Lossy-to-Lossless 3D Image Coding through Prior Coefficient Lookup Tables

Francesc Aulí-Llinàs, Michael W. Marcellin, Joan Serra-Sagristà, and Joan Bartrina-Rapesta

Abstract—This paper describes a low-complexity, high-efficiency, lossy-to-lossless 3D image coding system. The proposed system is based on a novel probability model for the symbols that are emitted by bitplane coding engines. This probability model uses partially reconstructed coefficients from previous components together with a mathematical framework that captures the statistical behavior of the image. An important aspect of this mathematical framework is its generality, which makes the proposed scheme suitable for different types of 3D images. The main advantages of the proposed scheme are competitive coding performance, low computational load, very low memory requirements, straightforward implementation, and simple adaptation to most sensors.

Index Terms—3D image coding, entropy coding, bitplane image coding, JPEG2000.

I. INTRODUCTION

Specialized devices that capture images with many spatial, spectral, or temporal components are prevalent in several fields. Any image with more than one component is referred to in this paper as a three-dimensional (3D) image. Treating a 3D image as a volume, let samples of the image be referred to as $W_{z,y,x}$, with z, y, x denoting the position of the sample in the depth, vertical, and horizontal coordinate axes of the volume, respectively. One component is defined as those samples situated at the same depth z' or, more precisely, component z' is defined as all samples $W_{z,y,x}$ such that $z = z'$. By convention, one component comprises the vertical and horizontal dimensions of space, respectively, in the y and x coordinates, whereas the depth axis provides a spatial, spectral, or temporal dimension depending on the capturing device.

Lossy/lossy-to-lossless compression of 3D images is a topic of interest in several communities. In the medical field, many efforts have explored the use of 3D wavelet transforms to decorrelate redundancy among components [1]–[3]. Other proposals have adapted popular bitplane coding systems to three dimensions [4]–[6]. Rapid access and remote volume visualization was studied in [7]–[9]. More recently, enhanced coding efficiency has been achieved by means of exploiting the symmetry of the human body [10], or using the Karhunen-Loève transform [11]. Among currently deployed standards,

JPEG2000 [12] attracts the most attention due to its advanced features and due to its inclusion in DICOM (Digital Imaging and Communications in Medicine). Hence, compression of 3D medical images with JPEG2000 has been extensively explored [5], [9], [13], [14].

In the remote sensing field, several bitplane coding schemes have been adapted to multi-dimensional data [15], [16]. The Karhunen-Loève transform has proven to be especially suitable to decorrelate spectral information [17]–[19]. Meanwhile, efficient approaches to exploit the intrinsic nature of remote sensing images have been proposed, including pixel classification [20] and models of anomalous pixels [21]. Again, JPEG2000 is widely known in the community due to the provision of advanced features such as multi-component transforms and effective interactive transmission protocols. Work that uses JPEG2000 to code 3D remote sensing images includes [17], [22]–[24]. Additionally, the recently published JPEG2000 Part 10 [25] provides specialized features aimed at the coding of volumetric data sets.

In the case of video coding, the temporal redundancy among frames of a video sequence is typically removed via motion compensation mechanisms. H.264/AVC [26] is currently the most advanced standard employing such mechanisms. Inter-frame redundancy can also be reduced by means of other mechanisms such as (motion-adaptive) transforms [14], or conditional replenishment [27]. JPEG2000 provides an excellent framework to explore these latter cases. Furthermore, for video sequences with little motion, such as those produced in videoconference or surveillance applications, JPEG2000 also provides an ideal framework [28].

Compression and/or display of 3D images often needs to be carried out using devices with limited resources. Therefore, coding systems should be devised while keeping in mind the device in which they will be executed [29], [30]. There are many works concerned with the computational complexity of techniques and algorithms [31], [32] deployed to code 3D images [11], [14], [19], [20], [22], [33]–[36].

In most works, the study of 3D images is carried out for one particular type of 3D image, or for one particular type of sensor, due to the statistically different nature of these images. The purpose of this paper is to introduce a low-complexity lossy/lossy-to-lossless coding strategy based on bitplane coding for the compression of different types of 3D images. The proposed strategy is implemented in the JPEG2000 framework due to its suitability for coding 3D images and its widespread use. The resulting system has very low memory requirements (only two components need to

Francesc Aulí-Llinàs, Joan Serra-Sagristà, and Joan Bartrina-Rapesta are with the Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, Spain. Michael W. Marcellin is with the Department of Electrical and Computer Engineering, University of Arizona, USA. Corresponding author is Francesc Aulí-Llinàs (phone: +34 935813571; fax: +34 935814477; e-mail: fauli@deic.uab.es).

be maintained in memory), lower computational complexity than JPEG2000 (with about 2/3 of the computational costs), and provides competitive coding performance. The main idea behind the proposed method is to *not* use any transform along the depth axis. Instead, we use an effective probability model for the bits emitted by the bitplane coding engine. This probability model considers solely the magnitude and sign of the coefficient at the same spatial location in the previous component. An important contribution of this research is the generality of the proposed approach. Specifically, it provides a common mathematical framework to model and capture the statistical nature of different types of signals.

The remainder of the paper is structured as follows. Section II briefly reviews common probability models employed in lossy/lossy-to-lossless image coding systems, and introduces the theoretical underpinnings of our model. Section III describes practical considerations for implementation in the JPEG2000 framework, and discusses scalability and complexity issues. Coding performance, computational complexity, and memory requirements are assessed in Section IV. Medical images, remote sensing images, and video are considered. The final section summarizes the work and gives concluding remarks.

II. PROBABILITY MODEL BASED ON PRIOR COEFFICIENT LOOKUP TABLES

A. Review of classic probability models

Let $\mathcal{W}_{z,y,x}$ be a wavelet coefficient from a 3D image. Let υ be the magnitude of the index obtained by quantizing $\mathcal{W}_{z,y,x}$. Let $[b_{M-1}, b_{M-2}, ..., b_1, b_0], b_i \in \{0,1\}$, be the binary representation of υ, with M denoting a sufficient number of bits to represent all coefficients. Finally, let $d \in \{+, -\}$ be the sign of $\mathcal{W}_{z,y,x}$. Bitplane coding strategies generally define bitplane j as the collection of bits b_j from all coefficients, and encode the image from the most significant bitplane $M-1$ to the least significant bitplane 0. The first non-zero bit of a coefficient, i.e., that $b_s = 1$ such that $\nexists\ s' > s$ with $b_{s'} = 1$, is called the significance bit of the coefficient. The sign of the coefficient is coded immediately after its significance bit. The remaining bits $b_r, r < s$, are called refinement bits. The significance state $\Phi(\cdot)$ of coefficient $\mathcal{W}_{z,y,x}$ in bitplane j is defined as

$$\Phi(\mathcal{W}_{z,y,x}, j) = \begin{cases} 0 & \text{if } j > s \\ 1 & \text{otherwise} \end{cases}. \tag{1}$$

Commonly, bits emitted by a bitplane coding engine are fed to an entropy coder able to exploit high-order statistics of symbols. The most popular approach to exploit such statistical redundancy is context-adaptive arithmetic coding. The main idea is to adaptively adjust the probabilities of emitted bits depending on the context of the coefficient. In more detail, let $\mathcal{W}_{z,y,x}^n, 1 \leq n \leq N$, denote N neighbors of $\mathcal{W}_{z,y,x}$. In general, contexts are selected as some function of these neighbors. Often, this function employs only $\{\Phi(\mathcal{W}_{z,y,x}^n, j)\}$ and considers the number and the position of neighbors that

are significant in the current or previous bitplanes. The context is passed to the arithmetic coder, which (if adaptive) adjusts probabilities as more symbols are coded. The probability mass function of the currently emitted symbol is denoted as $P_{sig}(b_j), j \geq s$, for significance coding, as $P_{ref}(b_j), j < s$, for refinement coding, and as $P_{sign}(d)$ for sign coding.

Key to the compression efficiency is the context formation approach. The approach used in JPEG2000, for example, employs a heuristic based on the image features captured in each wavelet subband [37, Ch. 8.3.2]. Context-adaptive arithmetic coding is a technology mostly employed for lossy/lossy-to-lossless regimes. Lossless (only) compression, on the other hand, commonly employs predictive techniques to estimate the magnitude of the current sample [38], [39]. For the case of 3D images, context formation approaches are studied in [14], [40] for lossy regimes, and predictive techniques that consider neighbors in previously coded components are studied in [41], [42] for lossless regimes.

B. Proposed model

Lossy-to-lossless compression schemes have not commonly been able to exploit the high correlation among components of a 3D image via probability models. It is conjectured that the significance state $\Phi(\cdot)$ of a 3D neighborhood is *not* an adequate indicator of the underlying signal [14]. The main insight of our research is a probability model that captures the statistical behavior of a 3D image by employing the magnitude –rather than the significance state– of partially reconstructed coefficients in previous components. To do so, the probability of the symbols emitted for $\mathcal{W}_{z,y,x}$ is determined *solely* with the magnitude and sign of the *prior coefficient* at the same location, i.e., $\mathcal{W}_{z-1,y,x}$. Specifically, no spatial neighbors are employed. Additionally, no transform or prediction is used along the depth axis. If the magnitude and sign of $\mathcal{W}_{z-1,y,x}$ are respectively denoted as φ and ϕ, the encoder uses only $P_{sig}(b_j \mid \varphi)$, $P_{ref}(b_j \mid \varphi)$, and $P_{sign}(d \mid \phi)$ in the encoding of $\mathcal{W}_{z,y,x}$.

Our model assumes that probabilities of emitted symbols can be determined by considering only the probability density function (pdf) for the coefficient to be coded *and* the prior coefficient at the same location. More precisely, let $p(w)$ denote the marginal pdf for coefficient w. We assume that all coefficients within the same wavelet subband of all components of an image are identically distributed. Let $g(\varphi \mid w)$ denote the conditional pdf for the prior coefficient given w. $p(w)$ and $g(\varphi \mid w)$ consider only the density of the magnitude of w and φ. The sign is treated separately (see below).

Table I describes test images having 5 different sensor types belonging to the remote sensing, medical, and video-conferencing fields. To illustrate the nature of the different 3D image types, Figure 1(a) depicts $p(w)$ for one image of each type. The data used in Figure 1 correspond to the high vertical-, low horizontal-frequency subband (HL) of the first decomposition level produced by the irreversible 2D CDF 9/7 wavelet transform. This transform is applied to each component of the image. Again, no transform is used along the depth axis of the volume. Figures 1(b)-(f) depict $g(\varphi \mid w)$ for

TABLE I: Characteristics of the images employed in experiments. (Z, Y, X) denotes the number of samples on the corresponding coordinate axes.

field	type	size (Z, Y, X)	dimensions	bit depth	image names
remote sensing	AVIRIS	224, 512, 512	1D spectral + 2D spatial	16 bps	cuprite, jasper sc01, jasper sc02, lowAltitude sc01, lowAltitude sc02, lunarLake
remote sensing	Hyperion	242, 768, 256	1D spectral + 2D spatial	12 bps	agricultural, urban, flooding, forestry, coastal, tornado
medical	Computed Tomography	112, 512, 512	3D spatial	12 bps	A, B, C, D, E, F
medical	Angiography (X-RAY)	151, 512, 512	1D temporal + 2D spatial	12 bps	A, B, C, D, E, F
videoconferencing	CIF	449, 288, 352	1D temporal + 2D spatial	8 bps	bridge, paris, salesman, closed bridge, news, waterfall

the wavelet subband described above. Despite the differences among these 3D image types, the statistical behavior is similar for all images of the same type (not shown in the figures).

For each image type, the joint pdf $h(w, \varphi) = p(w) \cdot g(\varphi \mid w)$ is used below to determine $P_{sig}(b_j \mid \varphi)$ and $P_{ref}(b_j \mid \varphi)$. Other work using the joint pdf $h(w, \varphi)$ also indicates that $h(w, \varphi)$ may be a suitable indicator of the signal's nature [43], [44]. Probabilities for significance coding at bitplane j are determined as the probability of insignificant coefficients coded at bitplane j divided by all coefficients coded in that bitplane as

$$P_{sig}(b_j = 0 \mid \varphi) = P(w < 2^j \mid w < 2^{j+1} , \varphi) =$$

$$\frac{P(w < 2^j , w < 2^{j+1} \mid \varphi)}{P(w < 2^{j+1} \mid \varphi)} =$$

$$\frac{P(w < 2^j \mid \varphi)}{P(w < 2^{j+1} \mid \varphi)} = \frac{\int_0^{2^j} p(w) \cdot g(\varphi \mid w) \, dw}{\int_0^{2^{j+1}} p(w) \cdot g(\varphi \mid w) \, dw} . \quad (2)$$

The conditioning event $w < 2^{j+1}$ in the first line of (2) arises by noting that to become significant in bitplane j, the coefficient must be insignificant in bitplane $j + 1$. Both dividend and divisor in the last line of (2) consider the density of coefficient w within the subband *and* the density of φ for a known value of w.

Probabilities for the *first* refinement bit of coefficients that become significant at bitplane j are determined according to

$$P_{ref}(b_{j-1} = 0 \mid \varphi) =$$

$$P(2^j \le w < 2^j + 2^{j-1} \mid 2^j \le w < 2^{j+1} , \varphi) =$$

$$\frac{P(2^j \le w < 2^j + 2^{j-1} , 2^j \le w < 2^{j+1} \mid \varphi)}{P(2^j \le w < 2^{j+1} \mid \varphi)} = \quad (3)$$

$$\frac{\int_{2^j}^{2^j + 2^{j-1}} p(w) \cdot g(\varphi \mid w) \, dw}{\int_{2^j}^{2^{j+1}} p(w) \cdot g(\varphi \mid w) \, dw} .$$

Extension to other refinement bitplanes entails the consideration of intervals with refinement bits equal to 0 in the dividend

of the last equation of (3).

Figures 2(a) and 2(b) respectively depict $P_{sig}(b_j = 0 \mid \varphi)$ and $P_{ref}(b_j = 0 \mid \varphi)$ for the coding of one bitplane of the same wavelet subband reported previously. In these graphics, the horizontal axis depicts the magnitude of the prior coefficient φ, and the vertical axis depicts probabilities of emitted symbols. These graphics report probabilities for different representative bitplanes depending on the image bit-depth. It is worth noting the differences among image types.

As an example, let us elaborate the case for the "cuprite" image. Figure 2(a) depicts probabilities for significance coding for bitplane $j = 7$. When φ is in the interval $\varphi \in [0, 2^7)$, coded coefficients have low probability to become significant (i.e., $P_{sig}(b_7 = 0 \mid \varphi) \ge 0.5$), whilst when $\varphi \in [2^7, \infty)$ the probability to become significant is higher (i.e., $P_{sig}(b_7 = 0 \mid \varphi) < 0.5$). Extended to other bitplanes, this observation seems to indicate that (probabilistically) coefficient $\mathcal{W}_{z,y,x}$ has a magnitude similar to $\mathcal{W}_{z-1,y,x}$. Similar evidence is found for the angiography image and the video sequence, but not for the Hyperion and Computed Tomography images, which have different statistical behavior.

For the "cuprite" image, Figure 2(b) indicates that the probability of the first refinement bit $j' = j - 1$ of coefficients that became significant at bitplane $j = 7$ vary widely depending upon φ. Analyzed in more detail, probabilities oscillate, crossing above and below 0.5, in intervals of length $2^{j'}$, or $2^6 = 64$. More formally, $P_{ref}(b_6 = 0 \mid \varphi) \ge 0.5$ for $\varphi \in \{[0, 192), [256, 320), [384, 448)\}$ and $P_{ref}(b_6 = 0 \mid \varphi) < 0.5$ for $\varphi \in \{[192, 256), [320, 384), [448, 512)\}$. This observation indicates that there exists correlation between refinement bits of the current coefficient and the previous coefficient that became significant at $j = 7$ and also at $j = 8$. We remark that this strong correlation is not due to the calibration process carried out in these AVIRIS images [45] since similar behavior is achieved with uncalibrated AVIRIS images. Again, the statistical behavior and probabilities vary depending on the image type.

The model employed for sign coding is as follows. Our main assumptions are that the sign of the current coefficient (i.e., d) is very likely to be the same as that of the prior coefficient (i.e., ϕ), and that these probabilities vary depending on the significant bitplane of coefficients. Figure 2(c) depicts $P_{sign}(d = \phi \mid \phi)$ in different bitplanes for the same images evaluated above. The dependence between d and ϕ is, in gen-
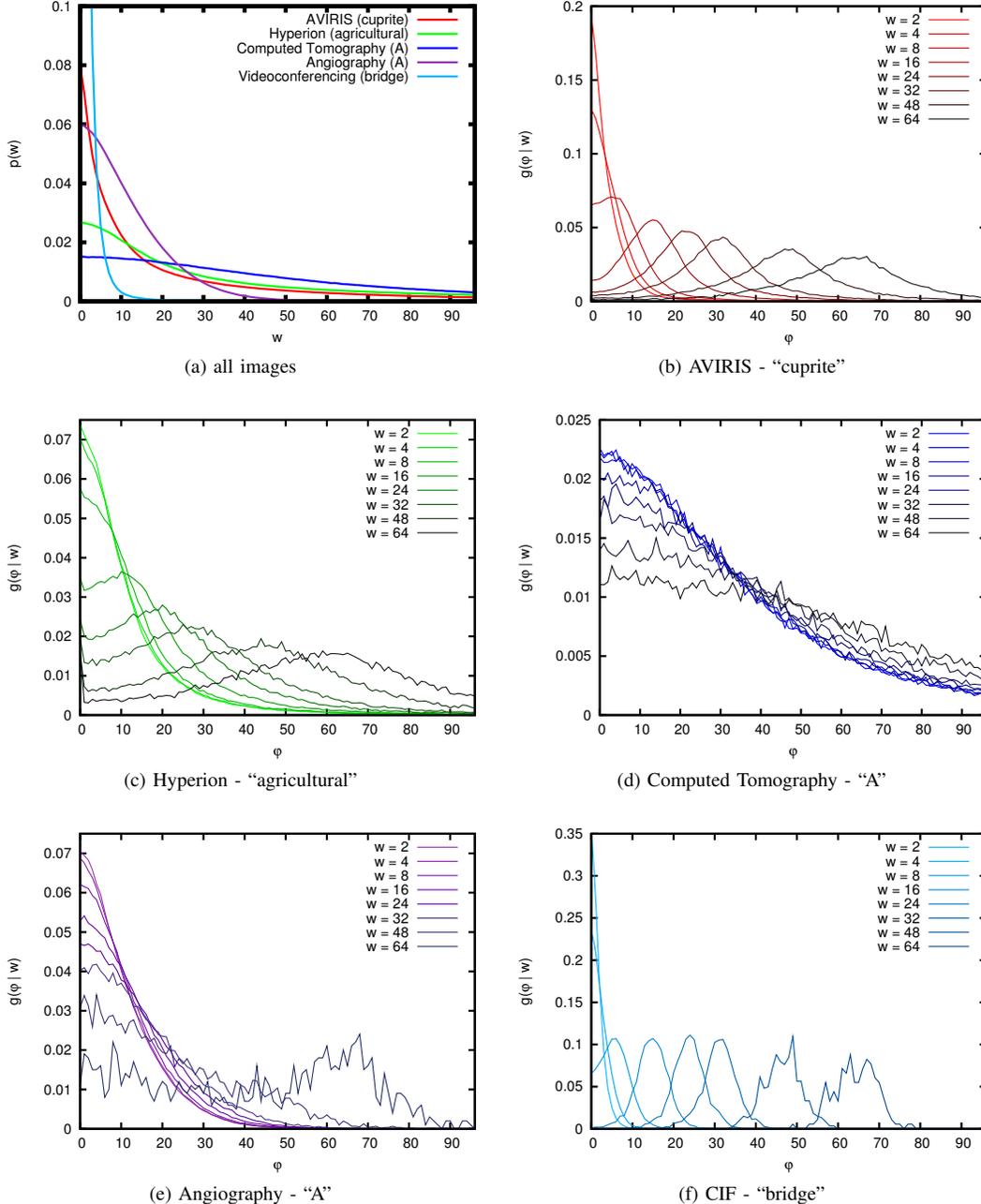
Fig. 1: Statistical analysis of different 3D image types. (a) reports $p(w)$ for all images and (b)-(f) reports $g(\varphi \mid w)$ separately for each image.

eral, strong at high bitplanes and weak at low bitplanes. This is intuitively explained by considering that the difference (as the *non*-absolute magnitude) between coefficients with opposite signs is smaller for coefficients that become significant at low bitplanes than for coefficients that become significant at high bitplanes. Though probabilities are different depending on the image type, in the case of signs, the statistical behavior is more similar for all image types than that found for significance and refinement coding.

III. PRIOR COEFFICIENT LOOKUP TABLES CODING SCHEME

A. Practical considerations

The prior coefficient-based probability model can be implemented with very low complexity as follows. First, pdfs are extracted for wavelet subbands of all components for *only one* image of the sensor in question. This procedure generates one $p(w)$ and one $g(\varphi \mid w)$ per wavelet subband, irrespective of the number of components. For each subband, one lookup table (LUT) containing $P_{sig}(b_j = 0 \mid \varphi)$ is generated for significance coding, and one LUT containing $P_{ref}(b_j = 0 \mid \varphi)$ is generated for refinement coding, via

(a) Significance coding

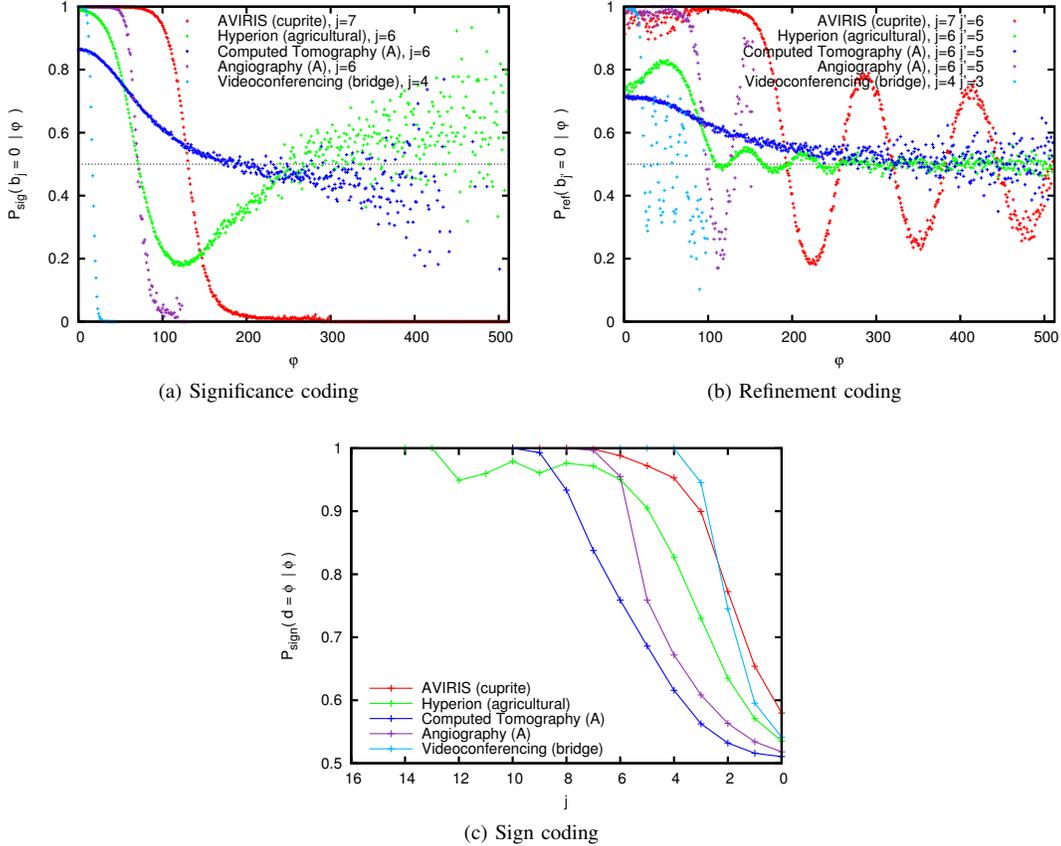(b) Refinement coding

(c) Sign coding

Fig. 2: Probabilities of symbols emitted by a bitplane coding engine for different types of images: (a) reports $P_{sig}(b_j = 0 \mid \varphi)$, where j is the coded bitplane; (b) reports $P_{ref}(b_{j'} = 0 \mid \varphi)$, where j is the significant bitplane and j' is the first refinement bitplane; (c) reports $P_{sign}(d = \phi \mid \phi)$ for different significant bitplanes j.

Equations (2), (3) and the extracted pdfs.[1] Let us denote the LUT for significance coding as $\mathcal{L}_{sig,u}$, with u standing for the wavelet subband to which coefficients belong. $\mathcal{L}_{sig,u}$ contains M_u rows, which refers to the number of bitplanes needed to represent all coefficients in that subband. For each row, there are 2^{M_u} columns representing all possible values of φ. Cells of the LUT contain pre-computed probabilities, so that $P_{sig}(b_j = 0 \mid \varphi)$ is accessed as $\mathcal{L}_{sig,u}[j][\varphi]$. The LUT for refinement coding has the same structure with an extra dimension that accounts for the bitplane at which the refined coefficient became significant. The refinement LUT is referred to as $\mathcal{L}_{ref,u}$, and is accessed as $\mathcal{L}_{ref,u}[j][j'][\varphi]$, with j denoting the bitplane at which the coefficient became significant, and j' denoting the current refinement bitplane. For sign coding the procedure is similar. The LUT containing $P_{sign}(d = \phi \mid \phi)$ is generated per wavelet subband. This LUT is denoted as $\mathcal{L}_{sign,u}$, and contains M_u rows and one column. $\mathcal{L}_{sign,u}[j]$ contains $P_{sign}(d = \phi \mid \phi)$ for bitplane j.

LUTs are computed offline and are assumed to be known by coder and decoder, without need to explicitly transmit them. In the experiments described in the next section, we use (all

components) of one image from a given sensor to populate the LUTs for that sensor. Results are then generated by applying the LUTs to other images from that sensor. We note that the generation of the LUTs requires only *one* image, which leads to a rapid training stage. We call this method prior coefficient lookup tables (PCLUT).

B. Implementation in JPEG2000 framework

We now describe the implementation of PCLUT in the JPEG2000 framework. A typical coding system for JPEG2000 [12] is constituted by three main stages: sample data transformation, sample data coding, and codestream re-organization. The sample data transformation stage compacts image energy and prepares samples for the next stage. The main operation carried out in this first stage is the application of the wavelet transform to decorrelate spatial redundancy of components. Two filter-banks are supported for 2D decor-relation in JPEG2000 Part 1 [12]: the CDF 9/7 irreversible wavelet transform for lossy compression, and the CDF 5/3 reversible wavelet transform for lossy-to-lossless compression. JPEG2000 Part 2 [46] supports a wider variety of transforms that can be applied spatially and/or in the depth axis, which are especially suitable for the coding of 3D images. After the first stage, the image is logically partitioned in small 2D sets of

---

[1]In practice, the probability values of the LUTs can be estimated using relative frequencies conditioned on φ, avoiding the need for numerical integration.
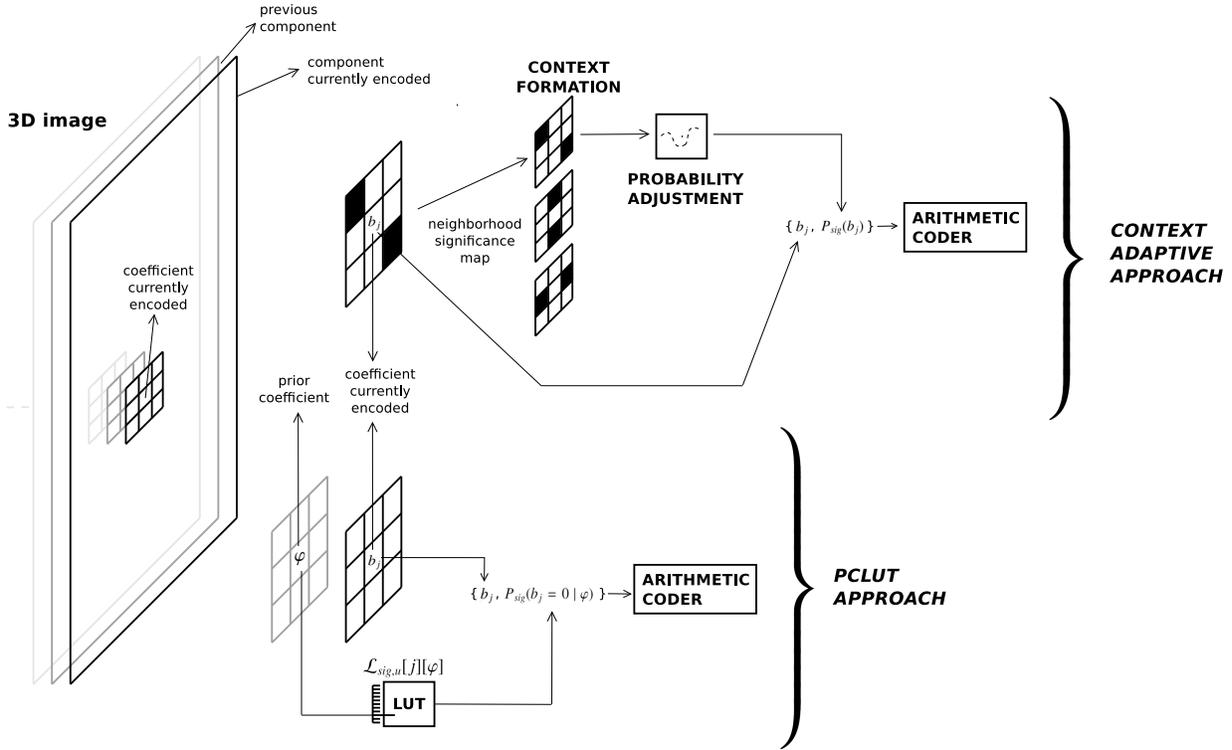
Fig. 3: Illustrative representation of the steps carried out by a context-adaptive approach and PCLUT when determining probabilities of emitted symbols. The currently encoded symbol is denoted as b_j and the probability determined for this symbol is denoted as $P(b_j)$.

wavelet coefficients, called codeblocks that are independently coded by the sample data coding stage.

The purpose of the sample data coding stage, also called tier-1, is to produce an embedded codestream for each codeblock. Tier-1 carries out three sub-bitplane coding passes at each bitplane called Significance Propagation Pass (SPP), Magnitude Refinement Pass (MRP), and Cleanup Pass (CP). Each coefficient is scanned only once in each bitplane. SPP and CP are devised for significance coding, whereas MRP is devised for refinement coding. Probabilities of emitted symbols are determined using the context-adaptive mechanisms of JPEG2000.

Our model is integrated in tier-1 by feeding the arithmetic coder with the symbol and its probability determined through the pre-computed LUTs. Figure 3 depicts the steps commonly required by a traditional context-adaptive approach and by PCLUT. The context-adaptive approach codes the symbol b_j employing two steps. The first step, which is named "context formation" in the figure, computes the context of the current coefficient using the significance map of its neighbors. This is illustrated in the figure as the black (significant) and white (insignificant) neighbors of the coefficient currently encoded. The context of the coefficient is then mapped to one context from the set defined by JPEG2000. Each context employs a different probability, which is adaptively adjusted in the second step of this approach as more data are coded. Both the probability and the symbol are fed to the arithmetic coder. This process requires access to eight neighboring coefficients and computational resources to calculate the context and the

probability. As seen in the figure, PCLUT requires access to only one previous coefficient and one access to the LUT to determine the probability of the emitted symbol. We remark that this operation is repeated intensively during the coding process. Specifically, it is carried out once for each coefficient in each bitplane. Thus, simplification of this process is essential to reduce computational costs.

The third stage of the JPEG2000 coding system is codestream re-organization, also known as tier-2, which codes auxiliary data and organizes the final codestream into quality layers. Typically, the minimization of image distortion for a given target bitrate is conducted by a rate-distortion optimization process.

C. Scalability considerations

An important feature provided by JPEG2000 is scalability in terms of spatial location, resolution, component, and quality [37]. Spatial location scalability and resolution scalability are not compromised by our probability model, but mechanisms for component scalability and quality scalability are affected. Component scalability is the ability to access and decode selected components of the image without needing to decode the full codestream.

Our model determines probabilities for $\mathcal{W}_{z,y,x}$ using $\mathcal{W}_{z-1,y,x}$, which produces a causal effect that encompasses the full 3D volume. For instance, if component z'' needs to be recovered, all previous components $z''' < z''$ also need to be decoded. Obviously, the first component of the

TABLE II: Component scalability for different wavelet filter-banks and the proposed coding scheme. The table reports the average number of components decoded when only one component is selected for extraction. Results are approximate (image boundaries may reduce dependencies). l denotes the number of wavelet decomposition levels, whereas C is the size of clusters employed by PCLUT.

	general	$l=1$	$l=3$	$l=5$
Haar wavelet	$l+1$	2	4	6
CDF 9/7 irrev.	$7l+1$	8	22	36
CDF 5/3 rev.	$3l+1$	4	10	16
	general	$C=2$	$C=8$	$C=32$
PCLUT (ave., max)	$(C/2, C)$	$(1,2)$	$(4,8)$	$(16,32)$

full volume is *not* coded using the prior coefficient-based approach since there is no prior coefficient. In the experimental results reported in the next section, the first component is coded using the normal JPEG2000 context-based approach. When component scalability is needed, this same strategy can be employed in selected components of the image. This is akin to using key frames in video coding. For example, the first of every C components of the volume might be coded without using the prior coefficient approach. In the worst case, to recover one selected component, C components have to be decoded. On average, only $C/2$ components have to be decoded.

As seen in the next section, this strategy does not penalize coding performance significantly except when C is very small. Due to the non-zero-length impulse response of wavelet filters, the degree of scalability achieved with our strategy is often similar to that achieved when the image is decorrelated along the depth axis using wavelet transforms. See Table II for an evaluation of the component scalability achieved by three common wavelet filter-banks (employed in the following experiments) when they are used to decorrelate along the depth axis of a volume. As can be seen, component scalability can be impacted significantly when moderate to large numbers of decomposition levels are applied. Our approach achieves a high degree of component scalability for $C \leq 8$ without penalizing coding performance (see below).

JPEG2000 achieves quality scalability through the use of quality layers. The formation of quality layers entails partial transmission of coefficient magnitudes, so that the decoder can successively refine the magnitude of coefficients as more layers are transmitted. In previous sections we assumed that the magnitude (i.e., φ) and the sign (i.e., ϕ) of the prior coefficient are available at the moment that bits for $\mathcal{W}_{z,y,x}$ are encoded/decoded. This assumption holds when the image is decoded losslessly, since all bits of the previous component are decoded before decoding the current one. Nonetheless, when forming quality layers or when lossy compression is used, the exact magnitude of the prior coefficient may not be available. Let us assume that the prior coefficient has been only partially encoded/decoded until bit j''. The encoder/decoder then reconstructs the magnitude of the prior coefficient as

$$\hat{\varphi} = \begin{cases} 0 & \text{if } j'' > s \\ (\hat{v} + \delta) \cdot \lambda 2^{j''} & \text{if } j'' \leq s \end{cases}, \quad (4)$$

where $\hat{v} = [b_{M-1}, b_{M-2}, ..., b_{j''}]$, δ is the reconstruction factor (determined in this work according to [47]), and λ is the quantization step size. The sign is reconstructed as

$$\hat{\phi} = \begin{cases} \text{unknown} & \text{if } j'' > s \\ \phi & \text{if } j'' \leq s \end{cases}. \quad (5)$$

Pre-computed LUTs for significance and refinement coding can still be utilized when coefficients are partially transmitted by using $\hat{\varphi}$ instead of φ. For sign coding, pre-computed LUTs are used when $\hat{\phi} = \phi$, while equal probability $P_{sign}(d) = 0.5$ is used when $\hat{\phi} = $ unknown. As seen in the next section, the degradation of coding performance due to the use of $\hat{\varphi}$ and $\hat{\phi}$ is modest.

A simple strategy to form quality layers with PCLUT is to use $\hat{\varphi}$ and $\hat{\phi}$ as described previously in combination with the method proposed in [48]. The main assumption behind that rate-distortion optimization method is that bitplane boundaries are nearly optimal as truncation points for the whole image. This implies that truncating all codeblocks at the same bitplane roughly minimizes image distortion for the corresponding bitrate, so that quality layers may be formed at bitplane boundaries as in [49]. Of course, encoding rates or distortions can then only be controlled relatively coarsely. The advantage is that each coefficient can be coded assuming that $\hat{\varphi}$ and $\hat{\phi}$ are reconstructed with the information transmitted for the current layer or, equivalently, for the current bitplane, so both coder and decoder can easily use the same $\hat{\varphi}$ and $\hat{\phi}$ when coding/decoding coefficients.

To summarize, the probability model based on the prior coefficient can be employed in the following modes:

1) *Lossless compression without component scalability*: this mode uses the 2D reversible CDF 5/3 wavelet transform and φ and ϕ to determine probabilities of emitted symbols.
2) *Lossless compression with component scalability*: same strategy as above but the first of every set of C components is encoded with the un-modified JPEG2000 context-based approach.
3) *Lossy compression without component scalability*: this mode uses the 2D irreversible CDF 9/7 or the 2D reversible CDF 5/3 wavelet transform and $\hat{\varphi}$ and $\hat{\phi}$ to determine probabilities of emitted symbols.
4) *Lossy compression with component scalability*: same strategy as above but the first of every set of C components is encoded with the un-modified JPEG2000 context-based approach.

D. Computational complexity

In general, three stages of the coding system require the major computational efforts. These three stages correspond to the operations needed to: 1) remove redundancy along

the depth axis; 2) remove redundancy of components in the vertical and horizontal axes; and 3) code the transformed coefficients using bitplane and entropy coding techniques. The complexity of these operations is $O(N)$ (linear) in the number (N) of samples encoded. However, the slope varies dramatically depending on the details of the procedures. Thus, a more careful accounting of complexity is required.

The computational complexity of JPEG2000, as well as the proposed image codec, can be roughly approximated as the number of operations needed by the wavelet transform to decorrelate the image information *plus* the number of operations needed by the bitplane coding engine to code the emitted symbols. The computational complexity is then roughly $(\#W_z + \#W_{y,x}) + (\#S \cdot \theta)$, with $\#W_z$ and $\#W_{y,x}$ denoting the number of operations carried out by the wavelet transform to decorrelate the image along the depth and the spatial axes, respectively, $\#S$ denoting the number of emitted symbols, and θ denoting the number of operations carried out by the bitplane coding engine when coding one symbol.

As described in the next section, the wavelet transform is commonly applied first in the depth axis, and then in the vertical and horizontal axes, producing an anisotropic decomposition that is referred to as "1D+2D." With this type of decomposition, $\#W_z$ and $\#W_{y,x}$ can be approximated as the length of the wavelet filter-bank kernels multiplied by the total number of samples and then multiplied by 2 and $8/3$, respectively. 2 and $8/3$ account for the application of the wavelet transform in successive levels of decomposition. An extended analysis on the computational complexity of transforms for remote sensing images can be found in [19]. A transform strategy that applies the "1D+2D" decomposition with the reversible CDF 5/3 wavelet transform then requires roughly $(2 \cdot 4 + 8/3 \cdot 4) \cdot N$ operations, where $N = X \cdot Y \cdot Z$ is the number of samples in the image. The effective length of the CDF 5/3 filter-bank is 4 since half of the operations are done with a filter length of 5, and half with a filter length of 3.

The number of symbols emitted per each coefficient coded is dependent on the image, but is upperbounded by the bit depth of the original image. More realistically, the number of symbols emitted per coefficient averages around one half the bit depth as indicated in Table III. This table reports the average number of emitted symbols per coefficient when the "1D+2D" strategy is employed to code one image of each type losslessly with a conventional JPEG2000 implementation. On average, 6.22 symbols are emitted per coefficient coded.

The value of θ is also dependent on the image coded, but can be roughly approximated by appealing to Figure 3. In this figure, it can be seen that in addition to the symbol to be coded, 8 neighboring symbols are considered. This, together with probability adaptation and arithmetic coding, implies on the order of 12 memory accesses and an equal number of arithmetic operations (including bit manipulation). Our implementation employs optimization tricks from [37] resulting in about 12 total operations per symbol emitted, as determined using software profiling tools. The computational complexity of a 3D image codec using a conventional JPEG2000 implementation that employs the "1D+2D" strategy

is then roughly

$$(2 \cdot 4 + 8/3 \cdot 4) \cdot N + 6.22 \cdot 12 \cdot N \ , \qquad (6)$$

which is linear in the number of samples of the image as mentioned previously.

As discussed previously, the probability model of PCLUT is devised to remove the redundancy of the image along the depth axis and to simplify the coding of symbols emitted by the bitplane coding engine. The former avoids the use of a transform along the depth axis of the volume, while the latter results in a reduction of θ. Appealing again to Figure 3, in place of the 8 spatial neighbors, only one prior coefficient is considered, and no probability adaptation is carried out. The figure implies on the order of 4 memory accesses and as many calculations. The simplicity of the method does not admit much further optimization. Indeed, software profiling tools indicate that roughly 7 total operations are carried out in our implementation. As reported in Table III, the average number of emitted symbols per coefficient when decorrelating only the spatial axes of the volume is 7.16. Such a strategy emits more symbols than "1D+2D" due to the existing redundancy among the components of the 3D image. The computational complexity of a 3D image codec using PCLUT is then roughly

$$8/3 \cdot 4 \cdot N + 7.16 \cdot 7 \cdot N \ , \qquad (7)$$

which is also linear in the number of samples N.

Even though the computational complexity of (6) and (7) are both linear in N, the difference in computational complexity between the two codecs is approximately 35%.

IV. EXPERIMENTAL RESULTS

A. Coding performance

The coding performance of PCLUT is evaluated for five different types of 3D images, as reported in Table I. Detailed results are reported for the images of Table I. Summary results are reported for a much larger corpus of images. LUTs are generated using data from the first image of each type (as reported in the last column of Table I) and employed for the remaining ones. For the Computed Tomography (CT) images, the LUTs are generated from image "A" and "D" and used to code images "B,""C" and "E,""F," respectively, due to the use of two different types of CT sensors. Results are reported only for images that are *not* in the training set.

PCLUT is compared to three transform strategies that are employed in the literature to assess the performance of 3D coding schemes. These strategies are:

1) No transform along the depth axis, and 5 levels of wavelet transform in the spatial axes (denoted by "2D").
2) 1 level of Haar transform along the depth axis, and 5 levels of wavelet transform in the spatial axes (denoted by "Haar+2D").
3) 5 levels of wavelet transform along the depth axis, and 5 levels of wavelet transform in the spatial axes (denoted by "1D+2D").

TABLE III: Average number of emitted symbols per coded coefficient when PCLUT and JPEG2000 code images losslessly using a transform strategy that decorrelates image information along two and three axes of the volume, respectively.

	PCLUT 2D	JPEG2000 1D+2D
AVIRIS - cuprite	7.6	5.6
Hyperion - agricultural	7.6	6.2
Computed Tomography - A	8.9	8.5
Angiography - A	6.9	6.6
CIF - bridge	4.8	4.2
AVERAGE	*7.16*	*6.22*

In addition to the Haar transform mentioned above, two different wavelet transforms are considered as discussed below. Wavelet transforms are first applied in the depth axis, and then in the vertical and horizontal axes, producing an anisotropic decomposition. The anisotropic decomposition is chosen herein because it generally achieves better performance than other types of decomposition structures [16].

The first strategy above is supported in Part 1 of JPEG2000 and is the most common approach to code gray or YCbCr images. The other two strategies are supported by JPEG2000 Part 2 and are commonly employed in the medical and remote sensing fields to code data acquired by 3D-specific sensors. The modification of the probability model proposed by PCLUT goes beyond the scope of the JPEG2000 standard, though, as stated previously, PCLUT can be implemented in any bitplane image coding engine. JPEG2000 is chosen here due to its suitability to code 3D images, its widespread use, and to provide a fair comparison with the other strategies. Also, JPEG2000 provides an appropriate framework since it is commonly adopted in many scenarios that use 3D images. The first transform strategy is the simplest and is employed by memory-constrained applications. The second strategy employs a short filter-bank on the depth axis, leading to high component scalability and moderate memory requirements. We note that the memory requirements and component scalability of this strategy are very similar to the proposed PCLUT strategy (see below). The third strategy is used when neither memory requirements nor component scalability are restrictive.

Coding performance of JPEG2000 is evaluated for the three transform strategies, and is compared to that of PCLUT. JPEG2000 coding parameters are: irreversible CDF 9/7, or reversible CDF 5/3 wavelet transform depending on lossy or lossless compression, codeblock size of 64×64, single quality layer codestreams, no precincts. The base quantization step sizes, corresponding to bitplane 0, when the CDF 9/7 filter-bank is used are chosen according to the L_2-norm of the synthesis basis vectors of the subband [37, Ch. 10.5.1], which is a common practice in JPEG2000.

First, we evaluate coding performance for lossless compression (mode 1 of Section III-C). Columns 3-5 of Table IV report the codestream length, in bits per sample (bps), generated when encoding the 3D images with the three transform strategies described above. The 6th column reports the performance of PCLUT. Experimental results suggest that PCLUT achieves a significant gain in coding efficiency compared to

many other strategies, especially those that employ only 2D transforms. For two of the five image types (namely, Hyperion and Angiography), PCLUT achieves the best lossless results, surpassing "1D+2D" strategies. For four of the five image types, PCLUT surpasses the efficiency of the "Haar+2D" strategy. These results suggest that the proposed probability model is well suited for 3D images with spectral and spatial information along the depth axis. Even though PCLUT achieves a coding gain of approximately 11% compared to a "2D" strategy for video sequences, results suggest that temporal correlation among frames of a video sequence is more efficiently decorrelated with filters that consider many previous/following components. As indicated by these results, PCLUT is not especially indicated for video sequences. Even so, video coding results are reported to show the increase in efficiency with respect to a "2D" strategy and to perform a more general assessment on different types of 3D images.

The last three columns of Table IV report the coding performance of PCLUT for lossless compression with component scalability (mode 2 of Section III-C). Representative values for parameter C are selected. Coding performance is only slightly penalized for $C = 32$ and $C = 8$, which indicates that PCLUT can provide component scalability with competitive coding performance.

To further validate the achieved results, an extended corpus containing 93 images is employed to perform the same test as that of Table IV. This corpus contains 18 AVIRIS images, 21 Hyperion images, 16 CT images, 23 Angiography images, and 15 CIF video sequences. Table V reports the results achieved with the extended corpus, on average for each type of 3D image. The results obtained are similar to those of Table IV, which confirms the effectiveness of PCLUT to code these types of 3D images.

Figure 4 depicts lossy coding performance for the same strategies as above, in terms of Signal to Noise Ratio (SNR).[2] The figure depicts performance for selected images that do *not* correspond to those used to extract the LUTs. Results for a wider variety of images are discussed below. The graphs in Figure 4 report the lossy coding performance with and without component scalability (modes 4 and 3 of Section III-C, respectively). Again, results suggest that the proposed probability model significantly improves the performance of the coding

---

[2]The "Haar+2D" strategy for the angiography and videoconferencing images in Figures 4(d) and (e) uses the reversible Haar transform since it achieves slightly superior coding performance than the irreversible Haar transform.

TABLE IV: Evaluation of lossless coding performance. Results reported in bits per sample (bps).

| | | JPEG2000 | | | PCLUT | PCLUT | | |
| | | | | | | $C = 32$ | $C = 8$ | $C = 2$ |
		2D	Haar+2D	1D+2D	2D	2D	2D	2D
AVIRIS	jasper sc01	7.66	6.56	5.54	5.88	5.92	6.08	6.74
	jasper sc02	7.01	6.15	5.28	5.57	5.60	5.73	6.27
	lowAltitude sc01	7.83	6.86	5.95	6.39	6.42	6.55	7.10
	lowAltitude sc02	7.66	6.56	5.54	5.88	5.92	6.08	6.74
	lunarLake	7.83	6.86	5.95	6.39	6.42	6.55	7.10
AVERAGE		*7.60*	*6.60*	***5.65***	*6.02*	*6.06*	*6.20*	*6.79*
Hyperion	urban	7.09	6.37	6.27	6.03	6.03	6.11	6.46
	flooding	6.58	6.02	6.10	5.86	5.86	5.91	6.12
	forestry	6.73	6.09	6.06	5.79	5.80	5.86	6.15
	coastal	7.09	6.37	6.27	6.03	6.03	6.11	6.46
	tornado	6.58	6.02	6.10	5.87	5.86	5.91	6.12
AVERAGE		*6.81*	*6.17*	*6.16*	***5.92***	*5.92*	*5.98*	*6.26*
Computed Tomography	B	8.41	8.23	8.07	8.00	8.02	8.05	8.20
	C	8.33	8.16	8.01	7.97	7.97	8.01	8.14
	E	4.04	3.80	3.60	3.90	3.90	3.92	3.99
	F	3.84	3.60	3.40	3.67	3.67	3.69	3.77
AVERAGE		*6.16*	*5.95*	***5.77***	*5.89*	*5.89*	*5.92*	*6.03*
Angiography	B	6.48	6.40	6.33	6.21	6.22	6.24	6.35
	C	6.35	6.28	6.22	6.09	6.09	6.12	6.22
	D	6.37	6.30	6.23	6.09	6.10	6.12	6.23
	E	6.48	6.40	6.33	6.21	6.22	6.24	6.35
	F	6.35	6.28	6.22	6.09	6.09	6.12	6.22
AVERAGE		*6.41*	*6.33*	*6.27*	***6.14***	*6.14*	*6.17*	*6.27*
CIF	paris	4.85	3.92	3.06	3.94	3.97	4.05	4.40
	salesman	4.44	3.78	3.10	4.13	4.13	4.16	4.29
	closed bridge	4.28	3.86	3.36	3.85	3.86	3.90	4.06
	news	3.56	2.64	1.86	3.25	3.25	3.28	3.40
	waterfall	4.70	4.15	3.60	4.25	4.26	4.30	4.47
AVERAGE		*4.37*	*3.67*	***3.00***	*3.88*	*3.89*	*3.94*	*4.12*

TABLE V: Evaluation of lossless coding performance for an extended corpus of images. Results reported in bits per sample (bps).

| | JPEG2000 | | | PCLUT | PCLUT | | |
| | | | | | $C = 32$ | $C = 8$ | $C = 2$ |
	2D	Haar+2D	1D+2D	2D	2D	2D	2D
AVIRIS (18 images)	7.47	6.52	**5.60**	5.95	5.99	6.12	6.69
Hyperion (21 images)	6.58	6.02	6.10	**5.86**	5.86	5.91	6.12
Computed Tomography (16 images)	5.88	5.68	**5.52**	5.80	5.80	5.81	5.84
Angiography (23 images)	6.46	6.38	6.32	**6.19**	6.20	6.23	6.33
CIF (15 images)	4.31	3.71	**3.12**	3.80	3.81	3.86	4.05

system. Note that PCLUT uses the same transform strategy as that labeled "2D." By changing only the probability models of emitted symbols, PCLUT boosts coding performance to that achieved by a more sophisticated transform strategy that uses 1 level of wavelet transform along the depth axis of the volume. Classic approaches of context-adaptive coding achieve much less coding gain when 3D context models are employed (see, for instance, [14]). The transform strategy "1D+2D" achieves better rate-distortion performance than PCLUT at low bitrates for most images. Nonetheless "1D+2D" strategies have very high memory requirements and low component scalability (see below), which may render them impractical in devices with limited resources.

In addition to the aforementioned coding strategies, Fig-

ure 4(e) also reports the results achieved when using two profiles of the video coding standard H.264.[3] The H.264 profile "baseline" achieves coding performance similar to that of the "1D+2D" strategy, whereas the H.264 "high" profile achieves the highest results reported in that figure. Even though PCLUT improves over the performance of both the "2D" and "Haar+2D" strategies, results suggest that the "1D+2D" strategy and H.264 achieve better coding performance for this particular type of 3D images.

It is also apparent from Figure 4 that the difference between

[3]The free libraries x264 (see http://www.videolan.org/developers/x264.html) have been employed to code the sequences with H.264. Coding parameters are selected to achieve highest compression efficiency regardless of the computation time spent.

(a) AVIRIS - "jasper"

(b) Hyperion - "urban"

(c) Computed Tomography - "B"
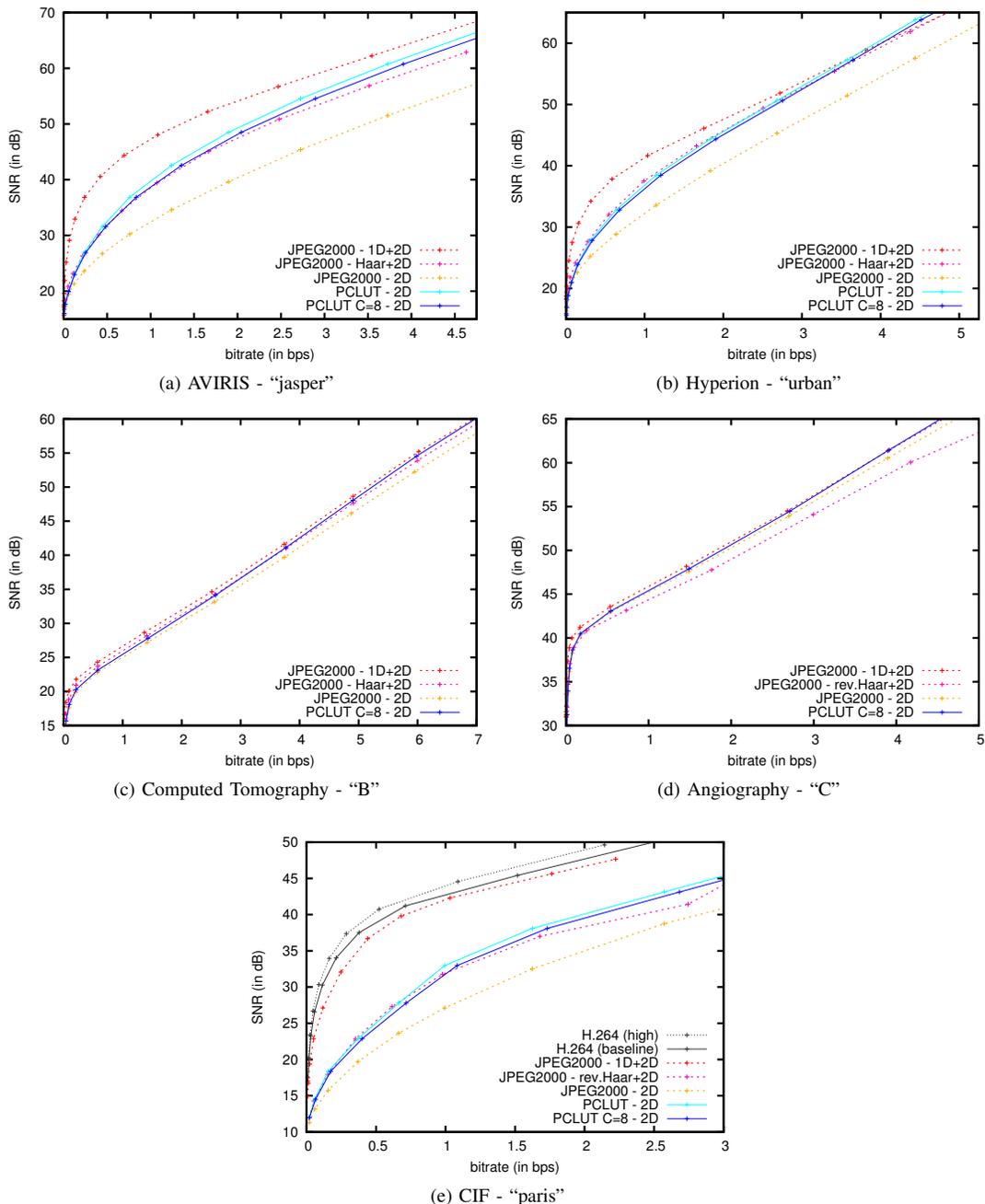
(d) Angiography - "C"

(e) CIF - "paris"

Fig. 4: Evaluation of lossy coding performance.

using and not using component scalability is small for most images. For the selected computed tomography and angiography images this difference is not discernible, so only one PCLUT mode is shown. We note that the points depicted in the graphs of Figure 4 are the natural truncation points achieved at the bitplane boundaries, which are used for quality scalability purposes.

Results for other images of Table I, as well as the extended corpus discussed above, are similar to those shown in Figure 4, confirming the effectiveness of PCLUT to code these 3D images. Results obtained for the images of Table I can be found at [50].

The enhancement of coding performance achieved by PCLUT leads to images with better visual appearance. Figure 5 depicts three images belonging to the remote sensing, medical, and videoconferencing fields. Each image is coded using the "1D+2D," and "2D" transform strategies, as well as PCLUT. At the selected target bitrates, the image decoded with the "2D" strategy is blurry and ringing artifacts are clearly visible. The image decoded with the "1D+2D" strategy is more visually appealing and does not present visual artifacts. While requiring significantly less computational resources than the "1D+2D" and "2D" strategies (see below), PCLUT achieves an image that is significantly clearer than that achieved with the "2D" strategy and similar, though not as good as, to that

12

achieved with the "1D+2D" strategy.

The last test assessing coding performance employs images in which the correlation along the depth axis has been decreased artificially. The aim of this test is to evaluate whether the proposed scheme is effective for images that have low correlation between components. The AVIRIS images employed in Table IV are subsampled in the component direction selecting one out of two, three, and five components. Since the original images have 224 components, this produces three artificial images for each original having 112, 75, and 45 components, respectively. As expected, the correlation in the component direction decreases as fewer components are included in the image. Table VI reports the results achieved when these images are coded losslessly by the aforementioned strategies. For completeness, the table also provides the results achieved when coding the full volume. The correlation coefficient between components of each image is reported in parentheses beside the image name. As expected, the "2D" strategy achieves similar coding performance regardless of the correlation between components. The small variations are caused due to the different subsets of components being selected for each subsampling ratio. Also as expected, the performance achieved for each image by "Haar+2D," "1D+2D," and PCLUT all decrease with decreasing correlation. The decrease in performance is similar for all three strategies. In general, PCLUT achieves better and worse performance than that achieved by the "Haar+2D" and "1D+2D" strategies, respectively, the same behavior as for the original image. Only for the "low Altitude sc01" and "low Altitude sc02" images having 45 components, "Haar+2D" achieves better performance than PCLUT. Notably, "Haar+2D" also equals or betters the performance of "1D+2D" for these two cases. On the other hand, for the "jasper sc02" image having 45 components, PCLUT achieves the same coding performance as that achieved by the "1D+2D" strategy. These results indicate that the performance decreases of the proposed scheme are similar to those of other schemes when the correlation between components decreases. Similar results hold for the other types of images.

B. Computational costs

Computational performance tests are performed on an Intel Core i7 CPU at 2.93 GHz. All methods are implemented in our Java implementation BOI[4], and executed on a Java Virtual Machine v1.6 using GNU/Linux v2.6. Results are reported as CPU processing time. The implementation of the JPEG2000 context-adaptive approach uses several software optimizations as suggested in [37, Ch. 17.1.2]. PCLUT uses a similar degree of optimization. Since all methods are tested in the same code base, differences in throughput are relevant, even though all run-times might be reduced by the use of a compiled language. Similar run-times are produced for the four modes of PCLUT utilization described in Section III-C, thus only results for the first one (lossless compression without component scalability) are reported.

[4]See: http://www.deic.uab.es/~francesc

Table VII presents the throughput achieved for the three aforementioned JPEG2000 strategies, and PCLUT. The table distinguishes the main stages carried out by the coder: tier-1 coding; application of the wavelet transform on the components and (possibly) on the depth axis; and other operations. Other operations include tier-2 coding, level shift, quantization, and codestream re-organization. Experimental results suggest that, on average for all the five types of images, PCLUT is approximately 30%, 31%, and 33% more computationally efficient than the strategies "2D," "Haar+2D," and "1D+2D," respectively. The main gain is achieved at the tier-1 coding stage due to the simplification of the probability model. These results coincide with the computational complexity analysis described in Section III-D.

The peak memory requirements of these strategies may vary significantly depending on the implementation. The use of the local wavelet transform [51], software optimization procedures, or hardware-based optimization may have an important impact on the memory requirements. Therefore, our evaluation is only indicative. We report peak memory requirements using the local wavelet transform along the depth axis, which is a common approach since otherwise the full volume would be maintained in memory. Table VIII presents results for the Haar, CDF 9/7, and CDF 5/3 wavelet filter-banks. The same results hold for all images of each type. Results indicate that peak memory requirements for strategies that employ more than one level of wavelet decomposition along the depth axis, namely the "1D+2D" strategies, are an order of magnitude higher than strategies that only apply one or zero levels of decomposition. For constrained resource devices, this can make a significant difference. For the "cuprite" image, for instance, the peak memory requirement of PCLUT is 0.9% of the total raw size of the image (which is 112 MB). On the other hand, although the memory requirements of PCLUT and "Haar+2D" are equivalent, we remark that the implementation of PCLUT in hardware-based implementations would require much less circuitry due to the simplicity of our approach.

In summary, PCLUT is roughly 1/3 more computationally efficient than the "Haar+2D" transform strategy while achieving the same coding performance, the same memory requirements, and the same component scalability. Compared to the "2D" strategy, PCLUT is 1/3 more computationally efficient, with a significant increase in coding performance and negligible increase in memory and scalability capabilities. "1D+2D" transform strategies achieve better coding efficiency than PCLUT only at low lossy compression rates and at the expense of very high memory requirements and low component scalability.

V. CONCLUSIONS

The coding of 3D images is an important topic of interest in several fields. The most common approach begins with the application of a decorrelating transform. For lossy/lossy-to-lossless regimes this transform is typically followed by a bitplane coding engine and context-adaptive arithmetic coding. Numerous works have studied transform strategies, bitplane coding engines, and context formation approaches to enhance

(a) JPEG2000 1D+2D - 33.28 dB   (b) PCLUT 2D - 23.59 dB   (c) JPEG2000 2D - 21.98 dB

(d) JPEG2000 1D+2D - 39.64 dB   (e) PCLUT 2D - 30.84 dB   (f) JPEG2000 2D - 25.63 dB

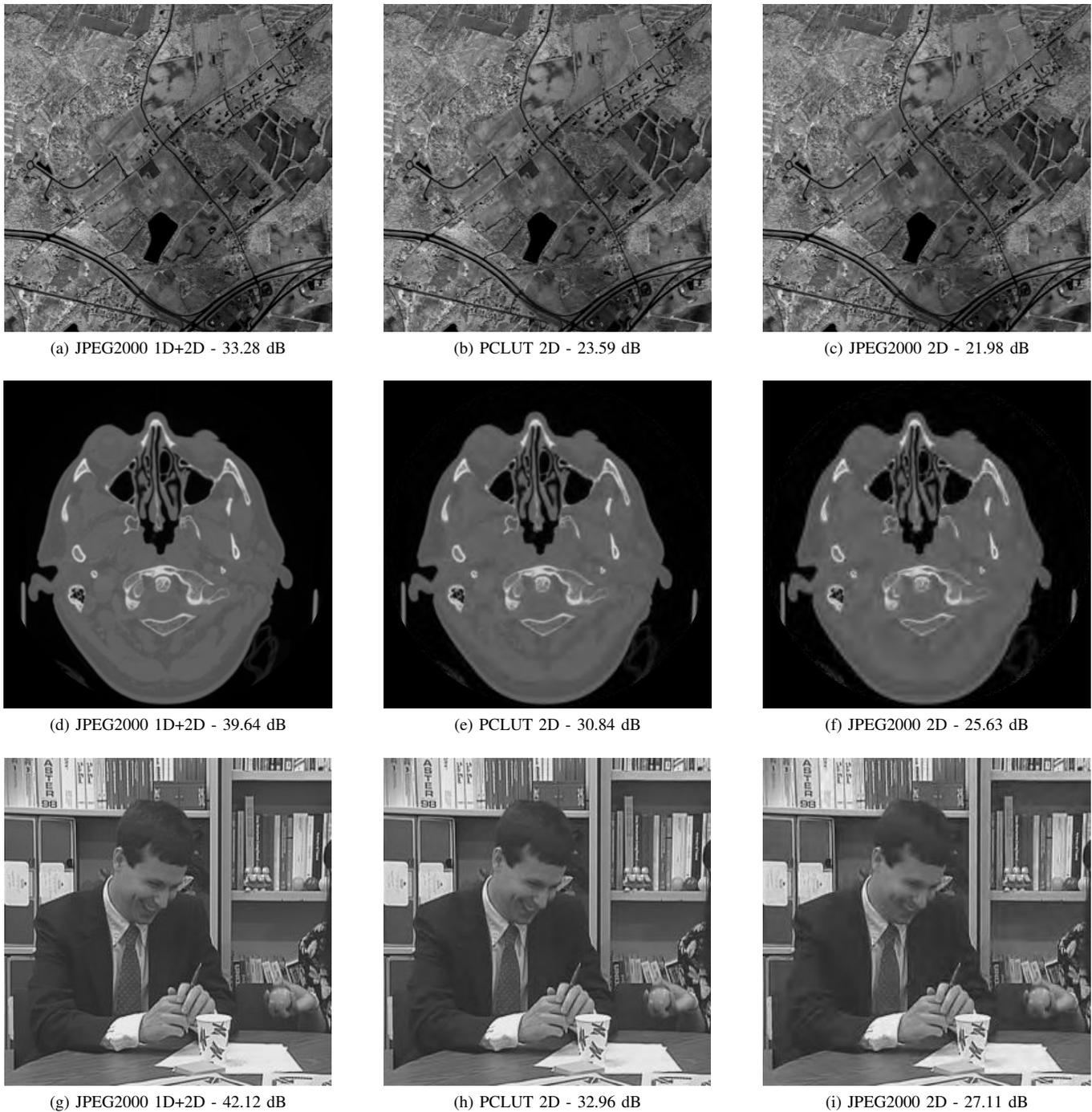(g) JPEG2000 1D+2D - 42.12 dB   (h) PCLUT 2D - 32.96 dB   (i) JPEG2000 2D - 27.11 dB

Fig. 5: Visual evaluation. (a), (b), (c) is an area from component 48 of AVIRIS "lowAltitude sc01" coded at 0.19 bps. (d), (e), (f) is component 9 of CT "F" coded at 0.14 bps. (g), (h), (i) is an area from frame 50 of video sequence "paris" coded at 1 bps.

the coding efficiency of 3D image coding schemes. Most studies have been centered on one particular type of 3D image.

This paper introduces a scheme for the coding of many different types of 3D images. This scheme employs a novel probability model for symbols emitted by the bitplane coding engine. Only employing the partially reconstructed magnitude and sign of coefficients encoded in previous components, the proposed method is able to achieve competitive coding performance in lossy-to-lossless regimes. This boost in coding efficiency is achieved through the use of an accurate probability model that is able to adjust the probabilities fed to the arithmetic coder with high precision. This increase in coding efficiency is commonly *not* achieved by other state-of-the-art probability models that employ traditional 2D or 3D context-adaptive approaches. The main advantages of the proposed approach are low computational complexity, very low memory

TABLE VI: Evaluation of lossless coding performance when the correlation along the depth axis decreases. Results are reported for the AVIRIS images in bits per sample (bps).

			JPEG2000			PCLUT
			2D	Haar+2D	1D+2D	2D
full volume (224 components)	jasper sc01	(0.96)	7.66	6.56	5.54	5.88
	jasper sc02	(0.96)	7.01	6.15	5.28	5.57
	lowAltitude sc01	(0.93)	7.83	6.86	5.95	6.39
	lowAltitude sc02	(0.93)	7.66	6.56	5.54	5.88
	lunarLake	(0.96)	7.83	6.86	5.95	6.39
	AVERAGE	*(0.95)*	*7.60*	*6.60*	*5.65*	*6.02*
1 out of 2 components (112 components)	jasper sc01	(0.95)	7.66	6.75	6.07	6.28
	jasper sc02	(0.95)	7.55	6.66	6.00	6.20
	lowAltitude sc01	(0.92)	7.82	7.04	6.46	6.84
	lowAltitude sc02	(0.93)	7.96	7.15	6.54	6.97
	lunarLake	(0.95)	6.92	6.23	5.60	5.79
	AVERAGE	*(0.94)*	*7.58*	*6.77*	*6.13*	*6.42*
1 out of 3 components (75 components)	jasper sc01	(0.94)	7.67	6.92	6.43	6.54
	jasper sc02	(0.94)	7.56	6.83	6.36	6.45
	lowAltitude sc01	(0.90)	7.84	7.21	6.82	7.12
	lowAltitude sc02	(0.91)	7.97	7.33	6.92	7.29
	lunarLake	(0.94)	6.92	6.34	5.81	5.92
	AVERAGE	*(0.92)*	*7.59*	*6.93*	*6.47*	*6.66*
1 out of 5 components (45 components)	jasper sc01	(0.91)	7.73	7.11	7.01	7.03
	jasper sc02	(0.91)	7.61	7.01	6.91	6.91
	lowAltitude sc01	(0.89)	7.88	7.37	7.39	7.63
	lowAltitude sc02	(0.90)	8.01	7.50	7.50	7.84
	lunarLake	(0.93)	6.97	6.47	6.09	6.16
	AVERAGE	*(0.90)*	*7.64*	*7.09*	*6.98*	*7.11*

TABLE VII: Computational throughput evaluation when decoding the full image (coded in lossless mode without quality scalability). Results reported as CPU processing time (in seconds).

	PCLUT - 2D				JPEG2000 - 2D				JPEG2000 - Haar+2D				JPEG2000 - 1D+2D			
	tier-1	wav.	other	*TOT.*	tier-1	wav.	other	*TOT.*	tier-1	wav.	other	*TOT.*	tier-1	wav.	other	*TOT.*
AVIRIS - cuprite	15.0	2.7	1.8	*19.4*	27.3	2.5	1.3	*31.2*	24.4	4.1	1.3	*29.9*	22.6	5.8	1.8	*30.1*
AVIRIS - jasper sc01	18.1	2.6	1.9	*22.7*	28.7	2.5	1.3	*32.5*	25.7	4.3	1.4	*31.4*	23.1	5.8	1.7	*30.6*
Hyperion - agricultural	12.0	2.0	1.6	*15.6*	20.2	1.9	1.4	*23.5*	18.7	4.1	1.5	*24.2*	19.0	5.3	1.2	*25.5*
Hyperion - urban	14.8	1.9	1.6	*18.3*	20.6	1.8	1.4	*23.8*	19.0	3.2	1.6	*23.9*	19.1	5.3	1.2	*25.7*
Computed Tomography - A	10.8	1.3	1.0	*13.1*	15.1	1.2	0.7	*17.0*	15.0	1.9	0.7	*17.7*	15.7	2.7	0.8	*19.2*
Computed Tomography - B	11.2	1.3	1.0	*13.5*	15.2	1.2	0.7	*17.2*	15.1	1.9	0.7	*17.7*	14.9	2.7	0.8	*18.5*
Angiography - A	10.9	1.8	1.2	*13.9*	17.0	1.7	1.0	*19.7*	17.0	2.7	1.0	*20.7*	17.1	3.8	0.9	*21.8*
Angiography - B	11.5	1.7	1.1	*14.3*	17.4	1.7	1.0	*20.2*	17.1	2.7	1.0	*20.8*	17.5	3.8	0.9	*22.2*
Videoconf. - bridge	10.0	2.0	1.5	*13.5*	14.7	1.9	1.3	*18.0*	14.1	4.5	1.2	*19.8*	13.7	5.5	1.2	*20.5*
Videoconf. - paris	9.7	1.9	1.5	*13.1*	17.8	2.0	1.3	*21.1*	15.2	4.4	1.2	*20.8*	12.8	5.2	1.2	*19.2*
AVERAGE	*12.4*	*1.9*	*1.4*	*15.7*	*19.4*	*1.9*	*1.2*	*22.4*	*18.1*	*3.4*	*1.2*	*22.7*	*17.5*	*4.6*	*1.2*	*23.3*

TABLE VIII: Peak memory requirements evaluation. Results reported in Megabytes (MB).

	PCLUT	JPEG2000			
				CDF 5/3 rev.	CDF 9/7 irrev.
	2D	2D	Haar+2D	1D+2D	1D+2D
AVIRIS - cuprite	1.0	0.5	1.0	8.0	18.0
Hyperion - agricultural	0.8	0.4	0.8	6.4	14.4
Computed Tomography - A	1.0	0.5	1.0	8.0	18.0
Angiography - A	1.0	0.5	1.0	8.0	18.0
CIF - bridge	0.2	0.1	0.2	1.6	3.6
AVERAGE	*0.8*	*0.4*	*0.8*	*6.4*	*14.4*

requirements, straightforward implementation, short training stage, and adaptability to most types of 3D images. The proposed probability model can be integrated in most bitplane coding systems. Hardware implementations, surveillance and teleconferencing, and constrained resource devices are some applications that may benefit from the proposed scheme.

REFERENCES

[1] J. Wang and H. K. Huang, "Medical image compression by using three-dimensional wavelet transformation," *IEEE Trans. Med. Imag.*, vol. 15, no. 4, pp. 547–554, Aug. 1996.

[2] N. Sriraam and R. Shyamsunder, "3-D medical image compression using 3-D wavelet coders," *ELSEVIER Digital Signal Processing*, vol. 21, no. 1, pp. 100–109, Jan. 2011.

[3] Y. Shang, L. Li, and B. Wah, "Optimization design of biorthogonal filter banks for image compression," *ELSEVIER Information Sciences*, vol. 132, no. 1-4, pp. 23–51, Feb. 2001.

[4] L. Zeng, C. P. Jansen, S. Marsch, M. Unser, and P. R. Hunziker, "Four-dimensional wavelet compression of arbitrarily sized echocardiographic data," *IEEE Trans. Med. Imag.*, vol. 21, no. 9, pp. 1179–1187, Sep. 2002.

[5] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto, and J. Cornelis, "Wavelet coding of volumetric medical datasets," *IEEE Trans. Med. Imag.*, vol. 22, no. 3, pp. 441–458, Mar. 2003.

[6] Z. Xiong, X. Wu, S. Cheng, and J. Hua, "Lossy-to-lossless compression of medical volumetric data using three-dimensional integer wavelet transforms," *IEEE Trans. Med. Imag.*, vol. 22, no. 3, pp. 459–470, Mar. 2003.

[7] F. Y. Shih and Y.-T. Wu, "Robust watermarking and compression for medical images based on genetic algorithms," *ELSEVIER Information Sciences*, vol. 175, no. 3, pp. 200–216, Oct. 2005.

[8] X. Qi and J. M. Tyler, "A progressive transmission capable diagnostically lossless compression scheme for 3D medical image sets," *ELSEVIER Information Sciences*, vol. 175, no. 3, pp. 217–243, Oct. 2005.

[9] K. Krishnan, M. W. Marcellin, A. Bilgin, and M. S. Nadar, "Efficient transmission of compressed data for remote volume visualization," *IEEE Trans. Med. Imag.*, vol. 25, no. 9, pp. 1189–1199, Sep. 2006.

[10] V. Sanchez, R. Abugharbieh, and P. Nasiopoulos, "Symmetry-based scalable lossless compression of 3D medical image data," *IEEE Trans. Med. Imag.*, vol. 28, no. 7, pp. 1062–1072, Jul. 2009.

[11] L. Wang, J. Wu, L. Jiao, and S. Zhang, "Three-dimensional medical image compression based on low-complexity RKLT," *IET Electronics Letters*, vol. 46, no. 6, pp. 395–397, Mar. 2010.

[12] *Information technology - JPEG 2000 image coding system - Part 1: Core coding system*, ISO/IEC Std. 15 444-1, Dec. 2000.

[13] Y. Zhang, B. T. Pham, and M. P. Eckstein, "Automated optimization of JPEG 2000 encoder options based on model observer performance for detecting variable signals in X-Ray coronary angiograms," *IEEE Trans. Med. Imag.*, vol. 23, no. 4, pp. 459–474, Apr. 2004.

[14] R. Leung and D. Taubman, "Transform and embedded coding techniques for maximum efficiency and random accessibility in 3-D scalable compression," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1632–1646, Oct. 2005.

[15] P. L. Dragotti, G. Poggi, and A. R. P. Ragozini, "Compression of multispectral images by three-dimensional SPIHT algorithm," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 1, pp. 416–428, Jan. 2000.

[16] E. Christophe, C. Mailhes, and P. Duhamel, "Hyperspectral image compression: Adapting SPIHT and EZW to anisotropic 3-D wavelet coding," *IEEE Trans. Image Process.*, vol. 17, no. 12, pp. 2334–2346, Dec. 2008.

[17] B. Penna, T. Tillo, E. Magli, and G. Olmo, "Transform coding techniques for lossy hyperspectral data compression," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 5, pp. 1408–1421, May 2007.

[18] J. Zhang, J. E. Fowler, and G. Liu, "Lossy-to-lossless compression of hyperspectral imagery using three-dimensional TCE and an integer KLT," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 4, pp. 814–818, Oct. 2008.

[19] I. Blanes and J. Serra-Sagrista, "Cost and scalability improvements to the Karhunen-Loeve transform for remote-sensing image coding," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2854–2863, Jul. 2010.

[20] M. Cagnazzo, L. Cicala, G. Poggi, and L. Verdoliva, "Low-complexity compression of multispectral images based on classified transform coding," *ELSEVIER Signal Processing: Image Communication*, vol. 21, no. 10, pp. 850–861, Nov. 2006.

[21] B. Penna, T. Tillo, E. Magli, and G. Olmo, "Hyperspectral image compression employing a model of anomalous pixels," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 4, pp. 664–668, Oct. 2007.

[22] ——, "Progressive 3-D coding of hyperspectral images based on JPEG 2000," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 125–129, Jan. 2006.

[23] Q. Du and J. E. Fowler, "Hyperspectral image compression using JPEG2000 and principal component analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 2, pp. 201–205, Apr. 2007.

[24] G. Carvajal, B. Penna, and E. Magli, "Unified lossy and near-lossless hyperspectral image compression based on JPEG 2000," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 4, pp. 593–597, Oct. 2008.

[25] *Information technology - JPEG 2000 image coding system - Part 10: Extensions for three-dimensional data*, ISO/IEC Std. 15 444-10, Dec. 2008.

[26] *Advanced video coding for generic audiovisual services*, International Telecommunication Union Std. H.264, 2005.

[27] F.-O. Devaux, J. Meessen, C. Parisot, J.-F. Delaigle, B. Macq, and C. D. Vleeschouwer, "A flexible video transmission system based on JPEG 2000 conditional replenishment with multiple references," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, Apr. 2007, pp. 825–828.

[28] ——, "Remote interactive browsing of video surveillance content based on JPEG 2000," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 8, pp. 1143–1157, Aug. 2009.

[29] F. Rizzo, J. A. Storer, and B. Carpentieri, "LZ-based image compression," *ELSEVIER Information Sciences*, vol. 135, no. 1, pp. 107–122, Jun. 2001.

[30] V. Baligar, L. Patnaik, and G. Nagabhushana, "Low complexity, and high fidelity image compression using fixed threshold method," *ELSEVIER Information Sciences*, vol. 176, no. 6, pp. 664–675, Mar. 2006.

[31] J. H. Reif and J. Storer, "Optimal encoding of non-stationary sources," *ELSEVIER Information Sciences*, vol. 135, no. 1-2, pp. 97–105, Jun. 2001.

[32] J. Walder, M. Kratky, R. Baca, J. Platos, and V. Snasel, "Fast decoding algorithms for variable-lengths codes," *ELSEVIER Information Sciences*, vol. 183, no. 1, pp. 66–91, Jan. 2012.

[33] G. Menegaz and J.-P. Thiran, "Three-dimensional encoding/two-dimensional decoding of medical data," *IEEE Trans. Med. Imag.*, vol. 22, no. 3, pp. 424–440, Mar. 2003.

[34] L. Santos, S. Lopez, G. M. Callico, J. F. Lopez, and R. Sarmiento, "Performance evaluation of the H.264/AVC video coding standard for lossy hyperspectral image compression," *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, vol. 5, no. 2, pp. 451–461, Apr. 2012.

[35] P. G. Peterson, S. K. Pak, B. Nguyen, G. Jacobs, and L. Folio, "Extreme compression for extreme conditions: Pilot study to identify optimal compression of CT images using MPEG-4 video compression," *SPRINGER Journal of Digital Imaging*, vol. 25, pp. 764–770, Dec. 2012.

[36] N. Alajlana, Y. Bazia, F. Melganib, and R. R. Yager, "Fusion of supervised and unsupervised learning for improved classification of hyperspectral images," *ELSEVIER Information Sciences*, vol. 217, no. 1, pp. 39–55, Dec. 2012.

[37] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image compression fundamentals, standards and practice*. Norwell, Massachusetts 02061 USA: Kluwer Academic Publishers, 2002.

[38] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE Trans. Commun.*, vol. 45, no. 4, pp. 437–444, Apr. 1997.

[39] *JPEG-LS lossless and near-lossless compression for continuous-tone still images*, ISO/IEC Std. 14 495-1, 1999.

[40] B. Felts and B. Pesquet-Popescu, "Efficient context modeling in scalable 3-D wavelet-based video compression," in *Proc. IEEE International Conference on Image Processing*, vol. 4, Sep. 2000, pp. 1004–1007.

[41] H. Wang, S. D. Babacan, and K. Sayood, "Lossless hyperspectral-image compression using context-based conditional average," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 4187–4193, Dec. 2007.

[42] E. Magli, "Multiband lossless compression of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 4, pp. 1168–1178, Apr. 2009.

[43] J. Liu and P. Moulin, "Information-theoretic analysis of interscale and intrascale dependencies between image wavelet coefficients," *IEEE Trans. Image Process.*, vol. 10, no. 11, pp. 1647–1658, Nov. 2001.

[44] F. Auli-Llinas, "Stationary probability model for bitplane image coding through local average of wavelet coefficients," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2153–2165, Aug. 2011.

[45] A. B. Kiely and M. A. Klimesh, "Exploiting calibration-induced artifacts in lossless compression of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 8, pp. 2672–2678, Aug. 2009.

[46] *Information technology - JPEG 2000 image coding system - Part 2: Extensions*, ISO/IEC Std. 15 444-2, Apr. 2004.

[47] F. Auli-Llinas and M. W. Marcellin, "Distortion estimators for bitplane image coding," *IEEE Trans. Image Process.*, vol. 18, no. 8, pp. 1772–1781, Aug. 2009.

[48] F. Auli-Llinas and J. Serra-Sagrista, "JPEG2000 quality scalability without quality layers," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, pp. 923–936, Jul. 2008.

[49] F. Auli-Llinas, J. Bartrina-Rapesta, and J. Serra-Sagrista, "Enhanced JPEG2000 quality scalability through block-wise layer truncation," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 1–11, 2010, article ID 803542.

[50] F. Auli-Llinas, M. Marcellin, J. Serra-Sagrista, and J. Bartrina-Rapesta. (2013) PCLUT - extended results. [Online]. Available: http://www.deic.uab.es/~francesc/research/pclut/extended_results

[51] C. Chrysafis and A. Ortega, "Line-based, reduced memory, wavelet image compression," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 378–389, Mar. 2000.