# Three-Dimensional Encoding/Two-Dimensional Decoding of Medical Data

Gloria Menegaz*, *Member, IEEE*, and Jean-Philippe Thiran, *Member, IEEE*

***Abstract*—We propose a fully three-dimensional (3-D) wavelet-based coding system featuring 3-D encoding/two-dimensional (2-D) decoding functionalities. A fully 3-D transform is combined with context adaptive arithmetic coding; 2-D decoding is enabled by encoding every 2-D subband image independently. The system allows a finely graded up to lossless quality scalability on any 2-D image of the dataset. Fast access to 2-D images is obtained by decoding only the corresponding information thus avoiding the reconstruction of the entire volume. The performance has been evaluated on a set of volumetric data and compared to that provided by other 3-D as well as 2-D coding systems. Results show a substantial improvement in coding efficiency (up to 33%) on volumes featuring good correlation properties along the $z$ axis. Even though we did not address the complexity issue, we expect a decoding time of the order of one second/image after optimization. In summary, the proposed 3-D/2-D multidimensional layered zero coding system provides the improvement in compression efficiency attainable with 3-D systems without sacrificing the effectiveness in accessing the single images characteristic of 2-D ones.**

***Index Terms*—3-D/2-D, compression, lossless, volumetric data, wavelets.**

## I. Introduction

MOST of the current medical imaging techniques produce three-dimensional (3-D) data distributions. Some of them are intrinsically volumetric, like magnetic resonance (MR), computerized tomography (CT), positron emission tomography (PET), and 3-D ultrasound, while others describe the temporal evolution of a dynamic phenomenon as a sequence of two-dimensional (2-D) images, so that they are more properly labeled as 2-D+time. The huge amount of data generated every day in the clinical environment has triggered considerable research in the field of volumetric data compression for their efficient storage and transmission. The basic idea is to take advantage of the correlation among the data samples in the 3-D space to improve compression efficiency. The most widespread approach combines a 3-D decorrelating transform with the extension of a coding algorithm that has proved to be effective on 2-D images. In [1], the 3-D version of the set partitioning in hierarchical trees (SPIHT) [2] algorithm for image compression is applied to volumetric medical images. The same guideline is followed in [3], where the authors also address the problem of context modeling for efficient entropy coding. The performance of the 3-D extension of the embedded zerotree wavelet (EZW)-based coding algorithm [4] is analyzed in [5]–[7]. A slightly different approach is described in [8], where a 3-D-DCT is followed by quantization, adaptive bit allocation and Huffman encoding. In [9]–[11], a 3-D separable wavelet transform is used to remove interslice redundancy, while in [12] different sets of wavelet filters are used in in the $(x, y)$ plane and $z$ direction, respectively, to account for the difference between the intraslice and and interslice resolution.

This led to the common consensus that the exploitation of the full 3-D data correlation potentially improves compression. The main drawback of 3-D systems is computational complexity. If an increase in the encoding time might be tolerated, a swift decoding is of prime importance for the efficient access to the data. A possible solution has been proposed in [1] and [5]. It consists in splitting the volume in coding units of 8 or 16 images each and processing those independently in order to save memory and reduce the coding time. Coding units are fixed *a priori*, as well as the number of images which are decoded at one time.

Our solution that is based on the observation that it is common practice to analyze 3-D data distributions one image at a time for medical examination. Accordingly, in order to be suitable within a picture archiving and communication system (PACS) a coding system must provide a fast access to the single 2-D images. In the proposed solution, the decoding time is kept low by minimizing the amount of information to be decoded to reconstruct any 2-D image (or, more in general, subset of images) of the dataset. This is accomplished by independently encoding each subband image, and making the corresponding information accessible through the introduction of some special characters (i.e., markers) into the bitstream. Once the user has specified the position of the image of interest along the $z$ axis, the set of subband images that are needed for its reconstruction is determined and the related information is decoded. The inverse discrete wavelet transform (IDWT) is performed locally and the single image is recovered. The coding scheme is based on the multirate 3-D subband coding of video described in [13]. What we retain is the strategy used for entropy coding, namely the multidimensional context-adaptive arithmetic coding [14]. The subtended subband structure is nevertheless different. We perform a 3-D-DWT on the volume instead of treating differently the spatial and temporal dimensions.

The paper is organized as follows. Section II gives an overview on the global system. In Section III, the lifting scheme and integer wavelet transform are revisited. Section IV describes the procedure followed to determine the set of subband images needed to reconstruct a given image of interest.

Manuscript received December 10, 2000; revised August 30, 2002. *Asterisk indicates corresponding author.*

*G. Menegaz is with the Department of Computer Science, University of Fribourg, CH-1700 Fribourg, Switzerland (e-mail: gloria@ieee.org).

J.- P. Thiran is with the Signal Processing Institute, School of Engineering Techniques and Sciences, Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland.

Digital Object Identifier 10.1109/TMI.2003.809689

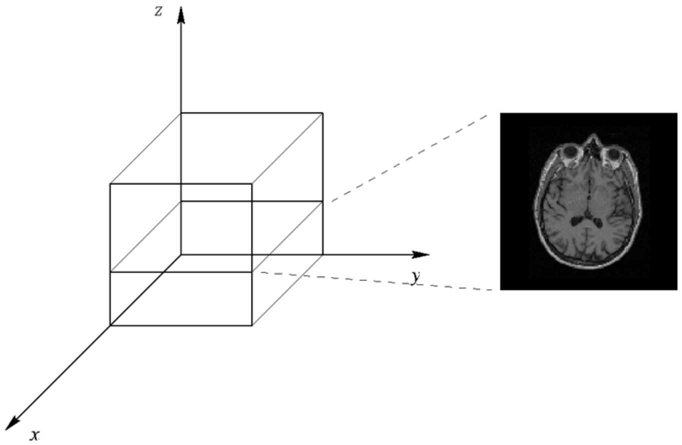


Fig. 1. Volumetric data. We call $z$ the third dimension, and assume that the images are the intersections of the volume with a plan orthogonal to $z$ axis.

The coding principle is presented in Section V and Section VI illustrates the different working modalities. The compression performance is analyzed in Section VII, and Section VIII derives conclusions.

## II. The 3-D/2-D Multidimensional Layered Zero Coding (MLZC) System

The combination of the 3-D wavelet transform with an *ad-hoc* coding strategy provides high coding efficiency and fast access to any 2-D image of the dataset. Given the index of the image of interest along the $z$ axis ($z$ coordinate in Fig. 1), the corresponding portion of the bitstream is accessed and decoded to recover it at the desired quality. At the encoder, the data are first decorrelated by a 3-D DWT and then encoded via the MLZC technique. At the decoder, the set of wavelet coefficients necessary to reconstruct an image of index $\overline{z}$ is automatically determined and only the corresponding parts of the bitstream are decoded. The IDWT is performed locally, reducing the memory requirements and the computational cost.

The wavelet transform has many features that make it suitable for our application. The approximation properties of reasonably smooth signals have determined the success of wavelet-based techniques for image compression. Noteworthy, the JPEG2000 standard [15] follows the same approach. The implementation via the lifting steps scheme [16] is particularly advantageous in this framework. First, it provides a very simple way of constructing nonlinear wavelet transforms mapping integer-to-integer values [17]. This is very important for medical applications because it enables lossless coding. Second, perfect reconstruction is guaranteed by construction for any kind of signal extension along borders. This greatly simplifies the management of the boundary conditions and facilitates the selection of the coefficients needed to reconstruct an image. Third, it is computationally efficient. It can be shown that the lifting steps implementation asymptotically reduces the computational complexity by a factor 4 with respect to the classical filter-bank implementation [18]. Finally, the transformation can be implemented in-place, namely progressively updating the values of the original samples, without allocating auxiliary memory.

The 3-D-DWT is followed by successive approximation quantization and context adaptive arithmetic coding. Some markers are placed in the codestream for the random access to the encoded information. By combining the 3-D-DWT with 2-D spatial neighborhoods for entropy coding, the resulting MLZC algorithm features 3-D encoding/2-D decoding capabilities.

However, many degrees of freedom are left for the design of the system. The shape of the spatial support of the neighborhood defining the context and the placement rule of the markers in the bitstream lead to different working modes. The global-progressive (G-PROG) mode is obtained by encoding the volume as a whole and without putting any marker. This mode provides the best compression efficiency. Both 2-D and 3-D contexts can be used. The resulting bitstream is fully embedded, supporting a finely-graded range of bit-rates ensuring scalable quality on the volume, but 2-D decoding is not possible. The layer-per-layer (LPL), and LPL progressive (LPL-PROG) modes are obtained by adding some markers in order to enable random access to the information of interest in the bitstream. More specifically, the LPL mode provides random access to every subband image. The idea is to decode the entire information concerning the set of subband images needed to reconstruct the image of interest at full quality (i.e., lossless). To achieve quality scalability on the final 2-D image, other markers must be added, leading to the LPL-PROG mode. Direct access is possible to every quantization layer of every subband image. Scalable quality is obtained by successively decoding the quantization layers, i.e., the bitplanes, of the concerned subband images. The drawback is the bitstream over-heading due to the additional information needed for data addressing, which reduces compression efficiency. Fig. 2 summarizes the three working modalities and illustrates the position of the markers in the bitstream. In the figure, H is for the Header of the bitstream, and $L_i^j$ represents the quantization layer $i$ of the subband image at position $j$ in a given 3-D subband. In the G-PROG mode, the whole information concerning the quantization layer $i$ is encoded for all the subband images. In the LPL mode, all quantization layers are located in the same segment and markers are placed only between $L_n^j$ and $L_1^{j+1}$, $n$ being the number of quantization steps. This reduces the number of markers while preserving 2-D decoding capabilities, improving the compression efficiency at the expense of the SNR scalability in lossy regime. Indeed, such a mode is intended for recovering the 2-D image of interest at full quality. Finally, in the LPL-PROG mode, the order is the same but markers are put between $L_i^j$ and $L_i^{j+1}$, $\forall i, j$.

## III. Integer Wavelet Transform via Lifting

The spatial correlation among data samples is exploited by a fully 3-D separable wavelet transform. The signal is successively filtered and down-sampled in all spatial dimensions. The decomposition is iterated on the approximation low-pass band, which contains most of the energy [19]. Fig. 3 shows the classical filter-bank implementation of the DWT. The forward transform uses two analysis filters, $\tilde{h}(z)$ (low-pass) and $\tilde{g}(z)$ (bandpass), followed by subsampling, while the inverse transform first up-samples and then applies two synthesis filters,

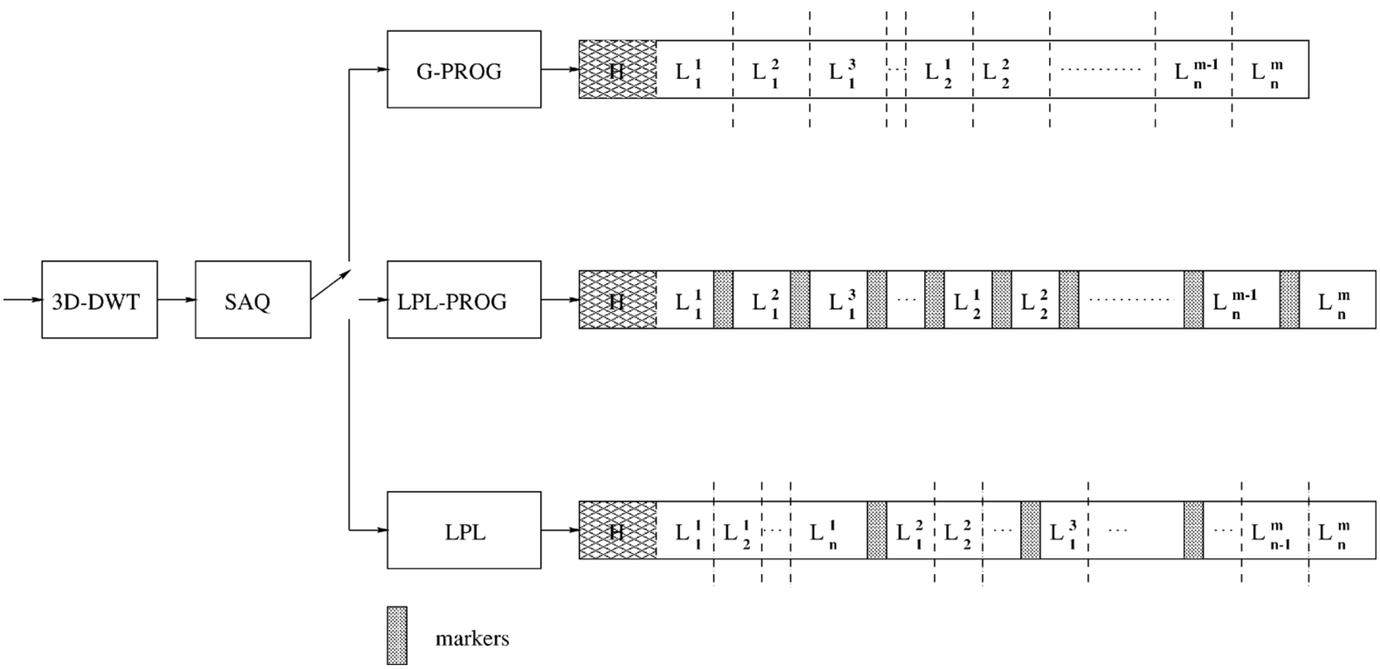

3D-DWT
SAQ
G-PROG
LPL-PROG
LPL
H
$L_1^1$ $L_1^2$ $L_1^3$ $L_2^1$ $L_2^2$ $L_n^{m-1}$ $L_n^m$
$L_1^1$ $L_2^1$ $L_n^1$ $L_1^2$ $L_2^2$ $L_1^3$ $L_{n-1}^m$ $L_n^m$
markers

Fig. 2. MLZC working modalities. H: bitstream header; $L_i^j$ quantization layer $i$ of the subband image at position $j$ in a given 3-D subband. In the G-PROG mode, the whole information concerning the quantization layer $i$ is encoded for all the subband images. In LPL mode, all the quantization layers are located in the same segment and markers are only between $L_n^j$ and $L_1^{j+1}$, $n$ being the number of quantization steps. In the LPL-PROG mode, the order is the same but markers are put between $L_i^j$ and $L_i^{j+1}$, $\forall i, j$.

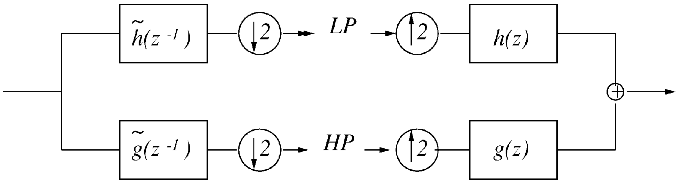


Fig. 3. DWT.

$h(z)$ (low-pass) and $g(z)$ (bandpass). Fig. 4 shows a two levels DWT on a natural image. The approximation subband is a coarser version of the original, while the other subbands represent the high frequencies (details) in the horizontal, vertical and diagonal direction, respectively.

In the proposed system, the DWT is implemented according to the recently developed lifting steps scheme [16]. The lifting scheme provides a way to perform any DWT with finite filters with a finite number of *lifting steps*. The lifting steps representation of a given filter is obtained by the Euclidean factorization of the polyphase matrix (see Fig. 5) of the filter bank into a sequence of $2 \times 2$ upper and lower triangular matrices. The polyphase matrix $P(z)$ is defined as

$$P(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} \tag{1}$$

where

$$\begin{aligned} h_e(z^2) &= \frac{h(z)+h(-z)}{2} & g_e(z^2) &= \frac{g(z)+g(-z)}{2} \\ h_o(z^2) &= \frac{h(z)-h(-z)}{2z^{-1}} & g_o(z^2) &= \frac{g(z)-g(-z)}{2z^{-1}} \end{aligned} \tag{2}$$

$h_e(z)$ and $h_o(z)$, respectively, $g_e(z)$ and $g_o(z)$, are the even and odd polyphase components of the synthesis filter $h(z)$, respectively, $g(z)$. If the determinant of $P(z)$ is equal to one, then the filter pair $(h, g)$ is *complementary*. In this case, the following theorem holds [16]:

*Theorem 1:* Given a complementary filter pair $(h, g)$, then there always exist Laurent polynomials $s_i(z)$ and $t_i(z)$ for $1 \le i \le m$ and a nonzero constant $K$ so that

$$P(z) = \prod_{i=1}^{m} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & \frac{1}{K} \end{bmatrix}. \tag{3}$$

The block diagrams for the forward and inverse transforms are illustrated in Figs. 6 and 7, respectively. Each triangular matrix corresponds to one lifting step. The number of lifting steps $m$ depends on both the length of the filters and the factorization. It is worth noticing that the result of the Euclidean factorization is not unique, so many lifting representations are possible for the same $P(z)$. From Figs. 6 and 7 it is easy to realize that the synthesis chain can be obtained by mirroring the filter-bank from the analysis counterpart and changing the sign of the filters. The global system can be seen as a sequence of *do/undo* steps, for which the perfect reconstruction property is ensured by construction. This provides additional degrees of freedom in the design of the filters, allowing any nonlinear operations into the basic blocks and any kind of signal extension outside the borders. In particular, the integer DWT is obtained by introducing a rounding operation after each lifting step [17]. As mentioned in Section II, the availability of an integer version of the transform enables lossless coding and makes the algorithm suitable for the implementation on a device. However, the integer coefficients are approximations of those that would be obtained by projecting the signal on the original wavelet basis. This can be modeled by an equivalent noise which becomes noticeable when the hypothesis of high-resolution quantization holds. It can be shown that it introduces an additional contribution to the quanti-

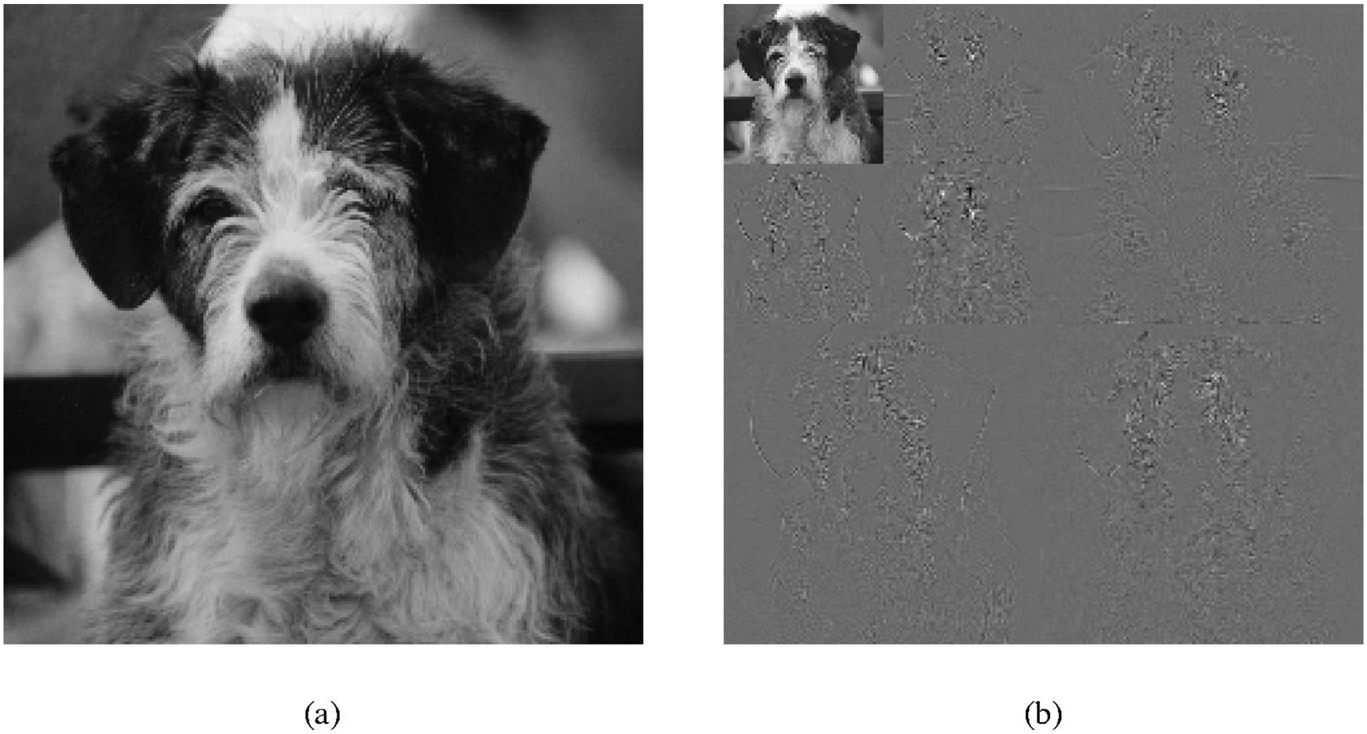


Fig. 4. DWT of a natural image. (a) Original; (b) DWT subbands for a two level decomposition. The approximation subband is a coarser version of the original, while the other subbands represent the high frequencies (details) in the horizontal, vertical and diagonal direction.

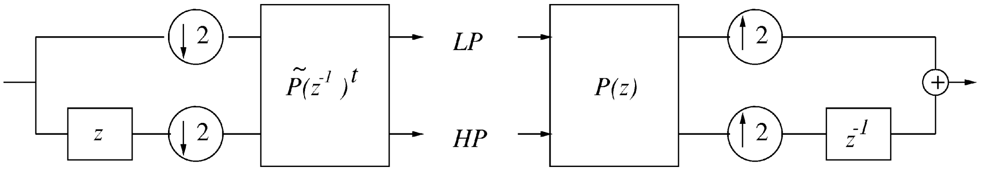


Fig. 5. Polyphase representation of the wavelet transform.

zation noise, which degrades the rate/distortion performance of the coding system [20]. Furthermore, it is responsible for an oscillatory trend of the PSNR along the $z$ axis, making the quality of the reconstructed image dependent on its position within the volume. The analysis of such a phenomenon is out of the scope of this paper. We refer to [21] for more details. What it is important to mention here is that the amount of such noise is proportional to the number of rounding operations, which in turn depends on the decomposition depth and the lifting chain length. Accordingly, we have restricted the choice of filters to the family of the *interpolating filters* [22] admitting a two-steps chain

$$P(z) = \begin{bmatrix} 1 & g_e(z) \\ h_o(z) & 1 + h_o(z)g_e(z) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ h_o(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & g_e(z) \\ 0 & 1 \end{bmatrix}. \tag{4}$$

As the choice of the filter-bank is not critical for compression performances, we choose the 5 × 3 [22] filter. Being extremely short – two steps of length two each – it minimizes the number of subband images to decode for recovering the 2-D image of interest, as will be discussed in Section IV.

The particular structure of the lifting chain facilitates the determination of the set of subband images needed for the point-wise IDWT (PW-IDWT). The separability of the transform allows to map such a task to the one-dimensional (1-D) case. The core of the problem consists in finding the set of subband coefficients needed to recover one signal sample. Then, results can be easily extended to intervals (i.e., signal segments) and, eventually, multiple dimensions.

## IV. Point-Wise IDWT

In this section, we formalize the PW-IDWT. It is basically a 1-D problem: each pixel of the image to recover is regarded as the sample $s(k)$ in position $k$ of the 1-D signal observed along the parallel to the $z$ axis passing trough it. Correspondingly, the set of subband coefficients that are needed for its reconstruction by IDWT maps to the coordinates of the subband images along the $z$ axis.

The proposed solution exploits the inherent recursive nature of the wavelet transform. The IDWT is an iterative process starting at the coarsest scale: the approximation subband at the finer $(l-1)$ level is reconstructed by filtering the set of coefficients at the coarser $l$ level according to [19]

$$a(l-1,k) = \sum_{n=-\infty}^{\infty} h(k-2n)a(l,n) + \sum_{n=-\infty}^{\infty} g(k-2n)d(l,n) \tag{5}$$

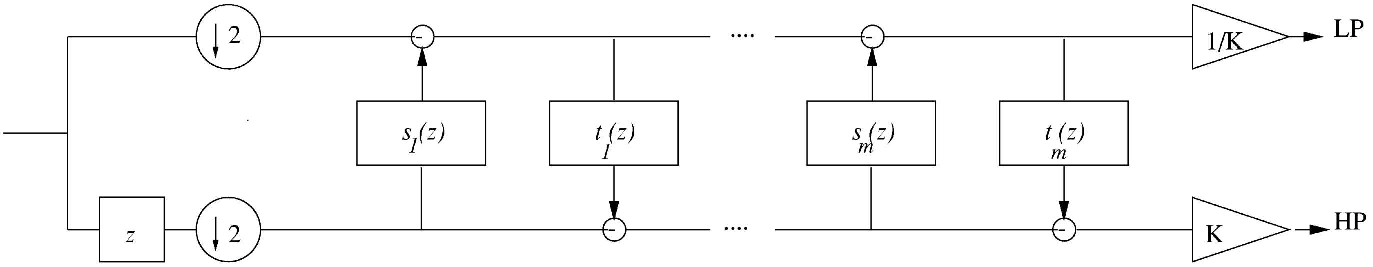


Fig. 6. Forward wavelet transform using lifting.

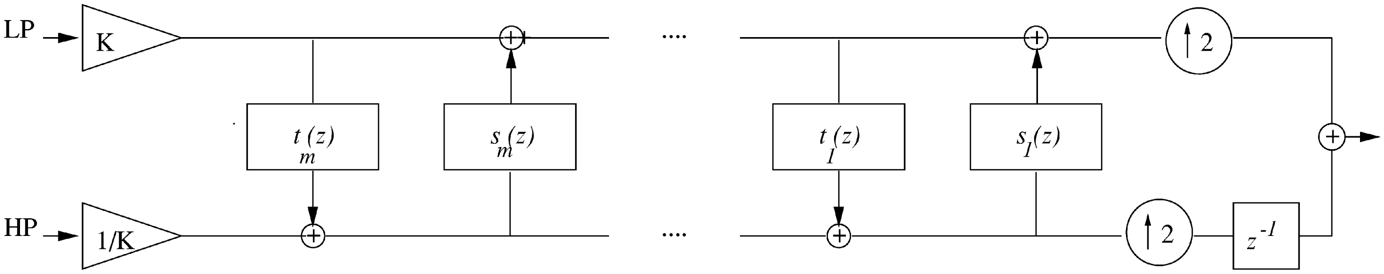


Fig. 7. Inverse wavelet transform using lifting.

where $a(\cdot)$ and $d(\cdot)$ are the approximation and detail subbands, respectively, and $l = 1, \ldots, L$ is the decomposition level, which increases with the depth of the decomposition. The signal $s(k)$ is reconstructed by iterating such a procedure for $L \geq l > 1$. The number of coefficients taking part to the convolution in a given subband depends on the length of the filter and on the number of decomposition levels. The method used to determine the positions of the involved coefficients in each subband consists in climbing back the synthesis filter-bank and keeping track of the positions of the subband coefficients that get involved step by step. Given the position $k$ of the sample of interest in the signal domain, we start by identifying the set of coefficients $\mathrm{GP}(1, j)$ that are needed at the finest resolution (i.e., $l = 1$). Here, $j$ is the subband index and takes the values $a$ for approximation and $d$ for details, respectively. For doing this, we look into the synthesis chain from its output, and follow it step by step, keeping track of the samples needed by the lifting steps filters. Due to the recursiveness of the IDWT, given $\mathrm{GP}(1, j)$ the procedure is iterated to get $\mathrm{GP}(2, j)$, at the next finer resolution (i.e., $l = 2$). The only difference is that now there is a *set* of samples to be recovered $\{\mathrm{GP}(1, a)\}$ instead of a single one. The iteration of such a procedure for $l = 1, \ldots, L-1$ results in the complete set of necessary subband coefficients.

The procedure can be easily generalized to sets $\{s(k), k \in K\}$ of samples in the signal space. Let $\mathrm{GP}(l, j)_k$ identify the coefficients in subband $j$ at level $l$ needed to reconstruct the signal sample in position $k$. Then, the solution for the set of samples $\{s(k)\}$ is

$$\mathrm{GP}(l, j) = \cup_k \mathrm{GP}(l, j)_k \quad \forall k \in K. \tag{6}$$

Formula (6) also applies to subband intervals. It is worth mentioning here that $\mathrm{GP}(l, j)_k$ depends on $k$ being even or odd. In general, with the usual structure of the lifting scheme starting with an $s$-type step, odd indexed samples correspond to larger $\mathrm{GP}(l, j)_k$. We refer to the Appendix A for the details.

TABLE I
NUMBER OF SAMPLES $\mathrm{GP}(l, j)_k$ IN SUBBAND $(l, j)$ NEEDED TO RECOVER THE SAMPLE AT POSITION $k$ USING THE $5 \times 3$ FILTER AND FOR $L = 3$ LEVELS OF DECOMPOSITION.

| $l$ | $j$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ | $k=6$ | $k=7$ |
|---|---|---|---|---|---|---|---|---|---|
| 3 | $a$ | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | $d$ | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 2 | $a$ | 1 | 2 | 2 | 2 | 1 | 2 | 1 | 2 |
| | $d$ | 2 | 3 | 3 | 3 | 2 | 3 | 2 | 3 |
| 1 | $a$ | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| | $d$ | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 |
| Total | | 9 | 15 | 13 | 15 | 11 | 15 | 11 | 15 |

TABLE II
NUMBER OF SAMPLES $\mathrm{GP}(l, j)_k$ IN SUBBAND $(l, j)$ NEEDED TO RECOVER THE SAMPLE AT POSITION $k$ USING THE 9/7 FILTER AND FOR $L = 3$ LEVELS OF DECOMPOSITION.

| $l$ | $j$ | $k=0$ | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ | $k=6$ | $k=7$ |
|---|---|---|---|---|---|---|---|---|---|
| 3 | $a$ | 5 | 5 | 5 | 6 | 6 | 6 | 5 | 5 |
| | $d$ | 6 | 6 | 6 | 7 | 7 | 7 | 6 | 6 |
| 2 | $a$ | 5 | 5 | 4 | 5 | 5 | 5 | 4 | 5 |
| | $d$ | 6 | 6 | 5 | 6 | 6 | 6 | 5 | 6 |
| 1 | $a$ | 3 | 4 | 3 | 4 | 3 | 4 | 3 | 4 |
| | $d$ | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 |
| Total | | 29 | 31 | 27 | 33 | 31 | 33 | 27 | 31 |

Tables I and II give $\mathrm{GP}(l, j)_k$ as a function of the sample position $k$ for $l = 1, \ldots, L, L = 3$. The number of samples required in each subband turns out to be a periodic function of $k$ with period $T = 2^L$. To outline the dependency of $\mathrm{GP}(l, j)_k$ from $k$, results are provided for $2^L = 8$ successive values of $k$. As the filter used is very short, the number of wavelet coefficients involved in the PW-IDWT is very small. For comparison, Table II

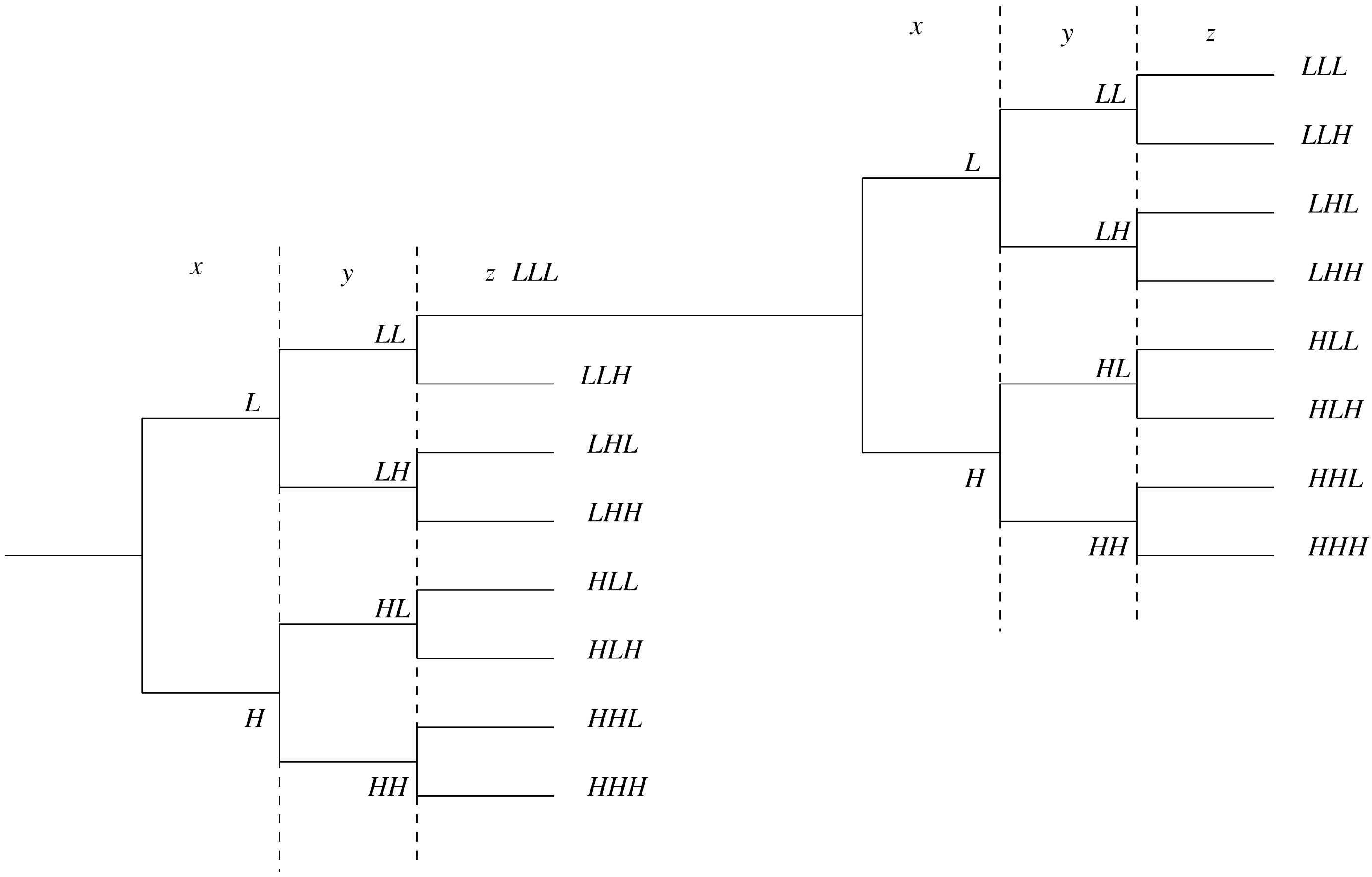


Fig. 8. Set of 3-D subbands resulting from $L = 2$ levels of decomposition.

shows the size of $\mathrm{GP}(l,j)_k$ for the 9/7 filter [23]. The number of coefficients is more than doubled. This means that in the considered application the amount of information to decode increases of more than the 50%, compromising the efficiency of the decoder. In the 3-D system, filtering is successively performed on the $x$, $y$, and $z$ directions. We assume that the 2-D images are stacked along the $z$ axis. Then, the positions of the wavelet coefficients that are needed in each subband map to the positions of the subband images–along the $z$ axis–within the corresponding subband. Accordingly, $\mathrm{GP}(l,j)_k$ identifies the $z$-coordinates of all the images in subband $(l,j)$ that are necessary to recover the image of interest. In this case, the index $j$ selects either low-pass $(j = a)$ or high-pass $(j = d)$ filtering along $z$. The total number $N_k$ of subband images needed for the reconstruction of image $k$ is given by

$$N_k = 4 \times [\mathrm{GP}(L,a)_k + \mathrm{GP}(L,d)_k] + \sum_{l=L-1}^{1} [3 \times \mathrm{GP}(l,a)_k + 4 \times \mathrm{GP}(l,d)_k]. \quad (7)$$

The intuition for this is given in Fig. 8. The number of 3-D subbands obtained by low-pass filtering along $z$ is equal to four for $l = L$ and is equal to three for $l < L$. Conversely, the number of $z$-high-pass subbands is equal to four at any level. Table III shows $N_k$ for the $5 \times 3$ and the 9/7 filters. Again, the $5 \times 3$ filter will provide a significant reduction of decoding time compared to the 9/7. The two filters perform quite similarly in terms of lossless rate, but the $5 \times 3$ minimizes the power of the rounding noise implied by the integer lifting. All this makes such a filter particularly suitable for our application.

TABLE III
TOTAL NUMBER OF SUBBAND IMAGES TO DECODE FOR RECONSTRUCTING IMAGE $k$ FOR $L = 3$.

| $N_k/k$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $(2,2)$ | 34 | 46 | 49 | 46 | 42 | 46 | 42 | 46 |
| $(9/7)$ | 58 | 139 | 111 | 161 | 82 | 147 | 111 | 139 |

## V. MLZC

MLZC is based on the layered zero coding (LZC) algorithm [13]. The main differences between LZC of [13] and the proposed MLZC algorithm concern the underlying subband structure and the definition of the *conditioning terms*. This section starts with an overview of the basic principles of the LZC method and then details the proposed system. Particularly, Section V-A summarizes the basics of LZC and Section V-B introduces the MLZC coding principle and describes how the conditioning terms are defined and generalized for 3-D and interband conditioning.

### A. LZC

In the LZC approach, each subband is quantized and encoded in a sequence of $N$ quantization layers, $L_0, \ldots, L_{N-1}$, corresponding to progressively finer quantization step sizes

$Q_0, \ldots, Q_{N-1}$. Each quantizer produces a sequence of symbols. The symbols for quantizer $Q_0$ are encoded into the quantization layer $L_0$. The information necessary to recover the symbols for quantizer $Q_n$, given that the symbols for quantizers $Q_0, \ldots, Q_{n-1}$ are already known, is encoded into layer $L_n$. Due to the inherent embedding, the information needed to recover the subband samples with the resolution corresponding to $Q_i$ is obtained by decoding the layers $L_0$ thorough $L_i$. The LZC method is based on the observation that the most frequent symbol produced by the quantizers is the zero symbol, and achieves high efficiency by splitting the encoding phase in two successive steps:

- *Zero coding:* encodes a symbol representing the *significance* of the considered coefficients with respect to the current quantizer (i.e., $Q_n(x[i])$ being zero or nonzero);
- *Magnitude refinement:* generates and encodes a symbol defining the value of each nonzero symbol. Zero Coding exploits some spatial or other dependencies among subband samples by providing such information to a *context-adaptive* arithmetic coder [24]. Basically, the expected statistical relationships among coefficients are modeled by defining some *conditioning terms* which summarize the *significance state* of the samples belonging to a generalized neighborhood of the coefficient being encoded. Such terms are then used for entropy coding by the context adaptive arithmetic coder. Different solutions are possible for the definition of the conditioning terms, accounting for both local and wide scale neighborhoods. We refer to [13] for more details.

### B. MLZC Coding Principle

MLZC applies the same quantization and entropy coding policy as LZC to a different subband structure. All the subbands are encoded according to the layered PCM scheme. In order to detail the way the spatial and interband relationships are exploited, we use the concepts of *generalized neighborhood* and *significance state* of a given coefficient. We define generalized neighborhood of a subband sample $c(l,j,\mathbf{k})$ in subband $j$ of level $l$ and position $\mathbf{k}$ the set $GN(l,j,\mathbf{k})$ consisting of both the coefficients in a given spatial neighborhood $N(l,j,\mathbf{k})$ and the parent coefficient $c(l+1,j,\mathbf{k}')$ in the same subband at the next coarser scale, where $\mathbf{k}' = \lfloor \mathbf{k}+1/2 \rfloor$

$$GN(l,j,\mathbf{k}) = N(l,j,\mathbf{k}) \cup c(l+1,j,\mathbf{k}'). \tag{8}$$

The MLZC scheme uses the significance state of the samples belonging to a *generalized* neighborhood of the coefficient to be coded for conditioning the arithmetic coding [14].

The generating rule for the sequence of quantization stepsizes is

$$\begin{aligned} Q_i &= \frac{Q_{i-1}}{2} \quad \text{for } 0 < i < N \\ Q_0 &= 2^{\lfloor \log_2 \{\max |[c(\mathbf{k},l,j)]|\} \rfloor}. \end{aligned} \tag{9}$$

The maximum value $Q_0$ is the largest power of two within the range of the magnitude of the subband samples. The significance of a coefficient with respect to $Q_0$ thus means that the most significant bit (MSB) of the coefficient is 1. Since (9) preserves the property of each $Q_i$ being a power of two, the corresponding information associated to the significance state of a sample is the $i^{th}$ bit of its binary representation being 0 or 1. A coefficient is *significant* if at least one 1 symbol has been generated by its quantization during the previous steps, i.e., $\exists i \leq \hat{i} : \sigma_i[\mathbf{k},l,j] = 1$ where $Q_{\hat{i}}$ is the current quantization step size. For each $Q_i$, the significance state of each coefficient is determined scanning the subbands starting from the lowest resolution. For the resulting symbol, two coding modes are possible: *significance* and *refinement* mode. The significance mode is used for samples that were not significant during all the previous scans, whether they are significant or not with respect to the current threshold. For the other coefficients, the refinement mode is used. The significance mode is used to encode the significance map. The underlying model consists in assuming that if a coefficient is lower than a certain threshold, it is reasonable to expect both its spatial neighbors and its descendants being lower than a corresponding threshold too. The significance map consists of the sequence of symbols

$$\sigma_i[\mathbf{k}] = \begin{cases} 1, & \text{if } \mathcal{Q}_i[x(\mathbf{k})] \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

where $\mathbf{k} = n_x\mathbf{i} + n_y\mathbf{j}$ defines the position of the considered sample and the $\mathcal{Q}_i$ operator quantizes $x(\mathbf{k})$ with step $Q_i$. In what follows, we call *local scale* neighborhood of a coefficient $c(l,j,\mathbf{k})$ in subband $(l,j)$ and position $\mathbf{k}$ the set of coefficients lying in the spatial neighborhood $N(\mathbf{k},l,j)$. Then, we will refer to the sequence of symbols resulting from the application of $Q_i$ to the set $c(l,j,\mathbf{k})$ as to $\sigma_i[l,j,\mathbf{k}]$. The significance state of the samples in the generalized neighborhood of $c(\mathbf{k},l,j)$ is represented by some conditioning terms $\chi(\cdot)$. The *local-scale* conditioning terms $\chi^s(\cdot)$ concern spatial neighborhoods while *interband* terms $\chi_i^f(\cdot)$ account for interband dependencies

$$\chi[l,j,\mathbf{k}] = \chi^s[l,j,\mathbf{k}] + \chi^f[l,j,\mathbf{k}] \; \forall \; l \neq L. \tag{11}$$

This rule does not apply to the coarsest subbands, i.e., $l = L$, for which no parents can be identified. In this case, only the local-space contribution is used. The $\chi^s[l,j,\mathbf{k}]$ are defined as linear combinations of functions representing the significance state of one or more samples in $N(l,j,\mathbf{k})$

$$\chi^s[\mathbf{k},l,j] = \sum_{p=0}^{P-1} w_p \sigma[l,j,\mathbf{k}'] \; \forall \; \mathbf{k}' \in N \tag{12}$$

where $p = p(\mathbf{k}')$. The weights $w_p = 2^p$ are such that each term of the summation contributes to the value of the $p^{th}$ bit-plane of $\chi^s[l,j,\mathbf{k}]$, $P$ is the bit depth of $\chi^s[l,j,\mathbf{k}]$, and $\sigma$ is the distribution of the sequence of symbols $\sigma(l,j,\mathbf{k})$ generated by quantizer $Q_i$. The set of local-scale bi-dimensional settings that have been tested is illustrated in Fig. 9. Contexts number 1 to 5 only account for coefficients which have already been encoded in the current step, while those of number 6 to 8 also use samples which will be successively encoded in the current step, so that their significance state refers to the previous scan, i.e., quantization factor. Since the number of entries of the probability table used by the context adaptive arithmetic coder is equal to the number of different values that $\chi$ can take, the grouping of $\sigma$ may become unavoidable when dealing with local-space neighborhoods of wide support. This sets an upper limit on the

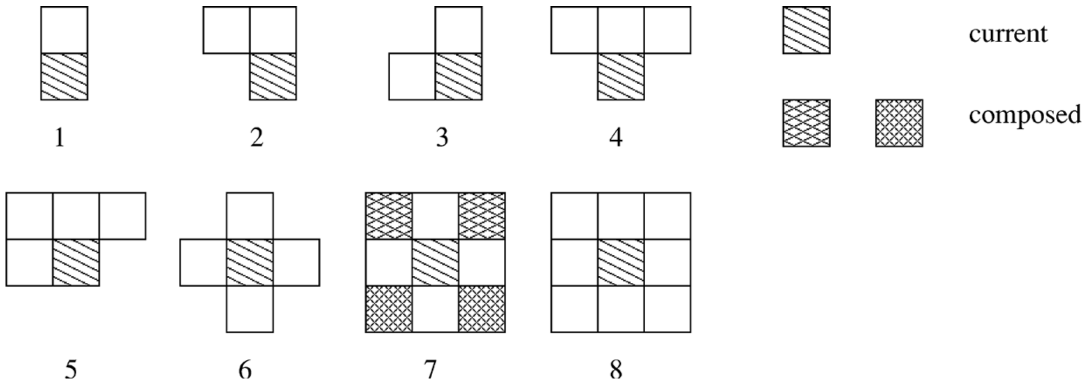


Fig. 9. Two-dimensional contexts.

number of possible contexts for avoiding the degradation of performance of the arithmetic coder. The contexts shown in Fig. 9 correspond to the following expressions:

$$\begin{aligned}
\chi^{s,(1)}[\mathbf{k}] =&\sigma[\mathbf{k}-\mathbf{i}]\\
\chi^{s,(2)}[\mathbf{k}] =&2\sigma[\mathbf{k}-\mathbf{i}-\mathbf{j}]+\sigma[\mathbf{k}-\mathbf{i}]\\
\chi^{s,(3)}[\mathbf{k}] =&2\sigma[\mathbf{k}-\mathbf{j}]+\sigma[\mathbf{k}-\mathbf{i}]\\
\chi^{s,(4)}[\mathbf{k}] =&4\sigma[\mathbf{k}-\mathbf{i}-\mathbf{j}]+2\sigma[\mathbf{k}-\mathbf{i}]+\sigma[\mathbf{k}-\mathbf{i}+\mathbf{j}]\\
\chi^{s,(5)}[\mathbf{k}] =&8\sigma[\mathbf{k}-\mathbf{i}-\mathbf{j}]+4\sigma[\mathbf{k}-\mathbf{i}]+2\sigma[\mathbf{k}-\mathbf{i}+\mathbf{j}]+\sigma[\mathbf{k}-\mathbf{j}]\\
\chi^{s,(6)}[\mathbf{k}] =&8\sigma[\mathbf{k}-\mathbf{i}]+4\sigma[\mathbf{k}-\mathbf{j}]+2\sigma[\mathbf{k}+\mathbf{j}]+\sigma[\mathbf{k}+\mathbf{i}]\\
\chi^{s,(7)}[\mathbf{k}] =&32(\sigma[\mathbf{k}-\mathbf{i}-\mathbf{j}]\vee\sigma[\mathbf{k}-\mathbf{i}+\mathbf{j}])\\
&+16(\sigma[\mathbf{k}+\mathbf{i}-\mathbf{j}]\vee\sigma[\mathbf{k}+\mathbf{i}+\mathbf{j}])+\\
&+8\sigma[\mathbf{k}-\mathbf{i}]+4\sigma[\mathbf{k}-\mathbf{j}]+2\sigma[\mathbf{k}+\mathbf{j}]+\sigma[\mathbf{k}+\mathbf{i}]\\
\chi^{s,(8)}[\mathbf{k}] =&128\sigma[\mathbf{k}-\mathbf{i}-\mathbf{j}]+64\sigma[\mathbf{k}-\mathbf{i}]\\
&+32\sigma[\mathbf{k}-\mathbf{i}+\mathbf{j}]+16\sigma[\mathbf{k}-\mathbf{j}]+\\
&+8\sigma[\mathbf{k}+\mathbf{j}]+4\sigma[\mathbf{k}+\mathbf{i}-\mathbf{j}]\\
&+2\sigma[\mathbf{k}-\mathbf{i}]+\sigma[\mathbf{k}+\mathbf{i}+\mathbf{j}]
\end{aligned}\tag{13}$$

where $\mathbf{i}$ and $\mathbf{j}$ are the unit vectors of the reference system $(x,y)$, and the indexes identifying the subbands have been subtended. MLZC enables 3-D local-scale conditioning terms. Even though it is reasonable to expect that the use of 3-D contexts would improve the coding efficiency, because of the exploitation of the full correlation among DWT coefficients, some care must be devoted to the design of $\chi^s[\mathbf{k}]$ in order to keep the dimensionality of the conditioning space sufficiently small. The 3-D local-scale conditioning terms have been obtained by extending to the third dimension the set of the $M$ most performant 2-D contexts. The support $N(l,j,\mathbf{k})$ of each selected 2-D context has been extended to the adjacent subband images as illustrated in Fig. 10. According to our conventions, the subband image with index $(\nu-1)$ is scanned before that with index $\nu$, making the significance state of the corresponding samples with respect to the current quantization level available for its encoding. Conversely, only the significance state relative to the previous scan is known for the subband image of index $(\nu+1)$. Since we expect a more pronounced correlation among the significance states of adjacent samples within the same scan, we decided to give more degrees of freedom to the extension of the inter-scale conditioning term in the previous $(\nu-1)$ than the next $(\nu+1)$ subband images. Particularly, two possible configurations have been tested for $(\nu-1)$. The first one consists of the sample with same $(x,y)$ coordinates as the one being encoded. The second is cross-shaped and it is centered in $(x,y)$ on the previous subband image. In this case, the significance state of the coefficients at $(\nu-1)$ are combined either all together or by peers. For $(\nu+1)$, only the sample with same $(x,y)$ coordinates has been used. The resulting configurations are illustrated in Fig. 10. The name associated with each context is in the form $(n_1 n_p n_3)$ where the indexes identify the 2-D context in the previous $(\nu-1)$, current $(\nu)$ and next $(\nu+1)$ layer, respectively. The case $n_1, n_3 = 0$ reveals that no samples have been considered in the corresponding layer. The significance state of the samples in the adjacent planes determines $\chi^s[\mathbf{k},l,j]$ according to

$$\begin{aligned}
\chi^{s,(1n_p0)}[\mathbf{k}] =&w_P\sigma[\mathbf{k}-\mathbf{q}]+\chi^{s,(n_p)}[\mathbf{k}]\\
\chi^{s,(1n_p1)}[\mathbf{k}] =&w_{P+1}\sigma[\mathbf{k}+\mathbf{q}]+w_P\sigma[\mathbf{k}-\mathbf{q}]+\chi^{s,(n_p)}[\mathbf{k}]\\
\chi^{s,(2n_p0)}[\mathbf{k}] =&w_{P+1}(\sigma[\mathbf{k}-\mathbf{i}-\mathbf{q}]\vee\sigma[\mathbf{k}+\mathbf{i}-\mathbf{q}]\\
&\times\vee\sigma[\mathbf{k}-\mathbf{j}-\mathbf{q}]\vee\sigma[\mathbf{k}+\mathbf{j}-\mathbf{q}])+\\
&+w_P\sigma[\mathbf{k}-\mathbf{q}]+\chi^{s,(n_p)}[\mathbf{k}]\\
\chi^{s,(2n_p1)}[\mathbf{k}] =&w_{P+2}\sigma[\mathbf{k}+\mathbf{q}]+\\
&+w_{P+1}(\sigma[\mathbf{k}-\mathbf{j}-\mathbf{q}]\vee\sigma[\mathbf{k}+\mathbf{j}-\mathbf{q}]\\
&\times\vee\sigma[\mathbf{k}-\mathbf{i}-\mathbf{q}]\vee\sigma[\mathbf{k}+\mathbf{i}-\mathbf{q}])\\
&+w_P\sigma[\mathbf{k}-\mathbf{q}]+\chi^{s,(n_p)}[\mathbf{k}]\\
\chi^{s,(3n_p0)}[\mathbf{k}] =&w_{P+2}(\sigma[\mathbf{k}-\mathbf{i}-\mathbf{q}]\vee\sigma[\mathbf{k}+\mathbf{i}-\mathbf{q}])\\
&+w_{P+1}(\sigma[\mathbf{k}-\mathbf{j}-\mathbf{q}]\vee\sigma[\mathbf{k}+\mathbf{j}-\mathbf{q}])+\\
&+w_P\sigma[\mathbf{k}-\mathbf{q}]+\chi^{s,(n_p)}[\mathbf{k}]\\
\chi^{s,(3n_p1)}[\mathbf{k}] =&w_{P+3}\sigma[\mathbf{k}+\mathbf{q}]+w_{P+2}(\sigma[\mathbf{k}-\mathbf{i}-\mathbf{q}]\\
&\times\vee\sigma[\mathbf{k}+\mathbf{i}-\mathbf{q}])+\\
&+w_{P+1}(\sigma[\mathbf{k}-\mathbf{j}-\mathbf{q}]\vee\sigma[\mathbf{k}+\mathbf{j}-\mathbf{q}])\\
&+w_P\sigma[\mathbf{k}-\mathbf{q}]+\chi^{s,(n_p)}[\mathbf{k}]
\end{aligned}\tag{14}$$

where $\mathbf{k}=x\mathbf{i}+y\mathbf{j}+z\mathbf{q}$ is the 3-D coordinate vector and $w_P=2^P$, $P$ being the position of the MSB of $\chi^{s,(n_p)}$. Results show that the spatial contexts leading to better performances correspond to $n_p=6$, 7, and 8. Their 3-D extension leads to the triplets $(n_1 n_p n_3)$ with $n_1=1$, 2, 3, and $n_3=0$, 1.

### C. Interband Conditioning

The observed self-similarity among subbands within the subband tree makes the parent $c(l+1,j,\mathbf{k}')$ of the current coefficient $c(l,j,\mathbf{k}')$ the most natural candidate for interband

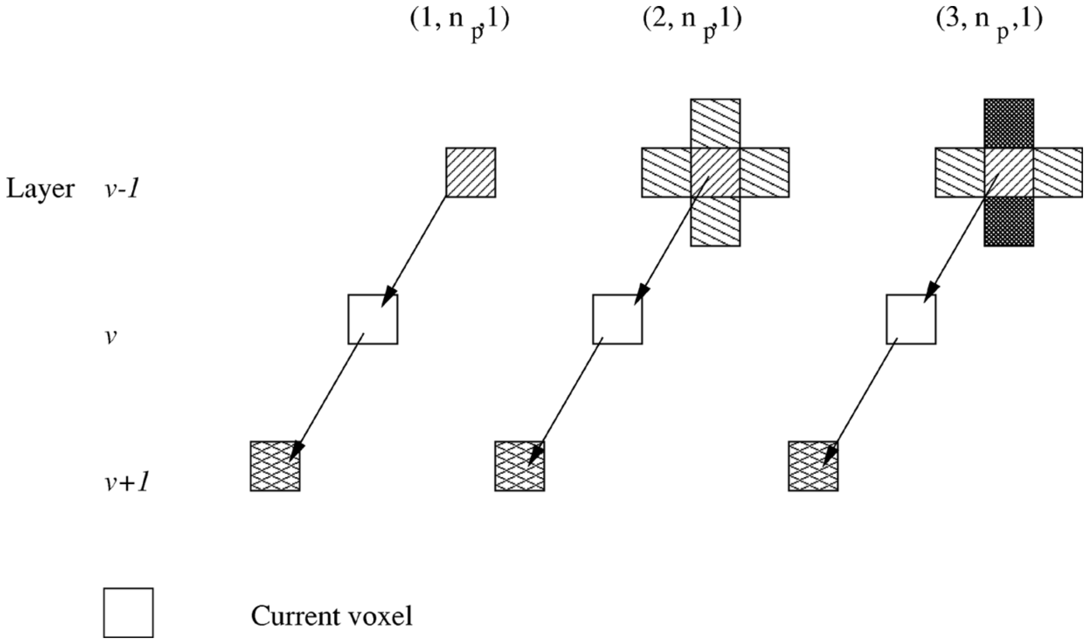


Fig. 10. Extension of the spatial support in the previous ($\nu - 1$) and the next ($\nu + 1$) subband images. Squares with same pattern represent voxels whose significance states are combined in the definition of the corresponding $\chi^s[\mathbf{k}, l, j]$.

conditioning. Accordingly, the expression for the interband conditioning term is

$$\chi^f[l, j, \mathbf{k}] = w_{P_{\mathrm{MSB}}} \sigma[l+1, j, \mathbf{k}'] \tag{15}$$

where $w_{P_{\mathrm{MSB}}} = 2^{P_{\mathrm{MSB}}}$ is the weight needed to define the MSB of the *global* context

$$\chi[l, j, \mathbf{k}] = \chi^s[l, j, \mathbf{k}] + \chi^f[l, j, \mathbf{k}] \; \forall \; l \neq L. \tag{16}$$

Again, this rule does not apply to the coarsest subbands, i.e., $l = L$, for which no parents can be identified. In this case, only the local-space contribution is used.

## VI. Bitstream Syntax

This section describes the bitstream syntax, i.e., the way the encoded information is organized. The ability to access any 2-D image of the set constrains the bitstream structure. In all the modes (G-PROG, LPL-PROG, and LPL), the subbands are scanned starting from coarsest resolution. The signal approximation $LLL_{l=L}$ is encoded first, and all the subbands at level ($l + 1$) are processed before any subband at the next finer level $l$. What makes the difference among the considered working modalities are the order of encoding of the subband images and the placement of the markers. We describe them in what follows, starting from the less constrained one.

### A. G-PROG Mode

The set of quantizers is applied to the whole set of subband images before passing to the next subband. The scanning order follows the decomposition level: all subbands at level $l$ are scanned before passing to level ($l - 1$). In other words, during step $i$, the quantizer $Q_i$ is applied to each image of each subband. This enables scalability on the whole volume: decoding can be stopped at any point into the bitstream. In this mode, the compression ratio is maximized, but the 3-D encoding/2-D decoding functionalities are not enabled.

### B. LPL-PROG Mode

This scheme is derived from the G-PROG mode by adding a marker into the bitstream after encoding every quantization layer of every subband image (see Fig. 2). Since the quantizers are successively applied – as in the G-PROG mode – subband-by-subband and, within each subband, image-by-image, progressiveness by quality is allowed on both the whole volume and any 2-D image, provided that 2-D local-scale conditioning is used. The drawback of this solution is the overloading of the encoded information.

### C. LPL Mode

One way of reducing the overloading implied by the LPL-PROG mode is to apply the whole set of quantizers to each subband image of position $\nu$ along the $z$ axis before switching to the next one ($\nu + 1$). The progressive by quality functionalities are suboptimal on both the single images and the whole volume. This degrades the performance in the lossy regime with respect to the G-PROG mode. Quality scalability could be improved by an *ad-hoc* procedure for rate allocation. We leave this subject for future investigation.

As previously mentioned, all these configurations have been tested in conjunction with both the 2-D and 3-D contexts. Nevertheless, the desired 3-D encoding/2-D decoding capabilities constrain the choice to bi-dimensional contexts without interband conditioning.

## VII. Results and Discussion

The performance of the MLZC 3-D encoding/2-D decoding system has been evaluated on the four datasets illustrated in Fig. 11.

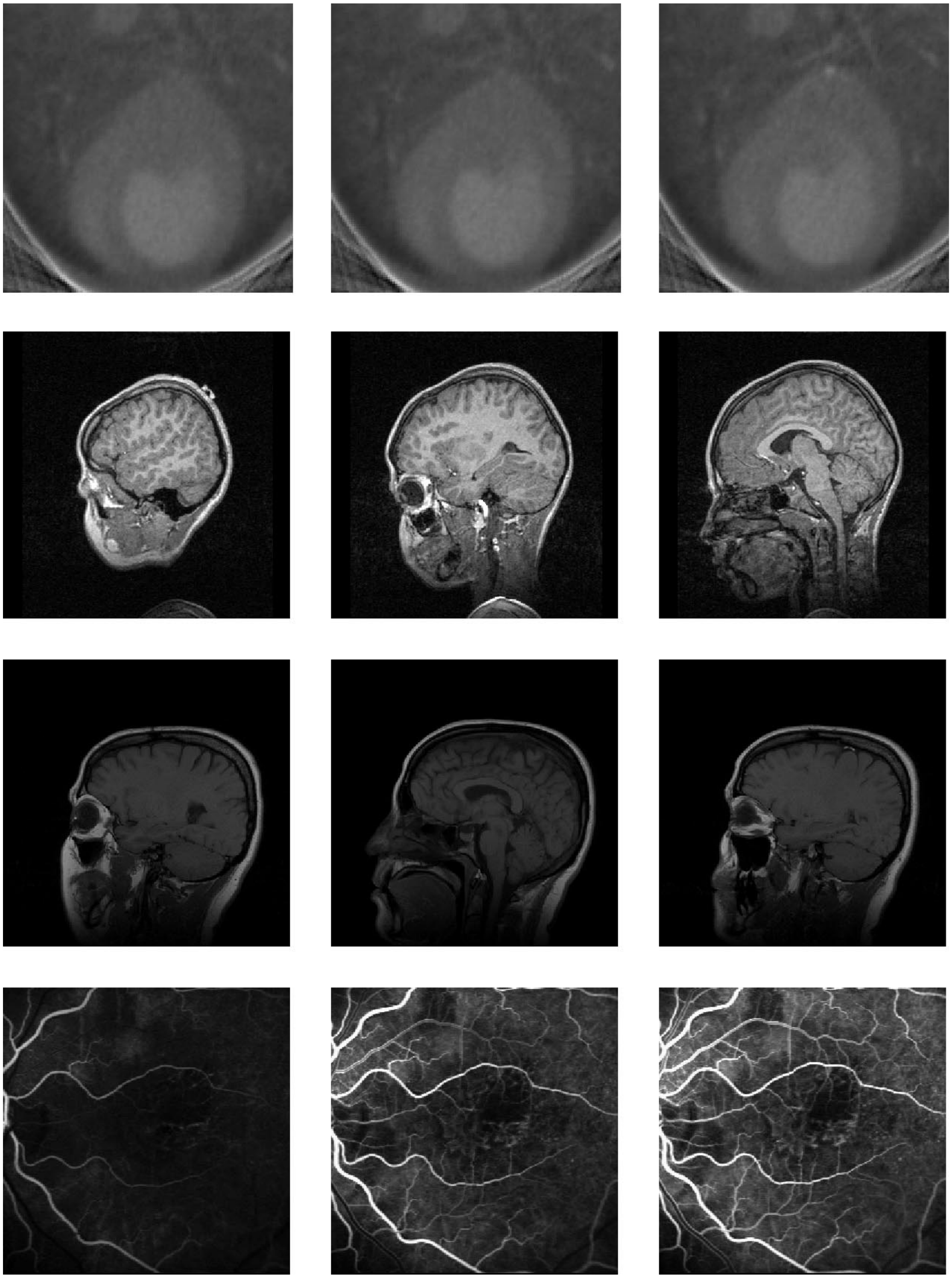


Fig. 11. Samples of the 3-D dataset. First line: DSR images. The brightest region in the middle represents the left ventricle of a canine heart. Second line: human head MRI, saggital view. Third line: MR-MRI; Fourth line: opthalmologic angiography sequence (2-D+time). The brightness results from the flow of the contrast medium into the vessels.

- Dynamic spatial reconstructor (DSR). The complete DSR set consists of a four-dimensional (3-D+time) sequence of 16 3-D cardiac CT data. The imaging device is a unique ultra-fast multislice scanning system built and managed by the Mayo Foundation. Each acquisition corresponds to one phase of the cardiac cycle of a canine heart and is composed of 107 images of size $128 \times 128$ pixels. A voxel represents approximately $(0.9 \text{ mm})^3$ of tissue.
- MRI head scan. This volume consists of 128 images of size $256 \times 256$ pixels representing the saggital view of an human head.
- MR-MRI head scan. This volume has been obtained at the Mallinckrodt Institute of Radiology (Washington Univer-

sity) [5]. It consists of 58 images of a saggital view of the head of size 256 × 256 pixels. Since this dataset has also been used as a test set by other authors [5], [3], [25] it allows to compare the compression performances of the MLZC to other 3-D systems.

- Opthalmologic angiography (ANGIO). The ANGIO set is a 3-D sequence (2-D+time) of angiography images of a human retina, consisting of 52 images of 256 × 256 pixels.

The different characteristics of the considered datasets make the resulting test set heterogeneous enough to be used for characterizing the system. The DSR volume is very smooth and features high correlation among voxels along all the three spatial dimensions. This makes it very easy to code and particularly suitable for the proposed coding system. It represents the "best case" test set, for which the coding gain of 3-D over 2-D systems is expected to be the highest. Conversely, the ANGIO dataset can be considered as the "worst case" for a wavelet-based coding system. The images are highly contrasted: very sharp edges are juxtaposed to a smooth background. Wavelet-based coding techniques are not suitable for this kind of data. The edges spread out in the whole subband structure generating a distribution of non zero coefficients whose spatial arrangement cannot be profitably exploited for coding. This is due to the fact that wavelets are not suitable descriptors of images with sharp edges [26], [27]. The problem of image representation is a hot topic in the field of signal processing, and is subject to a deep investigation. The MR-MRI set has been included for sake of comparison with the results provided by other authors [5]. Nevertheless, we do not consider it as representative of a real situation because it went through some preprocessing. In particular, it has been interpolated, scaled to isotropic 8-bit resolution and thresholded. Finally, the characteristics of the MRI set lie in between. Noteworthy, the structure and semantics of the MRI images make the volume suitable for an *object-based* approach to coding.

The 3-D/2-D MLZC system is a good the trade off between the gain in coding efficiency provided by fully 3-D algorithms and the fast access to data provided by 2-D coding systems, where each image is treated independently. As it allows to access any 2-D image without decoding the entire volume, it minimizes the decoding time while improving compression. Accordingly, the evaluation of the 3-D/2-D MLZC system performance concerns both coding gain and decoding delay. Different 2-D and 3-D coding algorithms have been considered. The benchmark for the 3-D case is the 3-D generalization of the well known EZW coding algorithm [4]. Because of the interband relationship among the coefficients within the subband tree, EZW-3-D does not allow 3-D/2-D functionalities. In fact, the parent-children relationship exploited for coding prevents the independent access to the coefficients of different subbands. The MLZC system has been analyzed by determining the lossless rate corresponding to the complete set of contexts in each working mode. As was explained in Section V-B, the 3-D spatial supports for the conditioning terms result from the extension of one of the most performant bi-dimensional configurations to the adjacent planes along the $z$ axis. The context that has been chosen for the definition of the 3-D conditioning terms is the (060). Indeed, results show that it is among the three most performant 2-D contexts on all the datasets in LPL-PROG as well as LPL mode. Fig. 12 shows the lossless rate as a function of the spatial conditioning terms (i.e., without interband conditioning) for DSR. As expected, the best performances in terms of lossless rate are obtained in the G-PROG mode. As it is the case for EZW-3-D, the G-PROG mode does not allow 2-D decoding. In the LPL and LPL-PROG modes, such a functionality is enabled at the expense of coding efficiency, which decreases because of the additional information to be encoded to enable random access.

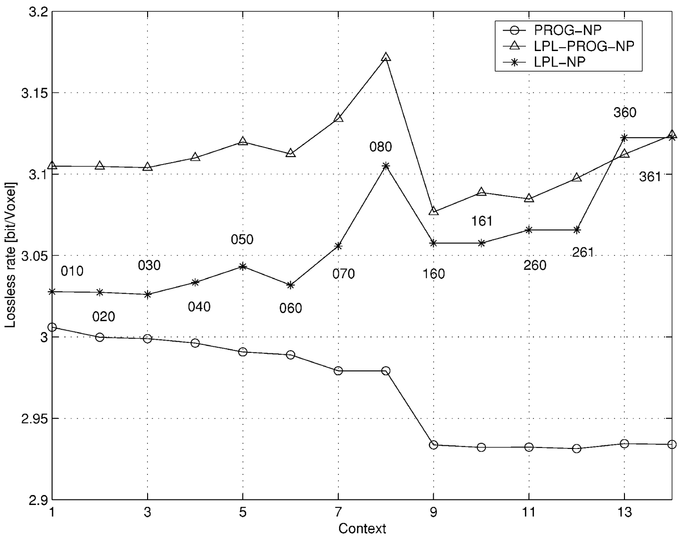


Fig. 12. Lossless rates as a function of the conditioning terms for DSR. No interband conditioning has been applied (NP). The LPL-PROG mode enables 2-D decoding while allowing progressiveness by quality on the 2-D images. Due to the markers added to the codestream, its performances are in between the G-PROG and LPL modes.

One of the constraints posed by 2-D decoding is that no interband conditioning can be used. Even though the exploitation of the information about the significance of the parent within the subband hierarchy can be fruitful in some cases, the compression performances are not much affected by such a limitation. For example, Fig. 13 illustrates the impact of interband conditioning on the G-PROG and LPL-PROG modes for DSR. In the G-PROG mode, the lossless rate is slightly improved for 2-D spatial conditioning, while it is basically left unchanged for 3-D contexts. Conversely, in the LPL-PROG mode performance is slightly degraded. This is related to the sparseness of the samples in the conditioning space. Due to the smoothness along the $z$ axis, the exploitation of the significance state of the neighboring voxels is fruitful for entropy coding up to a limit where the dimension of the conditioning space becomes so large that the available samples are not sufficient to be representative of the statistics of the symbols. The point where such a critical condition is reached depends on the characteristics of the dataset and, in particular, on its size. In general, larger volumes take advantage of wider spatial supports and interband conditioning. The observed dependency of the lossless rate on the design parameters of the conditioning terms (i.e., the spatial support and the use of interband conditioning) also applies to the bi-dimensional version of the MLZC algorithm, labeled as 2-D-PROG. Again, the efficiency of the entropy coding increases with the size of the spatial support up to a limit where the sparseness of

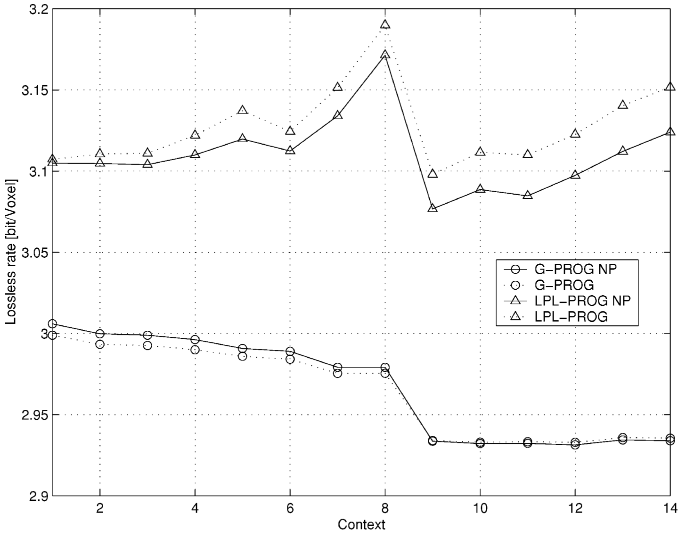


Fig. 13. Lossless rates for DSR. Impact of interband conditioning. The G-PROG and LPL-PROG modes are compared. Continuous line: without interband conditioning (NP); dotted line: with interband conditioning.

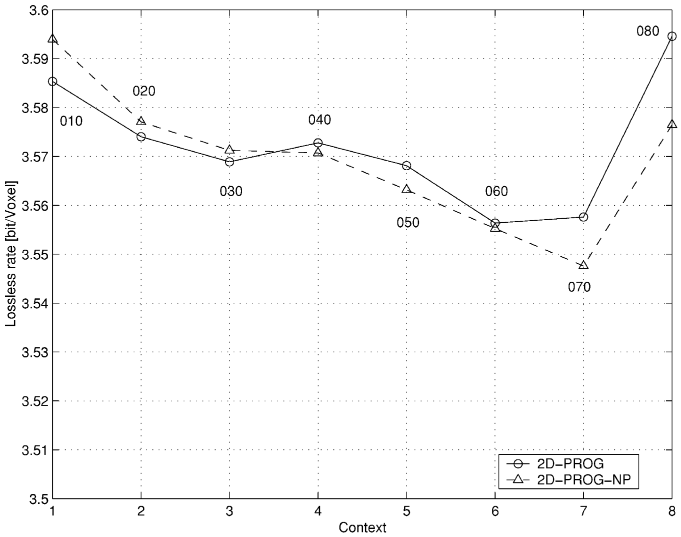


Fig. 14. Lossless rates for DSR set with the PROG-2-D algorithm. Impact of interband conditioning.

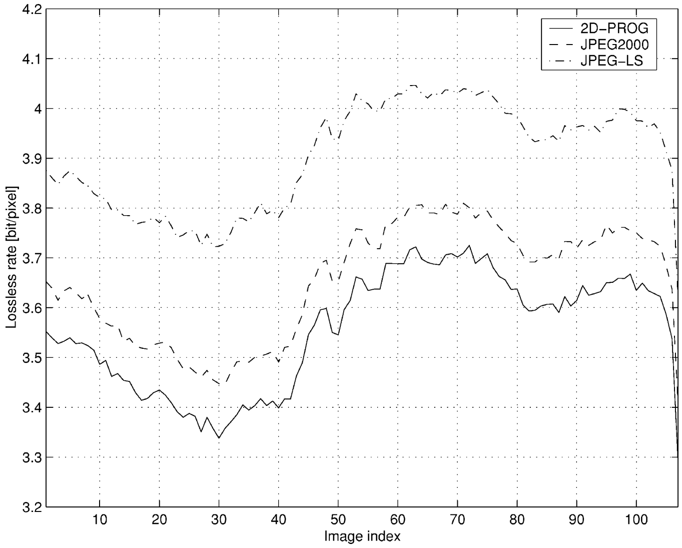


Fig. 15. Performance of 2-D algorithms on DSR dataset. The lossless rate of each 2-D image is represented as a function of its position along the $z$ axis. Continuous line: 2-D-PROG; dashed line: JPEG2000; dash-dot line: JPEG-LS. The lossless rate provided by the MLZC algorithm in the LPL modality is 3.06 bit/voxel.

the conditioning space does not allow an adequate representation of the statistics of the symbols to be encoded. Fig. 14 gives an example. For each context, the lossless rate has been averaged over the entire set of 2-D images of the volume. The (060) and (070) spatial supports lead to the highest compression ratios. However, the impact of the conditioning term is not very sensible (about 1.2% of the minimum rate).

The bench-mark for 2-D systems is the new coding standard for still images JPEG2000 [15], [28]. JPEG2000 has been designed to overcome some of the limitations of JPEG [29] and supports a wide variety of features. Among others are improved compression efficiency, lossy to lossless performances, scalability (by quality and by resolution) and region of interest (ROI)-based functionalities. As MLZC and EZW-3-D, it is wavelet-based and the DWT is implemented by the lifting steps scheme. Fig. 15 compares the performance of the different 2-D algorithms for DSR. In this case, the 2-D-PROG mode outperforms both JPEG2000 and JPEG-LS. Fig. 16 shows the resulting lossless rate as a function of the image index for the three datasets. The coding parameters, namely the filter used and the number of levels of decomposition, are the same as for the other methods. For DSR [Fig. 16(a)] the curve is quite flat and the average compression ratio is quite high (about 2.2), indicating that all the images are equally easy to code. In the case of MRI [Fig. 16(b)], the shape of the curve reflects the trend of the number of “nonbackground” pixels of the images with the position along the $z$ axis. Accordingly, the central images are encoded at a higher rate than the boundary ones. The oscillatory trend of MR-MRI [Fig. 16(c)] is probably due to preprocessing. Finally, the slope of the line for ANGIO [Fig. 16(d)] data is due to the increasing number of visible vessels due to the flow of the contrast medium which augments the contrast reducing the efficiency of the coder. The average compression ratio for these last two cases is about 1.8.

Table IV summarizes the performance of the different algorithms and working modes. The (060) and (160) contexts were chosen as references, and no interband conditioning was used. As was the case for JPEG2000, the data concerning the 2-D algorithms were obtained by running them on the whole set of 2-D images and taking the average of the resulting set of lossless rates. For the old JPEG standard (JPEG-LS), all of the seven available prediction modes were tested and the one providing the best performance (corresponding to $K = 7$ for all the datasets) was retained. As it was reasonable to expect, the coding gain provided by the 3-D over the 2-D systems depends on the amount of correlation and smoothness along the $z$ axis. Accordingly, it is quite pronounced for DSR and MR-MRI, for which the LPL mode leads to a rate saving of about 16%, respectively, 33% over JPEG2000, while it is lower for both MRI and ANGIO. For MR-MRI some results are available in the literature. We refer here to those presented in [5]. The first one was obtained for $L = 3$ and using the integer version of the $5 \times 3$

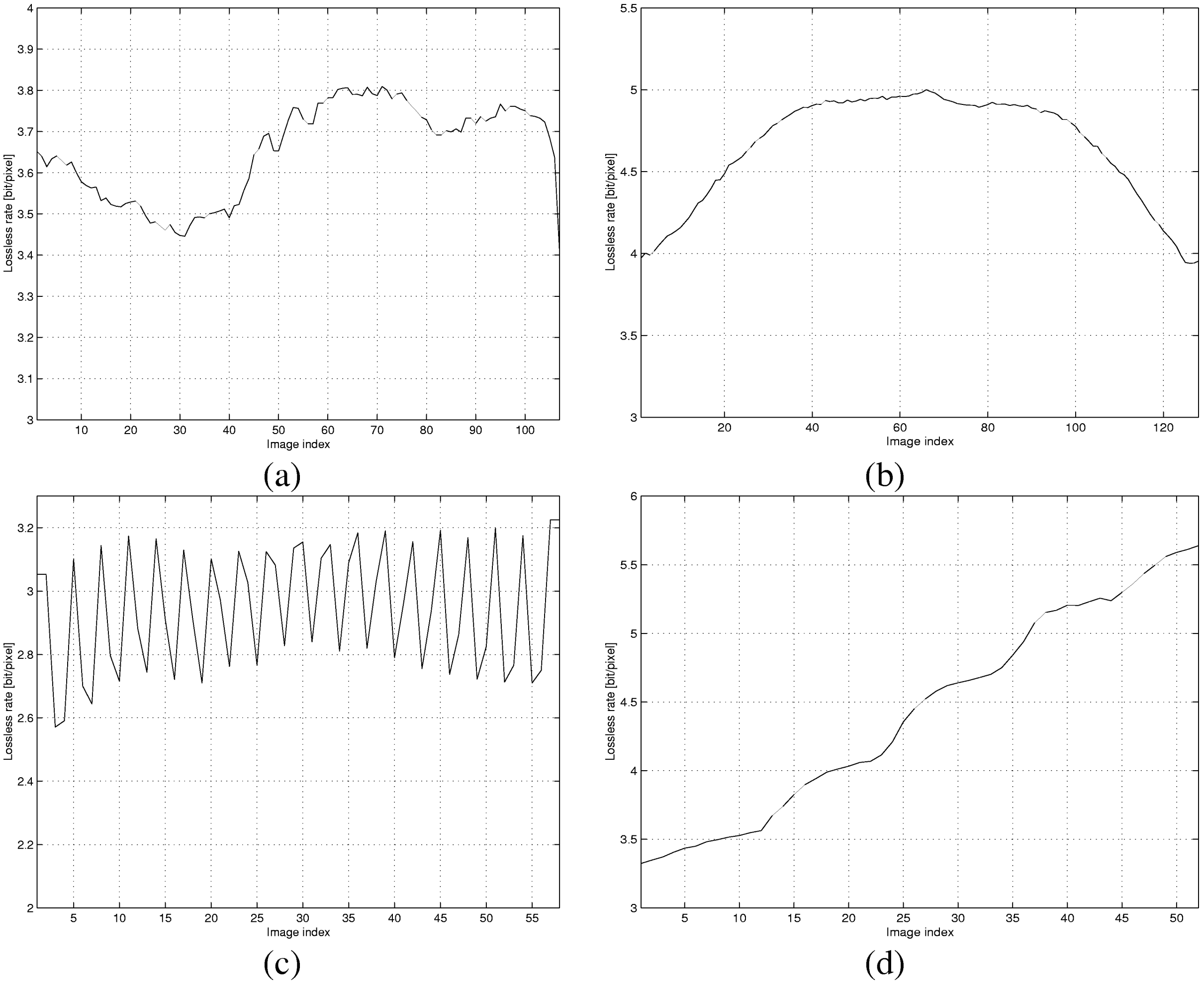


Fig. 16. Lossless rates obtained by applying JPEG2000 standard to (a) DSR, (b) MRI (c) MR-MRI and (d) ANGIO datasets.

TABLE IV
LOSSLESS PERFORMANCES (BIT/VOXEL) WITH $5 \times 3$ FILTER. THE DECOMPOSITION DEPTH IS $L = 4$ FOR DSR, MRI AND MR-MRI, AND $L = 3$ FOR ANGIO. THE TWO VALUES CORRESPOND TO THE CONTEXTS (060) AND (160) FOR EACH DATASET. NO INTERBAND CONDITIONING IS USED. THE LAST COLUMN IS THE PERCENTAGE OF RATE SAVING PROVIDED BY MLZC-LPL OVER JPEG2000.

| ctx/mode | G-PROG | LPL-PROG | LPL | EZW-3D | 2D-PROG | JPEG2000 | JPEG-LS | $\Delta$[%] |
|---|---|---|---|---|---|---|---|---|
| DSR | 2.99 | 3.11 | 3.03 | 2.88 | 3.56 | 3.62 | 3.90 | 16.3 |
| | 2.93 | 3.08 | 3.06 | | | | | |
| MRI | 4.58 | 4.63 | 4.55 | 4.46 | 4.62 | 4.65 | 5.10 | 2.2 |
| | 4.52 | 4.60 | 4.52 | | | | | |
| MR-MRI | 2.24 | 2.28 | 2.24 | 2.271 | 2.92 | 2.95 | 3.437 | 33.06 |
| | 2.19 | 2.23 | 2.22 | | | | | |
| ANGIO | 4.19 | 4.23 | 4.20 | 4.18 | 4.41 | 4.43 | 3.87 | 5.2 |
| | 4.16 | 4.22 | 4.21 | | | | | |

filter. The second was based on a two levels integer transform with the $(1+1, 1)$ filter on 16 slice coding units, and the compression efficiency data were averaged over the volume. The coding scheme — 3-D CB-EZW — was a version of EZW-3-D exploiting context modeling. The corresponding lossless rates were 2.285 and 2.195 bit/voxel. The best MLZC mode — the G-PROG without interband conditioning and context (370) — results in 2.143 bit/voxel.

The case of MRI is particularly interesting and deserves further comments. The majority of the voxels (about 80%) represent a “nondiagnostically relevant” information, i.e., the “background.” This makes it particularly suitable for

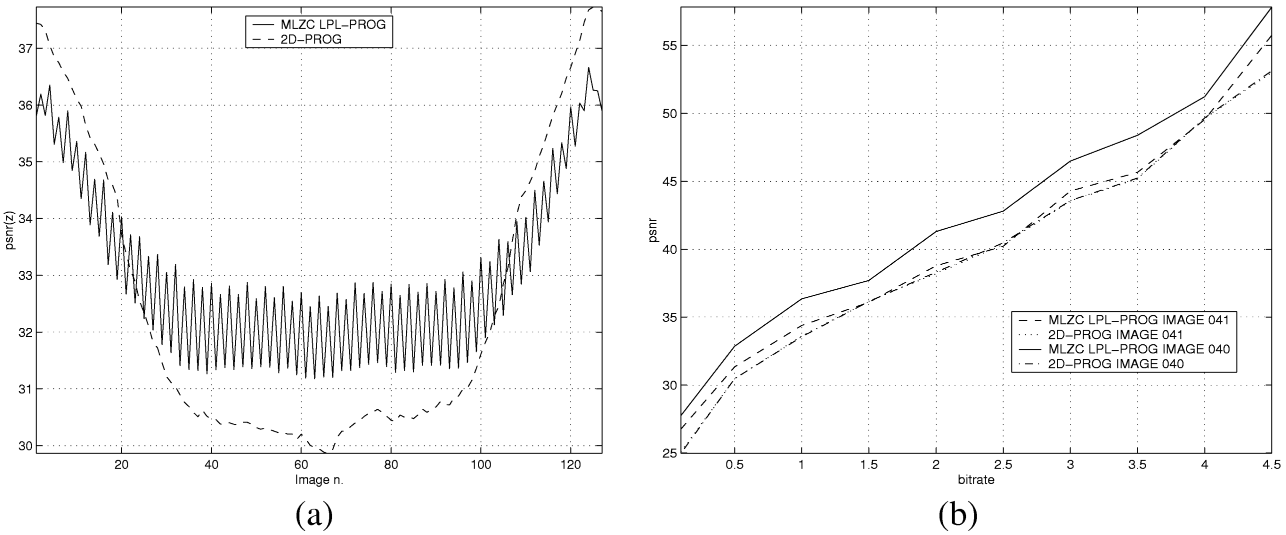


Fig. 17. Performances in lossy regime of LPL-PROG and 2-D-PROG on MRI. (a) PSNR as a function of the image index ($z$ coordinate); each image has been independently decoded at 0.5 bit/pixel. (b) PSNR as a function of the bitrate for images 40 and 41 of the dataset.

object-based coding: the object of interest (usually called ROI) is encoded independently and with highest priority than the rest of the image. In the framework of ROI-based coding, the weight assigned to a voxel depends on its semantics, which is assumed to be the criterion for the judicious allocation of the available resources (e.g., bit-budget, bandwidth). In this sense, the improvement in coding efficiency is related to the *prioritization* of the information to be transmitted. An extension of 3-D/2-D MLZC featuring ROI-based functionalities is currently under development. It will allow random access to any object of any 2-D image of the dataset at the desired up to lossless quality [7].

The best compression performances for ANGIO are obtained by JPEG-LS. As mentioned above, such a dataset is not suitable for wavelet-based coding, so that other algorithms can easily be more effective. Nevertheless, the LPL method provides an improvement of about 5% over JPEG2000. The 3-D encoding/2-D decoding approach can thus be considered as a good tradeoff between compression efficiency and the availability of higher level functionalities which are not available with JPEG-LS. Among these are quality scalability and both lossy and lossless representations of the encoded information within the same codestream.

The evaluation of the performance in lossy regime was out of the scope of this paper. The observed oscillatory trend of the PSNR along the coordinate axis entails the analysis of both the rounding noise implied by integer lifting [18] and the quantization noise. Fig. 17 gives an example. After encoding the volume in the LPL-PROG mode, every image of the dataset has been independently decoded at the resolution of 0.5 bit/pixel. Fig. 17(a) compares the corresponding PSNR to that obtained by separately encoding and decoding each image with the 2-D version of the algorithm (2-D-PROG) at the same rate. It is important to notice that the control over the decoding bitrate on the single 2-D images is only possible when they are decoded one by one. On average, the 3-D method outperforms the 2-D counterpart on the central portion of the dataset (images 20–100), which are not dominated by the background. In this case, the oscillation has period one, namely every other image has better quality. This makes the improvement in image quality provided by the 3-D system dependent on the position of the image within the dataset. Fig. 17(b) shows the PSNR for images of index 40 and 41 as a function of the decoding rate. The maximum and mean increase in the PSNR are about 4.7 and 2.7 dB for image 40, and about 0.8 and 2.7 for image 41, respectively. As mentioned above, this is due to both the rounding and the quantization noise. We are currently investigating this issue to extend the model proposed in [18] to the 3-D case and define a quantization policy ensuring a more uniform decoding quality, for a given rate, over the entire volume.

The other parameter to be considered for the evaluation of the performances of the 3-D/2-D MLZC system is the decoding delay, which entails the analysis of the complexity. In this paper, we did not address the problem of computational efficiency and no optimization was performed. Consequently, the decoding time is suboptimal and as such it is not meaningful neither representative of what it would be in the optimized version. As a general comment, even though a more detailed analysis of the complexity is required for the evaluation of the global performance of the system, there is clearly a tradeoff between the improvement in compression efficiency and the increase in complexity when switching from 2-D to 3-D systems. Nevertheless, this does not compromise their usefulness. What is important is the *absolute* decoding time, namely the time the user has to wait to access the decoded image, rather than the *relative* increase with respect to the 2-D counterpart. We expect our system being able to reach a decoding time of less than one second per image after optimization. Last but not least, large PACS can easily incorporate high processing power (e.g., a multiprocessor architecture) at a price that is negligible with respect to the whole cost of a PACS. Therefore we consider that the complexity of our method is not a major issue for real implementations.

In our opinion, the proposed approach to coding has a high potential, especially if combined with ROI-based functionalities.

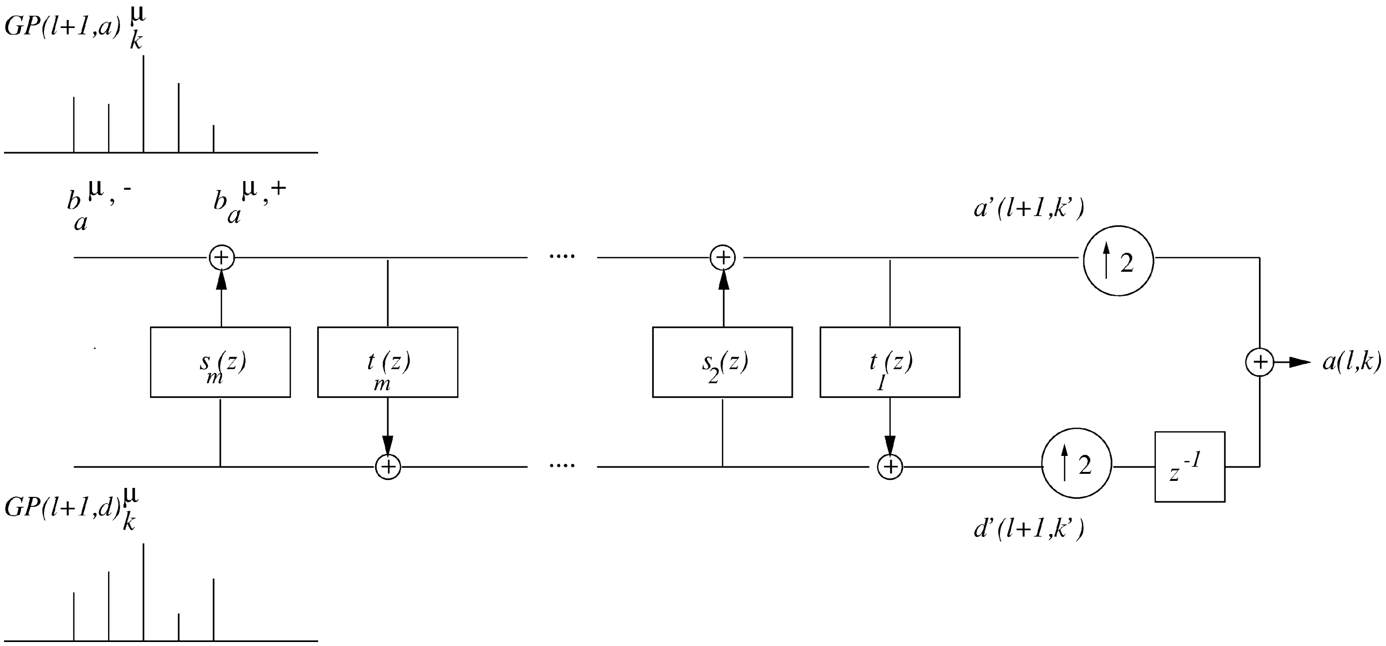


Fig. 18. Generalized projection of the subband sample $a(l,k)$ in the approximation band of level $l$, to the next coarser scale subbands. Depending on $k$ being even or odd, the sample to recover is $a'(l+1,k')$ or $d'(l+1,k')$, respectively. Accordingly, $\mu = a$ in the first case, and $\mu = b$ in the second.

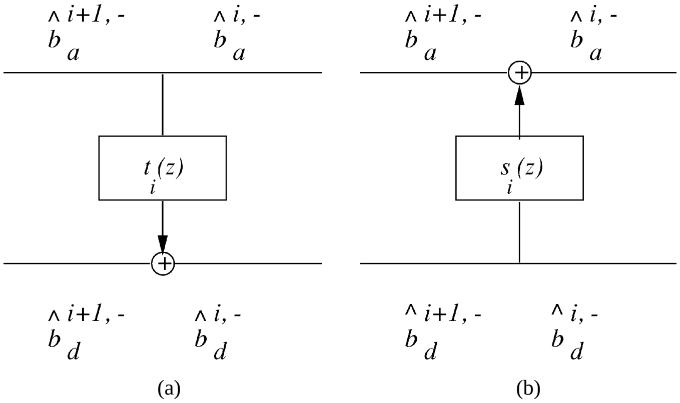
## VIII. Conclusion

We presented a wavelet-based coding system featuring 3-D encoding/2-D decoding capabilities. In this way, the improvement in coding efficiency provided by 3-D algorithms can be obtained without sacrificing the fast access to single 2-D images. Data decorrelation is performed by a fully 3-D-DWT. The implementation of the transform by the lifting scheme enables lossless functionalities, permits the in-place implementation, minimizing the run-time memory allocation, and reduces the computational complexity up to a factor 4. The set of subband images needed to reconstruct a given 2-D image depends on the length of the synthesis filters and the number of decomposition levels. The 2-D decoding mode is obtained by independently encoding each subband image. The amount of overloading of the bitstream in the different working modalities depends on many factors, like the dynamic range of the transformed coefficients, the decomposition depth, the statistics of the source and the availability of the PSNR scalability. The performance of the system was compared to that of other state-of-the-art 2-D and 3-D algorithms, including JPEG2000.

Results show that 3-D/2-D MLZC has an high potential, especially in the framework of the emerging model-based approach to coding. Accordingly, we are improving our system by generalizing it for object-based processing.

## Appendix I
## Point-Wise IDWT

In what follows, we provide a detailed description of the procedure followed to determine the set of wavelet coefficients needed for PW-IDWT. We call it generalized projection (GP) of the considered sample into the transformed domain.



(a) (b)

Fig. 19. Notations for the left end of GP during its evaluation: $\hat{b}_a^{i+1,-}$ is for the upper branch, and $\hat{b}_d^{i+1,-}$ for the lower branch, $i$ being the index of the considered step. (a): $t$-step; (b) $s$-step.

Let GP$(l,j)$ be the GP of $s(k)$ in subband $(l,j)$. For generality, we state the problem as follows: given a wavelet sample $a(l,k)$ at position $k$ in subband $(l,a)$ (i.e., the approximation subband at level $l$), we want to determine its GP at the next coarser level GP$(l+1,j)_k, j = a,d$. As GP$(l,j)_k$ depends on $k$ being even or odd, we introduce an additional index $\mu$ to make the distinction. The set of lifting steps involved in the synthesis of an even sample is indeed different from the set of those necessary for an odd sample. Then, GP$(l+1,j)_k^{\mu}, \mu = a,d$ indicates the GP in subband $(l+1,j)$ of a sample in subband $(l,j)$ of even ($\mu = a$) or odd ($\mu = b$) position, respectively, as illustrated in Fig. 18. In the figure, the first filtering step of the lifting chain is assumed to be of type $t$, as usually is. This corresponds to setting $s_1(z) = 0$ in (3). We call $a'(l,k')$ ($d'(l,k')$) the sample to recover at the end of the synthesis chain for $k$ even (odd). The GP is determined by identifying the positions of its ends, $b_j^{\mu,-}$ and $b_j^{\mu,+}$, respectively. Here, we subtend the indexes $l$ and $k$ for simplicity of notations. Let then $l_f^-$ ($l_{fi}^+$) be the causal (anti-causal) length of the $i^{th}$ lifting filter, where $f = s,t$ identifies the filter type. Finally, let $(\hat{b}_{f,a}^{i+1,-}, \hat{b}_{f,a}^{i+1,+})$ identify the border samples *before* the $i^{th}$ step $f$ in the synthesis chain, and

$(\hat{b}^{i,-}_{f,a}, \hat{b}^{i,+}_{f,a})$ the same quantity *after* $f$, as shown in Fig. 19 for the case $f = t$. The following updating rule holds [see Fig. 19(a) and (b)]:

$$\hat{b}^{i+1,-}_{a} = \hat{b}^{i,-}_{a} \tag{17}$$
$$\hat{b}^{i+1,+}_{a} = \hat{b}^{i,+}_{a} \tag{18}$$
$$\hat{b}^{i+1,-}_{d} = \min\{\hat{b}^{i,-}_{d}, \hat{b}^{i,-}_{a} + l^{-}_{s}\} \tag{19}$$
$$\hat{b}^{i+1,+}_{d} = \max\{\hat{b}^{i,+}_{d}, \hat{b}^{i,+}_{a} + l^{+}_{s}\}. \tag{20}$$

(As these rules refer to a generic $i^{th}$ step, the the $\mu$ index has been omitted). If one prediction step $t_i$ is considered, the updating rules become

$$\hat{b}^{i+1,-}_{a} = \min\{\hat{b}^{i,-}_{a}, \hat{b}^{i,-}_{d} + l^{-}_{t}\} \tag{21}$$
$$\hat{b}^{i+1,+}_{a} = \max\{\hat{b}^{i,+}_{a}, \hat{b}^{i,+}_{d} + l^{+}_{t}\} \tag{22}$$
$$\hat{b}^{i+1,-}_{d} = \hat{b}^{i,-}_{d} \tag{23}$$
$$\hat{b}^{i+1,+}_{d} = \hat{b}^{i,+}_{d}. \tag{24}$$

If the total number of lifting steps is $m$, then the ends of $\mathrm{GP}(l,j)^{\mu}_{k}$ are

$$\begin{aligned} b^{a,-}_{j} &= \hat{b}^{m,-}_{j} \\ b^{a,+}_{j} &= \hat{b}^{m,+}_{j} \\ b^{d,-}_{j} &= \hat{b}^{m+1,-}_{j} \\ b^{d,+}_{j} &= \hat{b}^{m+1,+}_{j}. \end{aligned} \tag{25}$$

Given $\mathrm{GP}(l+1,j)^{\mu}_{k}$, namely the GP of the sample $a(l,k)$, the GP of the set of samples $\{a(l,k), k \in \mathrm{GP}(l,j)^{a}\}$ can be obtained as

$$\mathrm{GP}^{\mu}(l+1,j) = \cup_{k}\mathrm{GP}^{\mu}(l+1,j)_{k} \ \forall k \in \mathrm{GP}(l,j)^{a} \tag{26}$$

where $\mathrm{GP}(l,j)^{a}$ is the GP of the signal sample $s(k)$ in the approximation subband of level $l$, which is the set where $a(l,k)$ lives. The GP of $s(k)$ is thus determined iteratively, climbing back the synthesis chain.

## APPENDIX II
## BITSTREAM OVERLOADING

There are two factors contributing to the bitstream overloading affecting the LPL-based modalities, both strictly related to the architecture of the entropy coder. The arithmetic coder we have adopted uses the *byte* as information unit for writing to or reading from the bitstream file. The string of bits corresponding to a sequence of symbols is temporarily stored into a buffer, whose most significant byte is written into the bitstream file once it has been completely filled. The one-byte length string can thus be considered as the elementary bitstream unit. In what follows, we will call this elementary block *coding unit*. Each coding unit thus represents a *set* of symbols, which cannot be accessed (i.e., decoded) independently. This implies that in order to independently encode/decode two sets of symbols it is necessary to follow a special procedure to generate two disjoint segments within the same bitstream which can then be accessed independently. This consists essentially in emptying the buffer and resetting all the internal variables of the entropy coder. We define the sequence of bytes consequently written in the bitstream file as *flush bytes*. This is not yet enough for partitioning the bitstream into semantically disjoint sets. Some special symbols or *markers*, must be introduced, which can be unambiguously interpreted as *separators* by the decoder. Since the whole set of values 0-255 attainable by the coding unit is used by the arithmetic coder, no special characters were *a priori* available to be used as markers. In order to overcome this problem, a special role has been assigned to one character of the set, together with an additional syntactic rule. Let $S$ be the chosen symbol. The rule can be stated as follows: the symbol $S$ is replaced by a sequence of two symbols depending on its semantic. The additional symbol would be $S_1$ in normal working condition, and $S_2$ when used to build the marker. In the specific: $S = 255$, $S_1 = 0$, and $S_2 = 255$.

The need of markers affects the coding system performance both directly, as additional information to be written into the bitstream, and indirectly, degrading the efficiency of the entropy coder by increasing the number of information units associated to symbol $S$. As the number of separators is a function of the volume size and the dynamic range of the transformed coefficients, it can be evaluated after the transformation has been performed. Conversely, neither the number of special $S$ symbols nor the number of flush bytes can be calculated *a priori*, because they depend on the statistics of the symbols to be encoded. We refer to [21] for more details.

## REFERENCES

[1] Y. Kim and W. A. Pearlman, "Lossless volumetric medical image compression," in *Proc. SPIE Applications of Digital Image Processing XXII*, vol. 3808, 1999, pp. 305–312.

[2] A. Said and W. A. Pearlman, "A new, fast, and efficient image coded based on set partitioning hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, June 1996.

[3] Z. Xiong, X. Wu, D. Y. Yun, and W. A. Pearlman, "Progressive coding of medical volumetric data using three-dimensional integer wavelet packet transform," in *Proc. 2nd IEEE Workshop Multimedia Signal Processing*, 1998, pp. 553–558.

[4] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, pp. 3445–3462, Dec 1993.

[5] A. Bilgin, G. Zweig, and M. V. Marcellin, "Three-dimensional image compression with integer wavelet transform," *Appl. Opt.*, vol. 39, no. 11, pp. 1799–1814, 2000.

[6] G. Menegaz, V. Vaerman, and J.-P. Thiran, "Object-based coding of volumetric medical data," in *Proc. Int. Conf. Image Processing (ICIP)*, vol. 3, 1999, pp. 920–924.

[7] G. Menegaz and J.-P. Thiran, "Lossy to lossless object-based coding of 3-D MRI data," *IEEE Trans. Image Processing*, vol. 11, pp. 1053–1061, Sept. 2002.

[8] K. K. Chan, C. C. Lau, K. S. Chuang, and C. A. Morioca, "Visualization and volumetric compression," in *Proc. SPIE*, vol. 1444, 1991, pp. 250–255.

[9] J. Wang and H. K. Huang, "Three-dimensional medical image compression using a wavelet transform with parallel computing," in *Proc. SPIE*, vol. 2431, Medical Imaging, 1995, pp. 162–172.

[10] A. Baskuet, H. Benoit-Cattin, and C. Odet, "On a 3-D medical image coding method using a separable 3-D wavelet transform," in *Proc. SPIE*, vol. 2431, Medical Imaging, 1995, pp. 173–183.

[11] J. Wei, P. Saipetch, and R. Panwar, "Volumetric image compression by 3-D discrete wavelet transform," in *SPIE*, vol. 2431, Medical Imaging, 1995, pp. 184–194.

[12] J. Wang and H. K. Huang, "Medical image compression by using three-dimensional wavelet transform," *IEEE Trans. Med. Imag.*, vol. 15, pp. 547–554, Aug. 1996.

[13] D. Taubman and A. Zakhor, "Multirate 3-d subband coding of video," *IEEE Trans. Image Processing*, vol. 3, pp. 572–588, Sept 1994.

[14] G. A. Triantafyllidis and M. G. Strinzis, "A context based adaptive arithmetic coding technique for lossless image compression," *IEEE Signal Processing Lett.*, vol. 6, pp. 168–170, July 1999.

[15] *Information Technology – JPEG2000 Image Coding System* , ISO/IEC Int. Standard 15 444–1, 2000.

[16] I. Daubechies and W. Sweldens, “Factoring wavelet transform into lifting steps,” *J. Fourier Anal. Appl.*, vol. 41, no. 3, pp. 247–269, 1998.

[17] A. R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, “Wavelet transforms that map integers to integers,” *Appl. Comput. Harmon. Anal.*, vol. 5, no. 3, pp. 332–369, 1998.

[18] J. Reichel, G. Menegaz, M. Nadenau, and M. Kunt, “Integer wavelet transform for embedded lossy to lossless image compression,” *IEEE Trans. Image Processing*, vol. 10, pp. 383–392, Mar. 2001.

[19] S. G. Mallat, “A theory for multiresolution signal decomposition: The wavelet representation,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 2, pp. 674–693, July 1989.

[20] J. Reichel, G. Menegaz, and M. Nadenau, “Integer wavelet decomposition for lossy image compression,” in *Proc. SPIE*, vol. 3808, Denver, CO, 1999, pp. 257–268.

[21] G. Menegaz, “Model-based coding of multi-dimensional data with applications to medical imaging,” Ph.D. dissertation, Signal Processing Lab. (LTS), Swiss Federal Inst. Technol. (EPFL), Lausanne, Switzerland, 2000.

[22] A. R. Calderbank, I. Daubechies, W. Sweldens, and B. L. Yeo, “Lossless image compression using integer to integer wavelet transforms,” in *Proc. Int. Conf. Image Processing (ICIP)*, 1997, pp. 596–599.

[23] I. Daubechies and J. C. Fauveau, “Biorthogonal bases of compactly supported wavelets,” *Commun. Pure. Appl. Math.*, vol. 45, pp. 485–560, 1992.

[24] W. Pennebacker, J. Mitchell, G. Langdon, and R. Arps, “An overview of the basic principles of the q-coder adaptive binary arithmetic coder,” *IBM J. Res. Manage.*, vol. 32, no. 3, pp. 717–726, Nov 1988.

[25] Y. Kim and W. A. Pearlman, “Stripe-based sphit lossy compression of volumetric medical images for low-memory usage and uniform reconstruction quality,” in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, 2000, pp. 2031–2034.

[26] E. J. Candes *et al.*, “Ridgelets and their derivatives: Representation of images with edges,” in *Curves and Surfaces*, Schumaker *et al.*, Eds. Nashville, TN: Vanderbilt Univ. Press, 1999.

[27] E. J. Candes *et al.*, “Curvelets – A surprisingly effective nonadaptive representation for objects with edges,” in *Curves and Surfaces*, L. L. Schumaker *et al.*, Eds. Nashville, TN: Vanderbilt Univ. Press, 1999.

[28] M. Rabbani and R. Joshi, “An overview of the jpeg2000still image compression standard,” *Signal Processing*, 2002.

[29] J. L. Mitchell and W. B. Pennebaker, *JPEG Still Image Compression Standard*. New York: Van Nostrand, 1993.