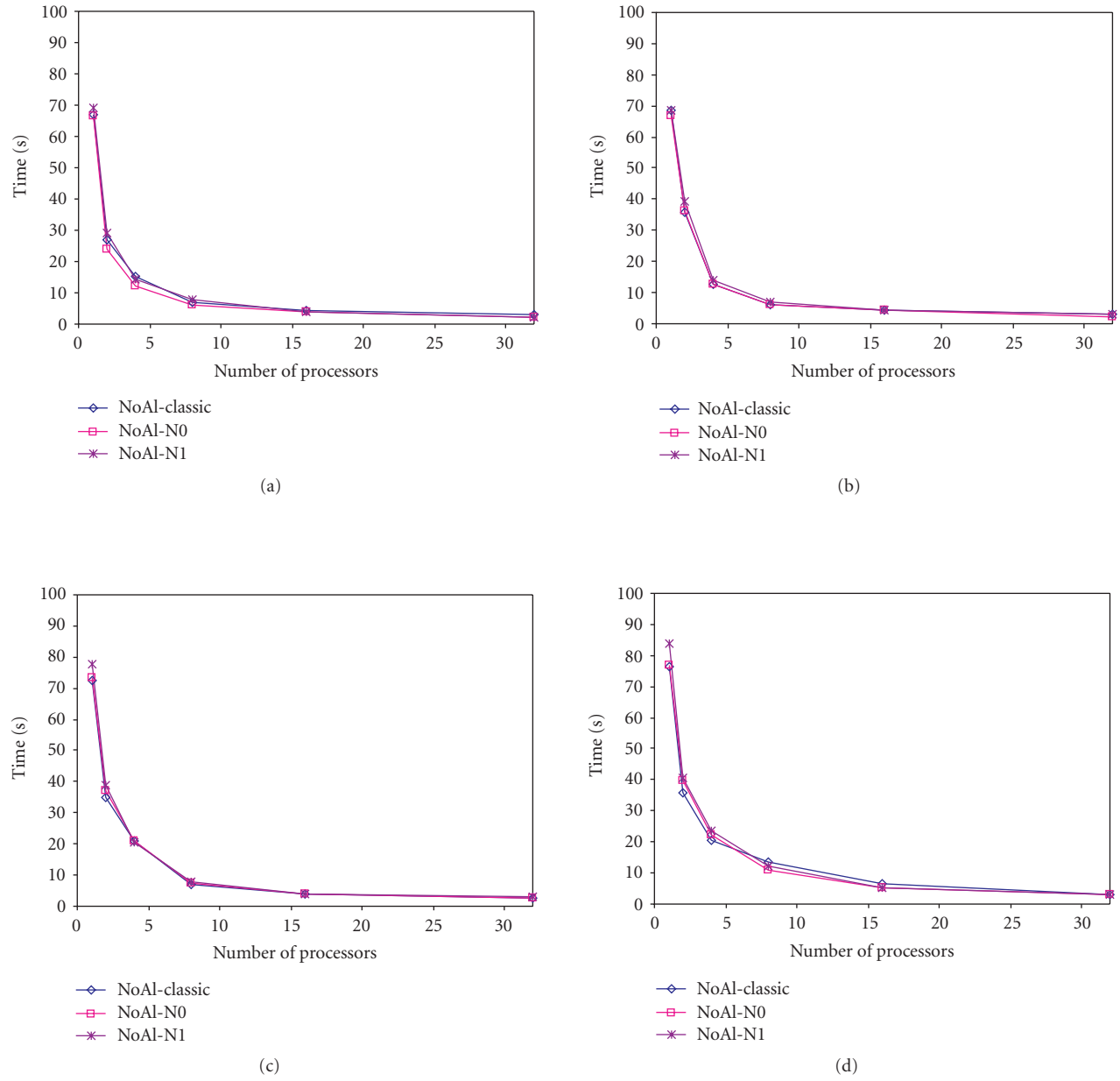

Figure 9: Processing time: frames of 1024 × 1024 pixels (a) 4 frames/GOF, (b) 8 frames/GOF, (c) 16 frames/GOF, (d) 32 frames/GOF.

every processor needs to get the edge data of the columns already available to them. This is required to properly perform the spatial decomposition avoiding any degradation on the quality of the frame. Sending the edge data required by each and every processor requires the exchange of data between a given processor and its neighbors, that is Processor Pi needs to exchange edge data with processors Pi−1 and Pi+1 (processors P1 and PN only exchange data with processors P2 and PN−1, resp.). This task is depicted as step C-1 in Figure 7. Upon receiving the data, each processor is able to perform the decomposition levels on its blocks of rows and edge data: the column transformation process.

After completing the described tasks, all the current GOF coefficients have been decorrelated and they are regrouped into disk by carrying a parallel I/O carried out jointly by all the participating processors. This is denoted by step I/O-2 in Figure 7.

It is important to note that transformed coefficients are not compressed. They would have to be properly encoded for this purpose. However, coding is out of the scope of this paper.

6. EXPERIMENTAL RESULTS

6.1. *Test data set*

We have performed a series of experiments using several raw angiography sequences consisting of 64 frames, and two

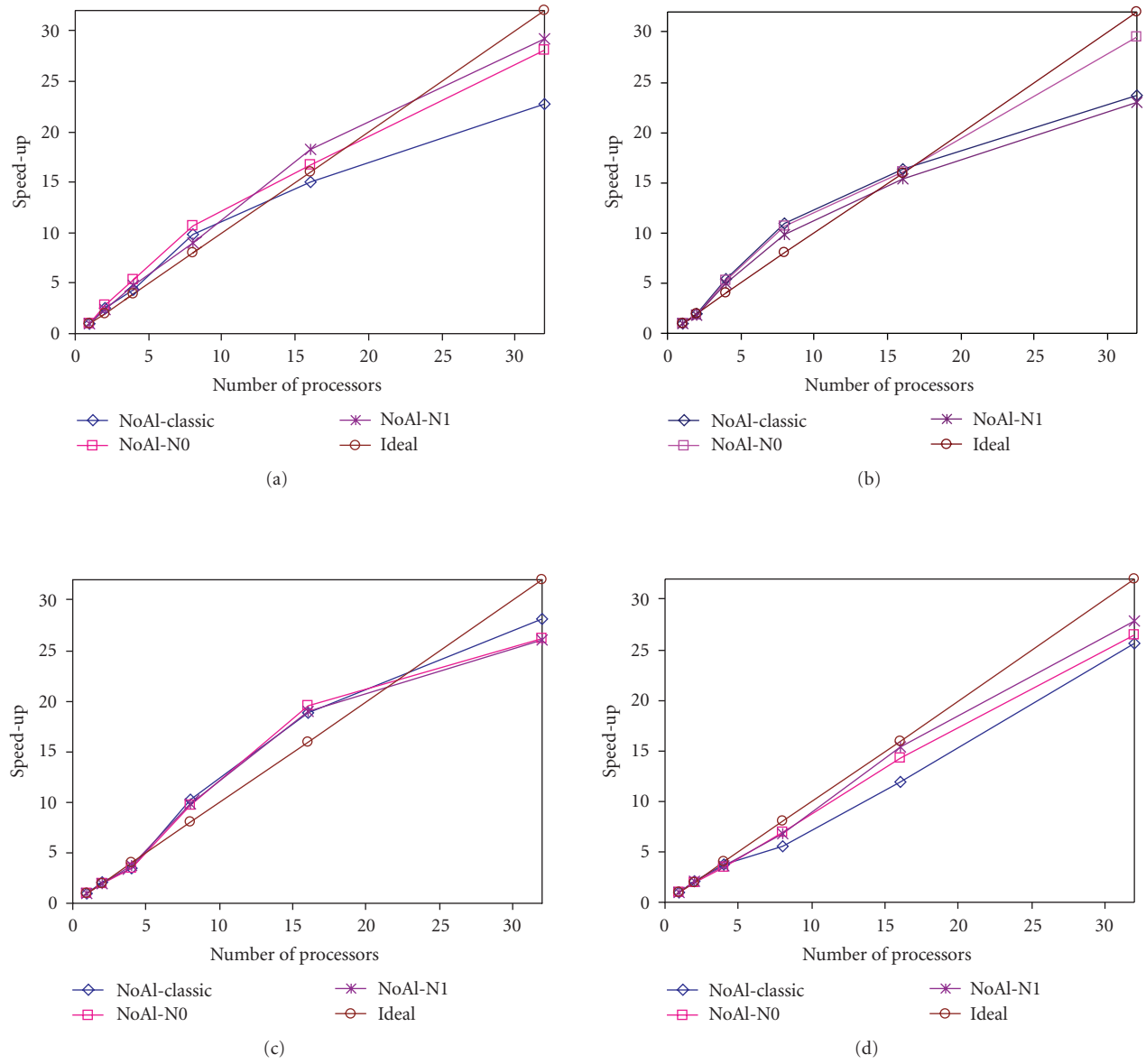

Figure 10: Speed-up: frames of 1024 × 1024 pixels (a) 4 frames/GOF, (b) 8 frames/GOF, (c) 16 frames/GOF, (d) 32 frames/GOF.

different frame sizes (512 × 512 pixels or 1024 × 1024 pixels per frame) in order to compare the system performance using two different workload sizes. Due to lack of space and due to the similar trends observed in the results for both frame sizes, we only include results for frames sizes of 1024 × 1024 pixels per frame. Figure 8 shows a typical image of an angiography sequence.

6.2. Computer system

All experiments have been carried out on a multiprocessor platform, the Origin 3000 Silicon Graphics Inc. family, SGI Origin 3800 [17] belonging to CIEMAT (Centro de Investigaciones Energéticas, Medio-ambientales y Tecnológicas, Spain). This multiprocessor platform consists of a multiprocessor characterized by a cc-NUMA (cache coherent-non uniform memory access) architecture with 128 MIPS R14000 processors, running at 600 MHz, with 1 GB of memory each, 1400 GB in RAID for storage, and using four fiber channel controllers.

The parallel implementations of the algorithms have been designed based on a message-passing paradigm, using the MPI library and its associated MPI-IO subsystem [6, 7]. All parallel jobs have been run in nondedicated mode, sharing with other users the system.

6.3. Metrics

We will evaluate the performance of the three algorithms in terms of three metrics, namely, the total processing time, the speed-up factor, and the first-order entropy.

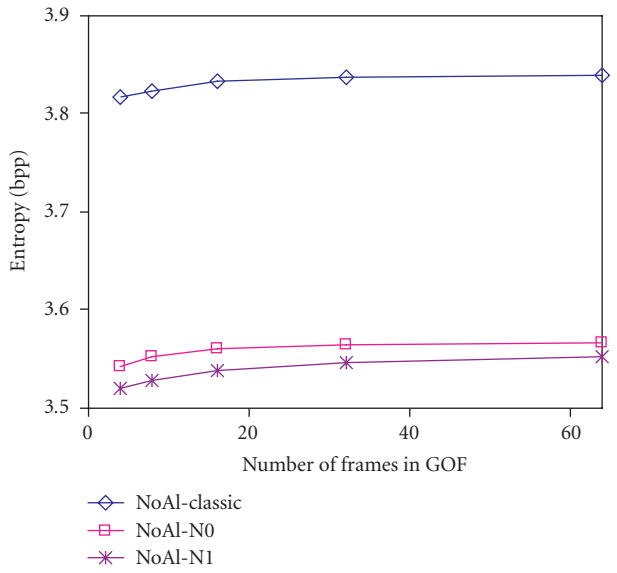

Figure 11: Entropy: 1024 × 1024 pixels per frame.

The total processing time required for decorrelating a complete angiography sequence includes the calculation of the 3D-WT, the I/O and communication times, and the initialization and ending of the MPI configuration.

Speed-up refers to how much a parallel algorithm is faster than a corresponding sequential algorithm. It is defined by the following expression:

$$S_p = \frac{T_s}{T_p}, \tag{1}$$

where T_p denotes the computational time per processor for a computation involving p processors (execution on p parallel processors) and T_s denotes the computational time involving a single processor system.

Sometimes the speed-up can be superlinear, that is, $S_p > p$. This can be due to the fact that a serial version of an algorithm may involve more overhead than a parallel version of the same algorithm, or that the hardware characteristics may favour the parallel version of the algorithm. This is particularly observed for applications requiring the processing of huge amount of data. For instance, if all data can be decomposed in caches or main memories of parallel processors, the number of cache misses will be considerably reduced.

The entropy, our third metric of interest, allows us to numerically assess the efficiency of the decomposition algorithms. The first-order entropy is defined as the average information content per symbol of the source [18]. It is measured in binary units for each of the transformed symbols or coefficients (expressed as bits for symbol or bits for pixel for the image case, simply denoted as *bpp*). A lower value of the entropy on a set of transformed coefficients means a lower redundancy in the set of symbols or elements. It is worth to note that the entropy metric is independent of any coding method. Since the calculation of the entropy can be very complex, we have chosen the first-order entropy as our metric to evaluate the coefficient as many authors. The first-order entropy, H, is simply expressed as follows:

$$H = -\sum_{i=1}^{m} p_i \log_2 p_i \text{ bits}. \tag{2}$$

6.4. Result analysis

In this section, we present our experimental results for the parallel implementations of the three wavelet transform methods under study, namely the NoAl-classic, NoAl-N0, and NoAl-N1 algorithms.

The experimental system has been tested considering six different configurations consisting of 1, 2, 4, 8, 16, and 32 processors.

Figure 9 shows the processing time for all the presented algorithms. We have taken into account different processor configurations in order to evaluate their impact on the runtime performance. The processing time includes the 3D-WT calculation (the wavelet transform part, up to 70%, is the most demanding part of the algorithm [15]), communication time, and I/O operations for a complete sequence decorrelation (64 frames). Figure 9 allows us to provide an absolute reference point of our experiments, in order to compare them with other experimental works.

Figure 10 presents the speed-up results corresponding to a frame size of 1024 × 1024 pixels. Four GOF sizes have been considered: 4, 8, 16, and 32 frames per GOF. A larger number of frames in a GOF allow us to reduce the inter-frame redundancy, especially when the sequence has not much motion.

As seen from Figure 10, all algorithms get good speed-up results using 8 processors or less, and for GOF sizes of up to 16 frames. The ideal speed-up is surpassed in many instances, overall up to 16 frames per GOF, and between 4 and 16 processors. In these cases, the workload distribution strategy has proven to be very effective reducing the overhead introduced by the memory paging system. In other words, a superlinear speed-up is obtained as a result of reducing the number of cache misses.

In Figure 10(d), we appreciate some interesting results. Since larger GOF sizes, that is, 32 frames/GOF, require more computing and I/O processing power, the speed-up factor is kept under the ideal one for all system configurations. However, we also notice that our proposed methodology greatly helps the NoAl-N0 and NoAl-N1 algorithms, whose computational requirements are higher than the requirements of the NoAl-classic algorithm. Our proposed approach is fully justified by this workload size for angiography sequences.

Regarding the effectiveness of the proposed 3D-WT algorithms, it is important to note that the NoAl-N0 and NoAl-N1 WT algorithms get better first-order entropy results than the NoAl-classic algorithm, for the test sequences under study. Figure 11 shows the first-order entropy results for all the algorithms.

7. CONCLUSIONS

In this paper, we have discussed and compared three algorithms for the nonalternating version of 3D-WT decorrelation.

Their parallel implementations have been developed and evaluated. Our results show that the proposed workload distribution is particularly suitable for a multiprocessor system based on a cc-NUMA (cache coherent-non uniform memory access) architecture.

The parallel versions of our algorithms reduce the overall processing time by effectively distributing the workload across several processors. In many instances, our algorithms improve the ideal speed-up for many of the cases being considered by avoiding the bottleneck of a single processor, cache misses, when a large GOF is used for temporal decomposition in the 3D-WT.

ACKNOWLEDGMENTS

This work has been partially supported by the Ministry of Science and Technology of Spain, under the CICYT Grant no. TIN2006-15516-C04-02 and the CONSOLIDER Grant no. CSD2006-46, and the Regional Science Council of Castilla-La Mancha.

REFERENCES

[1] M. Feil and A. Uhl, "Efficient wavelet-based video coding," in *Proceedings of the 16th International Parallel and Distributed Processing Symposium (IPDPS '02)*, p. 139, Fort Lauderdale, Fla, USA, April 2002.

[2] O. L. Nielsen and M. Hegland, "Parallel performance of fast wavelet transforms," *International Journal of High Speed Computing*, vol. 11, no. 1, pp. 55–74, 2000.

[3] E. Moyano-Ávila, F. J. Quiles, and L. Orozco-Barbosa, "Algorithms based on the standard wavelet transform for angiography sequences decomposition," in *World Congress on Medical Physics and Biomedical Engineering*, Springer, New York, NY, USA, 2006.

[4] A. Fournier, "Wavelets and their applications in computer graphics," in *Proceedings of the 22nd Annual Conference on Computer Graphics (SIGGRAPH '95)*, Los Angeles, Calif, USA, August 1995, course notes.

[5] W. A. Pearlman, B. J. Kim, and Z. Xiong, "Embedded video subband coding with 3D SPIHT," in *Wavelet Image and Video Compression*, pp. 397–432, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[6] P. Pacheco, *Parallel Programming with MPI*, Morgan Kaufmann Publishers, San Fransisco, Calif, USA, 1997.

[7] R. Thakur, "Introduction to parallel I/O and MPI-IO," in *Tutorial at 11th Annual Computing Institute*, San Diego Supercomputer Center, San Diego, Calif, USA, July 2005.

[8] S. G. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 12, pp. 2091–2110, 1989.

[9] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions of Image Processing*, vol. 1, no. 2, pp. 205–220, 1992.

[10] M. Modarressi and H. Sarbazi-Azad, "Parallel 3-dimensional DCT computation on k-ary n-cubes," in *Proceedings of the 8th International Conference on High Performance Computing in Asia Pacific Region (HPC-Asia '05)*, pp. 91–97, IEEE Computer Society, Beijing, China, November-December 2005.

[11] H. Yu and K.-L. Ma, "A study of I/O methods for parallel visualization of large-scale data," *Parallel Computing*, vol. 31, no. 2, pp. 167–183, 2005.

[12] P. Wapperom, A. N. Beris, and M. A. Straka, "A new transpose split method for three-dimensional FFTs: performance on an Origin2000 and Alphaserver cluster," *Parallel Computing*, vol. 32, no. 1, pp. 1–13, 2006.

[13] P. Thulasiraman, A. A. Khokhar, G. Heber, and G. R. Gao, "A fine-grain load-adaptive algorithm of the 2D discrete wavelet transform for multithreaded architectures," *Journal of Parallel and Distributed Computing*, vol. 64, no. 1, pp. 68–78, 2004.

[14] R. Kutil and A. Uhl, "Hardware and software aspects for 3-D wavelet decomposition on shared memory MIMD computers," in *Proceedings of the 4th International ACPC Conference Including Special Tracks on Parallel Numerics and Parallel Computing in Image Processing, Video Processing, and Multimedia (ACPC '99)*, vol. 1557 of *Lecture Notes in Computer Science*, pp. 347–356, Salzburg, Austria, February 1999.

[15] R. Norcen and A. Uhl, "High performance JPEG 2000 and MPEG-4 VTC on SMPs using OpenMP," *Parallel Computing*, vol. 31, no. 10–12, pp. 1082–1098, 2005.

[16] M. Katona, A. Pižurica, N. Teslić, V. Kovačević, and W. Philips, "A real-time wavelet-domain video denoising implementation in FPGA," *EURASIP Journal of Embedded Systems*, vol. 2006, Article ID 16035, 12 pages, 2006.

[17] Silicon Graphics Inc., "SGI Origin 3000," http://www.sgi.com/origin/3000/.

[18] R. C. González and R. E. Woods, *Digital Image Processing*, Prentice-Hall, Upper Saddle River, NJ, USA, 2nd edition, 2002.

E. Moyano-Ávila received the B.S. degree in 1993 in computer science, from the University of Granada, Spain, and the M.S. degree and the Ph.D. degree, both in computer science, from the University of Castilla-La Mancha, Spain, in 1999 and 2006, respectively. She is an assistant professor of the Department of Information Technologies and Systems at the same university. Her present research interests include parallel algorithms for multiprocessors and wavelet transform for biomedical video compression.

F. J. Quiles received the M.S. degree in physics (electronics and computer science) and the Ph.D. degree from the University of Valencia, Spain, in 1986 and 1993, respectively. In 1986, he joined the Computer Systems Department at University of Castilla-La Mancha, where he is currently a Full Professor of computer architecture and technology and Vice Rector of Research of the University of Castilla-La Mancha. He has developed several courses on computer organization and computer architecture. His research interests include high-performance networks, parallel algorithms for video compression, and video transmission. He has published over 120 papers in international journals and conferences.

L. Orozco-Barbosa received the B.S. degree in electrical and computer engineering from Universidad Autonoma Metropolitana, Mexico, in 1979, the Diplome d'Etudes Approfondies from École Nationale Supérieure d'Informatique et de Mathématiques Appliquées de Grenoble (ENSIMAG), France, in 1984 and the Doctorat de l'Université from Université Pierre et Marie Curie,France, in 1987, both in computer science. From 1991 to 2002, he was a faculty member at the School of Information Technology and Engineering (SITE), University of Ottawa, Canada. In 2002, he joined the Department of Informatics at Universidad de Castilla-La Mancha (Spain). He has also been appointed Director of the Albacete Research Institute of Informatics, a Regional Centre of Excellence. He has conducted numerous research projects with the private sector and served as Technical Advisor for the Canadian International Development Agency (CIDA). His current research interests include Internet protocols, image/video communications, wireless communications, traffic modeling, and performance evaluation. He is a Member of the IEEE.