Scanning Order Strategies for Bitplane Image Coding

Francesc Aulí-Llinàs, *Member, IEEE* and Michael W. Marcellin, *Fellow, IEEE*

***Abstract*—Scanning orders of bitplane image coding engines are commonly envisaged from theoretical or experimental insights, and assessed in practice in terms of coding performance. This work evaluates classic scanning strategies of modern bitplane image codecs using several theoretical-practical mechanisms conceived from rate-distortion theory. The use of these mechanisms allows distinguishing those features of the bitplane coder that are essential from those that are not. This discernment can aid the design of new bitplane coding engines with some special purposes and/or requirements. To emphasize this point, a low-complexity scanning strategy is proposed. Experimental evidence illustrates, assesses, and validates the proposed mechanisms and scanning orders.**

***Index Terms*—Bitplane image coding, scanning orders, probability models, JPEG2000.**

I. Introduction

BITPLANE image coding is the prevailing technology in lossy and lossy-to-lossless image compression. In the last two decades, a myriad of coding systems (e.g., [1]–[21]) and several compression standards [22]–[24] have adopted such technology in the core of their coding schemes. Arguably, the extent of bitplane image coding is due to two main factors: 1) the use of the binary representation of coefficients, which is very convenient for current hardware architectures; and 2) theoretical work that enhanced coding efficiency and furthered features of such coding systems.

The main insight behind bitplane coding is to refine the distortion of a transformed image transmitting each coefficient bit by bit. More precisely, let $[b_{M-1}, b_{M-2}, ..., b_1, b_0]$, $b_i \in \{0, 1\}$ be the binary representation for an integer u that corresponds to the magnitude of the index obtained by quantizing a coefficient w, with M denoting a sufficient number of bits to represent all coefficients. Let $d \in \{+, -\}$ be the sign of that coefficient. Bitplane coding strategies generally define bitplane j as the collection of bits b_j from all coefficients, and encode coefficients 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 significant bit of the coefficient. The sign of the coefficient is coded immediately after its significant bit. The remaining bits $b_r, r < s$ are called refinement bits.

When the codestream is truncated, all bits of a coefficient may not be available. This may arise naturally during the transmission of a compressed image. Thus, the decoder carries out an approximate dequantization operation consisting of assigning a reconstruction value $\hat{w}$ that lies somewhere in the indexed quantization interval. If $b_{j'}$ denotes the last available bit of u, in the case of a deadzone scalar quantization with step size λ, the reconstruction procedure is expressed as

$$\hat{w} = \begin{cases} 0 & \text{if } j' > s \\ \text{sign}(w)\ (\hat{u} + \delta) \cdot \lambda 2^{j'} & \text{otherwise} \end{cases}, \tag{1}$$

where $\hat{u} = [b_{M-1}, b_{M-2}, ..., b_{j'}]$, and $\delta \in [0, 1)$ adjusts the reconstruction value $\hat{w}$ within the indexed quantization interval $[\lambda 2^{j'}, \lambda 2^{j'+1})$. Typically, $\delta = 1/2$. Inherently, such a coding procedure generates a quality progressive codestream that improves the quality of the decoded image successively as more data are decoded.

In general, image coding systems can be evaluated considering three main characteristics: coding performance, computational complexity, and scalability. Coding performance refers to the system's compression efficiency, whereas computational complexity refers to its throughput for fixed computational resources. Scalability is the ability to identify and decode some aspect of the image without reading the entire compressed codestream. Scalability is desirable to enable the partial transmission and/or decoding of different components and/or spatial regions of an image at different levels of resolution and/or degrees of quality.

Unfortunately, some of these characteristics are opposed. High coding performance commonly requires computationally intensive procedures, for instance. Therefore, coding systems typically represent a trade-off. Without aiming to be exhaustive, we roughly classify bitplane image coding systems in four groups that include codecs by similar technology[1].

The *first group* of codecs [1], [2] includes those that introduced bitplane coding in the context of trees of coefficients from the wavelet transform [25]. The coding engine of those codecs uses hierarchically partitioned sets of coefficients among wavelet subbands. The significance state of coefficients is coded via a tree, which reduces the number of emitted

Dr. Francesc Aulí-Llinàs is with the Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, Spain (phone: +34 935813571; fax: +34 935814477; e-mail: fauli@deic.uab.es). Prof. Michael W. Marcellin is with the Department of Electrical and Computer Engineering, University of Arizona, USA (e-mail: marcellin@ece.arizona.edu). He is currently visiting professor and Marie Curie Fellow at the Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, Spain. This work has been partially supported by the European Union, by the Spanish Government (MICINN), by FEDER, and by the Catalan Government, under Grants RYC-2010-05671, FP7-PEOPLE-2009-IIF-250420, TIN2009-14426-C02-01, and 2009-SGR-1224.

[1]We note that some works may not be chronologically ordered and may belong to more than one group. The sole purpose of this taxonomy is to synthesize and describe the main advances produced in the bitplane coding literature throughout the years.

symbols. After significance coding, the engine carries out a second coding pass in each bitplane to emit the refinement bits of those coefficients that became significant in previous bitplanes. This scheme has low computational complexity and achieves good coding performance. Furthermore, it generates a quality progressive codestream (a particular form of quality scalability), which was a novel feature that became popular rapidly.

The *second group* of codecs focused on coding performance. Most works [3]–[6] deploy statistical models to evaluate probabilities of emitted symbols. Feeding the emitted bit and its probability to an arithmetic coder reduces the amount of information needed to represent the original message, thereby achieving compression. Another ingenious mechanism is the use of fractional bitplanes. Instead of coding each bitplane using two coding passes (significance and refinement), [7] proposes to fraction bitplanes in four coding passes that finely granulate the distortion emitted in each pass. To code coefficients from the first to the last coding pass in each bitplane produces a codestream with better rate-distortion performance. Some of these codecs [8]–[11] deploy scanning orders based on sorting strategies different from hierarchical trees, which explicitly or implicitly provide resolution scalability. Generally, the improvement on coding performance comes at the expense of higher computational costs.

The technology used by the *third group* of codecs was born during the standardization process of the image compression standard JPEG2000 [26]. Until then, coding systems attained high coding performance at an acceptable degree of computational complexity, and supported scalability by quality, by resolution, and by component. To fully gain interactive transmission, spatial scalability is required. It can be brought into the coding system by means of a conceptual division of wavelet subbands in small sets of coefficients, commonly called codeblocks. Codeblocks can be decoded independently and are stored in the codestream in small groups, so spatial regions of the image can be identified in the compressed domain without decoding. This mechanism is introduced in [12] and is implemented in the core of JPEG2000 [23]. Other spatially scalable codecs using different scanning strategies are [13], [14].

A *fourth group* of codecs have recently emerged broadening their goals and providing new features. Among many other capabilities, some of these codecs provide support for the coding of 3D or N-dimensional images [15], [16], resilience to transmission errors [17], [18], hardware-friendly architectures [19], [20], or high throughput [21].

As indicated by many works [7], [12], essential to achieving competitive coding performance are: 1) the scanning order followed by the coding engine; 2) the number of coding passes deployed per bitplane; and, 3) the model to estimate the probability mass function (pmf) of emitted symbols. Notwithstanding the numerous mechanisms and techniques available nowadays, we must bear in mind that complex scanning orders, too many coding passes, or complex probability models may increase the computational complexity of the coder unnecessarily. In spite of the great interest and the considerable extent and diversity of codecs based on bitplane coding technology, to the best of our knowledge there is no work that evaluates within a unified framework which is the right amount of complexity required for the coding engine to achieve competitive coding performance without sacrificing features in modern codecs.

The first part of this paper uses recent advances in context modeling and distortion estimation to appraise modern bitplane coding engines from different perspectives. The aim of this study is to disclose which are the particularities of the scanning order that make it efficient. Our assessment begins with the premise that coding systems should provide the same features as those of JPEG2000, which are considered excellent. Consequently, the framework of JPEG2000 is borrowed and its bitplane coding engine is substituted by other strategies. Two probability models are used to estimate the pmf of emitted symbols: the classic context-adaptive approach [6], [12], [27], [28], and the recently proposed local average approach [29].

The knowledge acquired in the aforementioned analysis can aid the design of new bitplane coders. To illustrate this point, the second contribution of this paper is a scanning order for bitplane coding engines that is conceptually simple and has low computational costs. The proposed scanning order provides the same features as JPEG2000 without sacrificing coding performance. The third contribution of this work is the introduction of the proposed scanning order into a 3D image coding scheme, which demonstrates that –in that context– the proposed strategy can speedup the JPEG2000 bitplane coder by a factor of 1.6. Key to reducing complexity without sacrificing performance is to carry out the scanning procedure only once per bitplane whilst producing many points where the codestream generated for each codeblock can be truncated.

The paper is structured as follows: Section II describes necessary theory; Section III evaluates four classic scanning orders; and Section IV introduces a simple yet efficient scanning order for bitplane coding. Section V assesses coding performance and throughput, and the last section concludes by summarizing this work.

II. Rate-distortion theory

A. Distortion estimators

Important to image codecs is the possibility to quantify the distortion produced when partial coefficients are transmitted. This is useful, for instance, to optimize the construction of the codestream, estimate the distortion after coding, or transcode already compressed images. When the distortion metric is Mean Squared Error (MSE), the distortion can be determined as the square of the difference between the original and the quantized coefficient, i.e., $G \cdot (w - \hat{w})^2$, with G denoting the energy gain factor of the subband to which the coefficient belongs. When computational resources are constrained, or when the original image is not available, distortion can be estimated rather than actually computed. One strategy is to estimate decreases in squared error at quantization intervals corresponding to bitplanes, so that the coder can determine the distortion decrease expected for significant and refinement bits. Such an approach was first proposed in [8], and was further developed in [30]. Its main advantages are that expected error decreases can be computed beforehand so that the application can use them through computationally efficient lookup tables, and that such estimators are computed without the need of the original image. Thus they are available at both the coder and decoder.

For notational simplicity, we assume that coefficients are normalized by the quantization step size λ in what follows. Let us first consider the squared error decrease produced by those coefficients that become significant at bitplane j^*. If v denotes the magnitude of such a coefficient, then $2^{j^*} \leq v < 2^{j^*+1}$. Let $p(v)$ denote the marginal probability density function (pdf) for v. The squared error decrease that can be expected when such coefficients are encoded to bitplane j^* is then determined as

$$\triangle D_{sig}^{j^*} = \int_{2^{j^*}}^{2^{j^*+1}} p(v) \left[v^2 - \left(v - (2^{j^*} + \delta \cdot 2^{j^*}) \right)^2 \right] \; dv. \tag{2}$$

Distortion decreases for refinement coding $\triangle D_{ref}^{j}, j < j^*$ can be derived similarly.

When assuming a uniform probability distribution within the significance interval, $p(v) = 1/2^{j^*}$ and the δ value that maximizes the average distortion decrease is at the center of the interval, i.e., $\delta = 1/2$. The approach in [30] shows that distortion estimators $\triangle D_{sig}^{j^*}$ and $\triangle D_{ref}^{j}$ can be more finely adjusted by determining $p(v)$ and δ using a model for densities of coefficients in wavelet subbands. That model approximates the actual coefficients' distribution within quantization intervals, leading to a more precise estimation of distortion decreases. Due to page constraints, we refer the reader to [30] for a detailed description, and we adopt $\triangle D_{sig}^{j^*}$ and $\triangle D_{ref}^{j}$ as determined in that work.

B. Probability models

Commonly, image codecs exploit high-order statistics of symbols emitted by the bitplane coding engine to achieve compression. The most popular approach to exploit statistical redundancy is context-adaptive arithmetic coding [6], [12], [27], [28]. The main idea behind it is to adaptively adjust the probabilities of emitted bits depending on the context of the coefficient. Let $w^n, 1 \leq n \leq N$ denote N neighbors of w, let $\Phi(w, j)$ denote the significance state of w in bitplane j according to

$$\Phi(w, j) = \begin{cases} 0 & \text{if } j > s \\ 1 & \text{otherwise} \end{cases} , \tag{3}$$

ant let $\phi(w, j)$ represent the sign of w at bitplane j as

$$\phi(w, j) = \begin{cases} \text{unknown} & \text{if } j > s \\ + & \text{if } j \leq s \text{ and } w > 0 \\ - & \text{if } j \leq s \text{ and } w < 0 \end{cases} . \tag{4}$$

In general, contexts are selected as some function of w^n. Often, this function considers $\{\Phi(w^n, j)\}$ for significance and refinement coding, and $\{\phi(w^n, j)\}$ for sign coding. The context is passed to the arithmetic coder, which (if adaptive) adjusts probabilities as more symbols are coded. This mechanism approximates the pmf of emitted symbols given their context. The pmf of the currently emitted symbol b_j 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.

Mechanisms that exploit coefficient magnitudes, rather than significance states, include the Tarp-filter [31]–[33] and the local average approach [29], [34]. The analysis carried out in the next section employs the local average approach due to its convenience to describe some aspects of the coding procedure. The local average of wavelet coefficients is defined as the arithmetic mean of the neighbors' magnitudes according to

$$\varphi = \frac{1}{N} \sum_{n=1}^{N} |w^n| \; . \tag{5}$$

In our previous work [29], φ is determined using the four immediate neighbors that are above, below, to the right, and left of the current coefficient, though other configurations may obtain similar results. The local average-based probability model characterizes the signal produced by a wavelet transform using the marginal pdf of coefficients, denoted as $g(w)$, and the conditional pdf of φ given w, denoted as $h(\varphi \mid w)$. Probabilities for significant bits are determined according to

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

$$\frac{P(w < 2^j \mid \varphi)}{P(w < 2^{j+1} \mid \varphi)} = \frac{\displaystyle\int_0^{2^j} g(w) \cdot h(\varphi \mid w) \; dw}{\displaystyle\int_0^{2^{j+1}} g(w) \cdot h(\varphi \mid w) \; dw} \; . \tag{6}$$

Probabilities for refinement bits emitted at bitplane j for coefficients that became significant at bitplane j^* are determined similarly and denoted as $P_{ref}(b_j = 0 \mid \varphi), j < j^*$.

Evidently, the actual magnitude of wavelet coefficients is not available for the decoder until all bitplanes are transmitted, so the coding process computes φ using the magnitude of partially transmitted coefficients and an estimate of insignificant coefficients. It is then denoted as $\hat{\varphi}$. Again, we refer the reader to [29] for an extended description, and we adopt probabilities $P'_{sig}(b_j = 0 \mid \hat{\varphi})$ and $P'_{ref}(b_j = 0 \mid \hat{\varphi})$ as they are determined in that work.

Sign coding is not tackled in our previous work. Analysis carried out in the next section compares the coding performance achieved by a context-adaptive probability model to that achieved by a local average-based probability model. To do so, the local average probability model employed in the next section performs sign coding by estimating $P_{sign}(d)$ using statistics of a corpus and considering $\{\phi(w^n, j)\}$ for four immediate neighbors of w. Although its performance is slightly inferior to that achieved with context-adaptive coding, the sole intention of this comparison is to illustrate the behavior of these models when used with different scanning orders.

III. Classic coding strategies

A typical coding system for JPEG2000 [26] is constituted by three main stages: sample data transformation, sample data coding, and codestream re-organization. The main operation carried out in the first stage is the application of the wavelet transform. Then, the image is logically partitioned in code-blocks, that are independently coded in the second stage. The third stage 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.

A. Optimal scanning order

As stated previously, this study borrows the framework of JPEG2000 and substitutes the sample data coding stage by other strategies. The first question that arises when contemplating a bitplane coding engine is in which order should bits be transmitted. Let us begin devising a scanning order that produces an optimal quality embedded codestream, i.e., a codestream that truncated at any bitrate guarantees that the quality of the decoded image can not be improved more. From a rate-distortion optimization perspective, that scanning order should always select the next coefficient to visit so that it maximizes the ratio between the decrease in image distortion and the increase in codestream length [7], [8]. More precisely, let $\triangle\mathcal{D}(w)$ be the decrease produced in image distortion when coding one more bit of coefficient w, and let $\triangle\mathcal{L}(w)$ be the increase in codestream length produced when such bit is coded. The distortion-length slope of coefficient w for the next to-be-coded bit is defined as $\mathcal{S}(w) = \triangle\mathcal{D}(w)/\triangle\mathcal{L}(w)$. The scanning sequence that visits every next coefficient selecting that $w^\star$ with highest $\mathcal{S}(w^\star)$ produces an optimal codestream.

From a practical perspective, however, scanning orders must take into account that the sequence of visited coefficients must be followed by coder and decoder in strict order. This renders impractical the above approach since neither $\triangle\mathcal{D}(w)$ nor $\triangle\mathcal{L}(w)$ are available for the decoder, and to explicitly transmit the order would penalize coding efficiency significantly. [8] proposed the use of distortion estimators and probability estimates for emitted symbols to approximate $\triangle\mathcal{D}(w)$ and $\triangle\mathcal{L}(w)$ in the decoder. That approach assumes only uniform probability distributions. It also employs parent-child relations among coefficients belonging to different resolution levels, which makes it impractical herein. Our previous work [35] implemented the scanning strategy proposed in [8] using highly precise distortion estimators [30] together with the local average-based probability model [29] to allow its use in codeblocks. Here, [35] is extended to allow the use of either the local average-based probability model (LAV), or the context-adaptive probability model (CAD). Specifically, $\triangle\mathcal{D}(w)$ is estimated as

$$\triangle\mathcal{D}'(w) = \begin{cases} \triangle D^j_{sig} \cdot \left(1 - P'_{sig}(b_j = 0)\right) & \text{if } j \geq s \\ \triangle D^j_{ref} & \text{otherwise} \end{cases} , \quad (7)$$

where $P'_{sig}(b_j = 0)$ is the estimate of $P_{sig}(b_j = 0)$ conditioned on either the local average $\hat{\varphi}$ as described above (LAV) or the adaptive contexts employed in JPEG2000 (CAD). The first line of (7) can be understood as the conditioned expected distortion decrease (given the coefficient is significant) multiplied by the probability that the coefficient becomes significant.

On the other hand, knowing the probability for the coded symbol in conjunction with the use of arithmetic coding aids in the determination of the $\triangle\mathcal{L}(w)$ estimate. Arithmetic coders are commonly able to approach entropy. Therefore, the cost of coding binary symbols can be estimated by the symbol entropy, which in turn can be estimated using the conditional probabilities employed by the arithmetic coder according to

$$\begin{aligned} \triangle L^j_{sig} &= \mathcal{H}\left(P'_{sig}(b_j = 0)\right) \\ &+ \left(1 - P'_{sig}(b_j = 0)\right) \cdot \mathcal{H}\left(P'_{sign}(d)\right) \end{aligned} \quad (8)$$

and

$$\triangle L^j_{ref} = \mathcal{H}\left(P'_{ref}(b_j = 0)\right) , \quad (9)$$

for significance and refinement coding, respectively. $\mathcal{H}(\cdot)$ in the expressions above denotes the binary entropy function. The cost of coding the sign is accounted for in the second line of Equation (8) since the sign is emitted just after the coefficient is found significant. $\triangle\mathcal{L}(w)$ is then estimated as

$$\triangle\mathcal{L}'(w) = \begin{cases} \triangle L^j_{sig} & \text{if } j \geq s \\ \triangle L^j_{ref} & \text{otherwise} \end{cases} . \quad (10)$$

Employing the above development, the encoder can seek the next coefficient to be coded as that one that has the highest estimated distortion-length slope, i.e., that $w^\star$ with highest

$$\mathcal{S}'(w^\star) = \frac{\triangle\mathcal{D}'(w^\star)}{\triangle\mathcal{L}'(w^\star)} = \begin{cases} \dfrac{\triangle D^j_{sig} \cdot \left(1 - P'_{sig}(b_j = 0)\right)}{\triangle L^j_{sig}} & \text{if } j \geq s \\ \dfrac{\triangle D^j_{ref}}{\triangle L^j_{ref}} & \text{otherwise} \end{cases} . \quad (11)$$

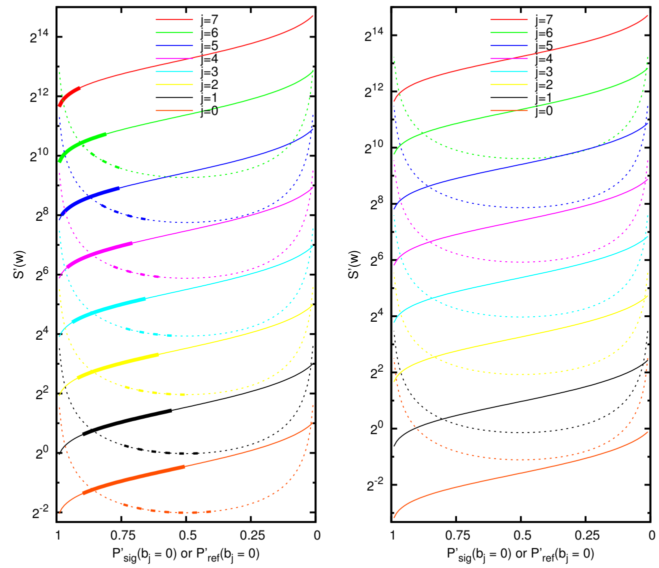
The use of $\mathcal{S}'(w)$ allows that the visiting sequence is adapted as more data are coded, without restricting which bits (significant or refinement) are transmitted first. Our experience suggests that $\mathcal{S}'(w)$ is a sufficiently good estimator of $\mathcal{S}(w)$, especially when using LAV. Although at high bitplanes the local average used in the computation of $\mathcal{S}'(w)$ is approximated roughly, experimental evidence indicates that at medium bitplanes, the error in $\hat{\varphi}$ is less than 50% of φ, on average. At the lowest bitplanes, this error is generally below 8%. The impact of this estimation error is discussed in Section III-B.

$\mathcal{S}'(w)$ determines the scanning order of the optimal strategy, so an analysis of $\mathcal{S}'(w)$ is of interest. As seen in Equations (7)-(11), $\mathcal{S}'(w)$ utilizes distortion estimators $\triangle D^j_{sig}$ and $\triangle D^j_{ref}$, which are constant in each bitplane, and $P'_{sig}(b_j = 0)$, $P'_{ref}(b_j = 0)$, and $P'_{sign}(d)$, which are variable. Figure 1(a) depicts $\mathcal{S}'(w)$ with respect to $P'_{sig}(b_j = 0)$ and $P'_{ref}(b_j = 0)$[2] for irreversible transforms. Let us begin by describing $\mathcal{S}'(w)$ for significance coding. The more skewed $P'_{sig}(b_j = 0)$ toward 0, the larger the $\mathcal{S}'(w)$ since these symbols can be encoded using few bits and the decrease in distortion is very large. Contrarily, the more skewed the probability $P'_{sig}(b_j = 0)$ toward 1, the smaller the $\mathcal{S}'(w)$. A similar shape is achieved for all bitplanes. On the other hand, $\mathcal{S}'(w)$ for refinement coding is larger as $P'_{ref}(b_j = 0)$ is more skewed, and is symmetric around $P'_{ref}(b_j = 0) = 0.5$ since 0 and 1 refinement bits have equal expected distortion decrease. Note that the analysis of Figure 1(a) suggests that some refinement bits should be coded before some significant bits within each bitplane, or even before some significant bits of higher bitplanes. This seems to question the separation of bits into bitplanes.

Even though Figure 1(a) depicts $\mathcal{S}'(w)$ for all possible values of $P'_{sig}(b_j = 0)$ and $P'_{ref}(b_j = 0)$, these probabilities typically take values above 0.5. Specifically, the higher the bitplane, the more biased the probabilities are toward 1, for both significance and refinement coding. This is intuitively explained considering the density of coefficients within wavelet subbands [29], which is nominally two-sided exponential (Laplacian). For significance coding this implies that the higher the bitplane, the fewer significant coefficients are found, and that the first refinement bit of coefficients has a higher probability to be 0 than 1.

To indicate this, $\mathcal{S}'(w)$ is emphasized in Figure 1(a) with a thicker line in the regions where the probabilities of emitted bits at that bitplane are relevant. Otherwise stated, the thicker lines represent common values for $\mathcal{S}'(w)$ at those bitplanes. Considering only these relevant probabilities, Figure 1(a) suggests that ordering the emission of bits in bitplanes is adequate from a rate-distortion optimization point of view. Within each bitplane, significant bits should be transmitted before refinement bits. Only when $P'_{sig}(b_j = 0)$ and $P'_{ref}(b_j = 0)$ are very high, typically at high bitplanes, refinement bits may be transmitted before significant bits.

Figure 1(b) depicts the same analysis as above but using distortion estimators determined for reversible transforms [30]. Similar conclusions are drawn except at the lowest bitplane, in which refinement bits should be transmitted before significant bits. This is triggered by the reconstruction quantization interval for significance coding at bitplane $j = 0$, which is $[1, 1]$ for reversible transforms instead of the typical $[1, 2)$ available for irreversible transforms. This causes reversible transforms to reconstruct significant coefficients at bitplane $j = 0$ as 1, whereas irreversible transforms reconstruct coefficients as 1.5 (or similar). This produces a smaller decrease in distortion for reversible transforms than for irreversible ones [30]. Furthermore, refined bits decrease the squared error in bitplane $j = 0$ by 0.5 for reversible transforms and only 0.25 for irreversible ones, on average, which accentuates more the difference between significance and refinement coding at bitplane $j = 0$. Experimental evidence suggests that to reverse the order of significance and refinement bits in the lowest bitplane improves coding performance slightly.



(a) irreversible transform (b) reversible transform

Fig. 1: Evaluation of $\mathcal{S}'(w)$ depending on the probability of emitted symbols. Both figures depict $\mathcal{S}'(w)$ distinguishing between significance coding (continuous line), and refinement coding (dotted line) at each bitplane.

B. Low complexity scanning orders

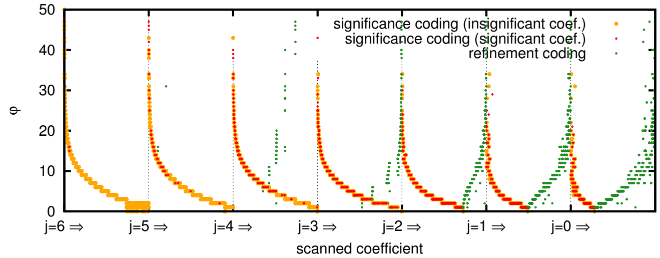
Figure 1 justifies the transmission of bits in bitplanes, but it indicates little with regard to the sequence of visited coefficients. As hinted in [29], $\hat{\varphi}$ is closely related to $P'_{sig}(b_j = 0)$ and $P'_{ref}(b_j = 0)$, which in turn are closely related to $\mathcal{S}'(w)$ and, by extension, to the sequence of visited coefficients. $\hat{\varphi}$ will serve to illustrate and assess the efficacy of scanning orders followed by different strategies.

The first scanning order that we assess is the optimal one described above using the LAV probability model. We begin by using φ instead of $\hat{\varphi}$ when selecting the next coefficient to visit. The use of φ is not possible in an actual system, but provides and idea of the scanning order that the scheme strives to achieve in theory. Figure 2(a) depicts visited coefficients when coding one representative codeblock.[3] The horizontal axis on this graph depicts scanned coefficients. The vertical black lines represent the beginning of bitplanes. The vertical axis depicts φ for the corresponding coefficient. Significance coding is depicted in yellow (and red) for coefficients that do not become (and do become) significant at that bitplane, respectively. Refinement coding is depicted in green.

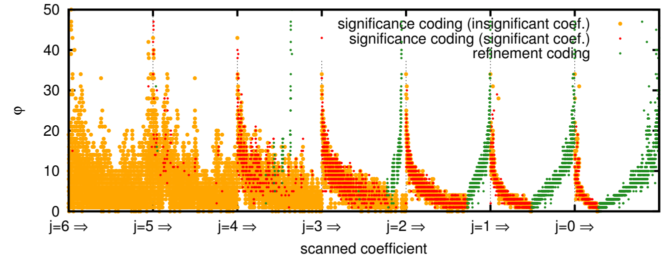
The scanning order followed by this theoretical-optimal strategy visits first those insignificant coefficients with highest φ, since those are most likely to become significant and

[2] $P'_{sign}(d)$ is set to 0.5 to simplify the analysis of Figure 1. Allowing $P'_{sign}(d)$ to vary does not change the results significantly.

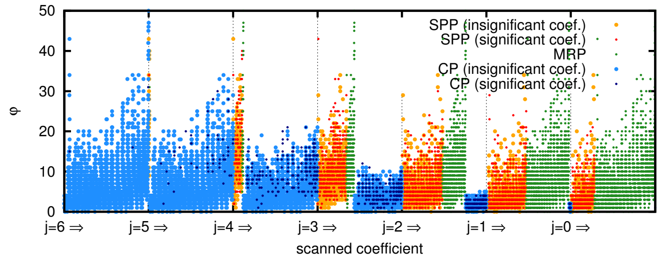
[3] Demonstrations of the scanning order strategies depicted in Figure 2 can be found at http://www.deic.uab.es/~francesc



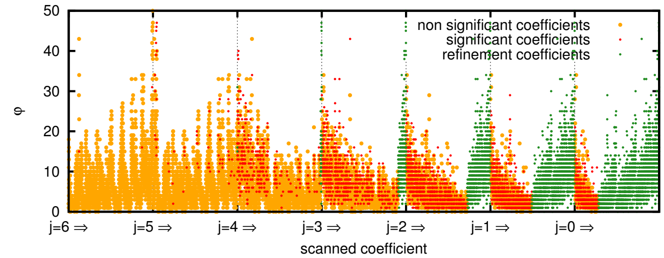
(a) optimal using φ (LAV)



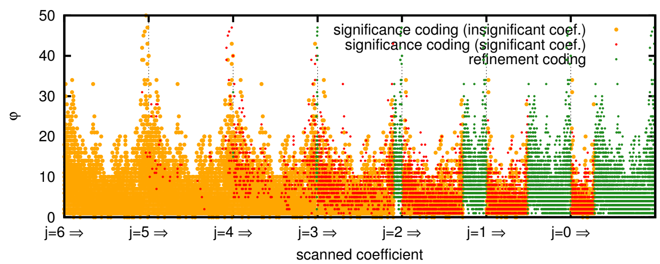
(b) optimal using $\hat{\varphi}$ (LAV)



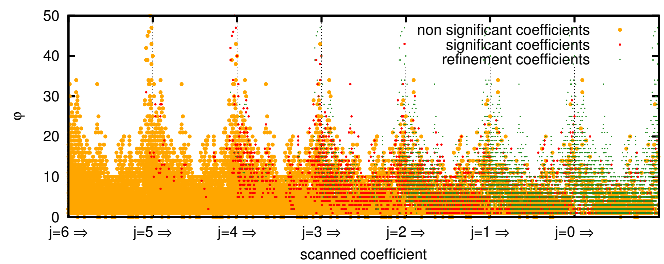
(c) JPEG2000 (LAV or CAD)



(d) hybrid



(e) two passes (LAV or CAD)



(f) single pass (LAV or CAD)

Fig. 2: Evaluation of different scanning orders. The horizontal axis of these figures represents the order of visited coefficients in each bitplane, whereas the vertical axis depicts φ of the visited coefficient. Data belong to one codeblock (size 64×64) of the High-vertical Low-horizontal frequencies subband of the second decomposition level (subband HL_2) produced when the irreversible 9/7 wavelet transform is applied to the "Portrait" image of the ISO 12640-1 corpus.

provide the largest distortion decreases. Obviously, distortion decreases are not known in advance. Even though the model sometimes fails (note that yellow coefficients provide no decrease), on average it achieves the maximum efficiency. Significant coefficients depicted in red generally provide larger distortion decreases than refined coefficients, which are depicted in green. At the highest bitplanes, refinement coding is carried out in the middle of the scanning sequence, whereas at the lowest bitplanes refinement coding is carried out after significance coding, which corresponds with the analysis of Figure 1. Contrarily to significance coding, refinement coding visits first those coefficients with lowest φ since $P_{ref}(b_j)$ is more biased for these coefficients than for coefficients with higher φ [29].

Figure 2(b) depicts the results when the scanning order uses $\hat{\varphi}$ instead of φ when selecting the next coefficient to visit. In the vertical axis of this and following figures, φ (and not $\hat{\varphi}$) is still plotted to provide an idea of the efficacy of the visiting sequence compared to that reported in Figure 2(a). Mostly at high bitplanes, the visiting sequence of Figure 2(b) is not as effective as in the previous case. At medium and low bitplanes the magnitude of most coefficients is estimated with reasonable precision, so the scanning pattern becomes closer to that depicted in Figure 2(a).

Though being very effective, the optimal scanning order requires computationally intensive procedures. Practical strategies should try to approximate the visiting sequence followed by the optimal one but using simple operations to select the next coefficient to visit. The scanning order of JPEG2000, for example, has three subbitplane coding passes called Significance Propagation Pass (SPP), Magnitude Refinement Pass (MRP), and Cleanup Pass (CP). Each coefficient is scanned only once in each bitplane. SPP visits those insignificant coefficients that have at least one significant neighbor. MRP refines the magnitude of already significant coefficients, and CP visits insignificant coefficients that were not scanned by SPP. The CP coding pass also has a special scanning mode called run mode that codes several insignificant coefficients with a single binary symbol. The visiting sequence is ordered in stripes of coefficients, each containing four rows of coefficients. Coefficients are scanned column-by-column from the top-left to the bottom-right corner in each stripe.

Stripes in JPEG2000 were introduced for hardware-friendly implementations. The division in three fractional bitplanes has its origins in the same roots as the optimal scanning strategy: the better the division in fractional bitplanes, the better the coding performance of the final codestream. In addition, coding passes provide suitable truncation points for the codestream that can be exploited by the rate-distortion optimization process [12].

The sequence of visited coefficients followed by JPEG2000 is depicted in Figure 2(c). To distinguish SPP and CP, coefficients scanned in CP are depicted in blue. Points depicted in light and dark blue represent, respectively, coefficients that do not become and do become significant. The use of three coding passes produces a trend similar to that of Figure 2(a). SPP scans those coefficients with highest φ, whereas coefficients with lowest φ are scanned in CP. MRP is scheduled between SPP and CP, which at high and medium bitplanes obtains a similar behavior to that of Figure 2(a). Since at the lower bitplanes most insignificant coefficients are scanned in SPP, refinement coding is carried out after significance coding, which also coincides with Figure 2(a). Since JPEG2000 uses a deterministic order within each coding pass, a rougher discrimination is produced within each coding pass (especially at the lowest bitplanes), though in general the scanning sequence of JPEG2000 is similar to that followed by the optimal strategy.

To provide a complete assessment of bitplane coding in this context, two more strategies are described: 1) a scanning order with two passes that carries out significance coding first followed by refinement coding; 2) a scanning order with one pass that carries out significance and refinement coding at the same time. Both strategies scan coefficients in raster mode (row by row). Figures 2(e) and 2(f) depict the visiting sequence of these two strategies, which are not well adapted to the data.

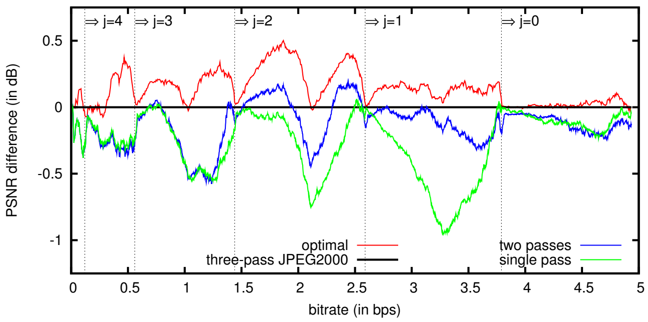
We note that the scanning orders followed by JPEG2000 and the last two strategies are not affected by the probability model used for arithmetic coding (Figures 2(c), 2(e), and 2(f)). Only the optimal scanning order (Figures 2(a), and 2(b)) adapts the visiting sequence depending on the probability model. Figure 2 depicts the optimal scanning strategy only using LAV for illustration purposes.

C. Coding performance evaluation

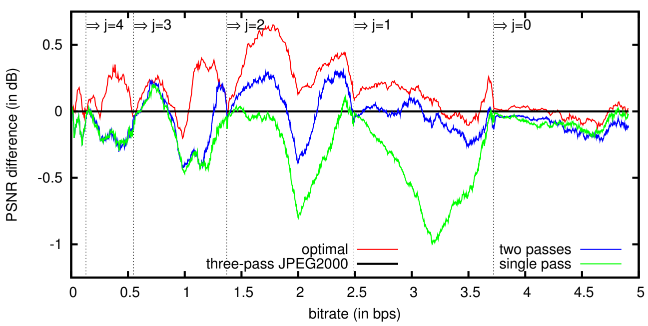
The scanning strategies outlined above have decreasing computational complexity in the order of their description. The next evaluation assesses their compression efficiency to ascertain whether their complexity is rewarded with coding performance or not. First, coding performance is assessed individually for codeblocks. Figure 3(a) depicts the PSNR difference between the 3-pass strategy of JPEG2000 and the other strategies. The performance of the three-pass JPEG2000 strategy is depicted as the horizontal (zero) line in this figure, whereas the other strategies are plotted above or below the horizontal line depending on whether they achieve better or worse coding performance, respectively, than three-pass JPEG2000. The PSNR difference is reported after each byte emitted by a bitplane coder. The ends of bitplanes are marked with horizontal lines in Figure 3(a). The practical version of the optimal strategy using $\hat{\varphi}$ attains better coding performance than the strategy of JPEG2000 at almost all bitrates, whereas the strategies that deploy two or one coding passes are generally worse. Figure 3(a) evaluates coding performance when the LAV probability model is used. Figure 3(b) evaluates coding performance for CAD. As noted above, the probability model does not affect the scanning order for three of the four schemes (including the 3-pass JPEG2000 scheme). However, it does affect the arithmetic coding in all cases. Thus, the 3-pass strategy is exactly JPEG2000 in Figure 3(b), but not in Figure 3(a).

Although achieving different results byte-by-byte, it is worth noting that the coding performance achieved by all strategies is virtually identical at the end of each bitplane, resulting in the quasi-periodic variations shown in Figure 3. This is consistent with [36], [37], which also indicate that bitplane boundaries are near-optimal truncation points from a rate-distortion point of view. When post-compression rate-distortion optimization is used to form the final codestream, only the coding performance achieved at the truncation points of the codeblock codestreams is considered. If codeblock codestreams could only be truncated at the end of bitplanes, all strategies would attain virtually the same coding performance for the whole image. However, truncation is typically allowed at the end of coding passes, which provides more segments to the rate-distortion optimization process to better optimize the quality of the final image. Figure 3(c) depicts distortion achieved byte-by-byte when coding bitplane $j = 1$ of the same codeblock as above. Distortion decreases roughly linearly from the end of one bitplane to the next when using the single pass strategy. When two passes are employed, the distortion decreases roughly linearly within each coding pass. The JPEG2000 strategy (not shown to avoid cluttering the figure) yields three linearly decreasing segments in the rate distortion curve. Evidently, two or three such segments are sufficient to provide a close approximation to the optimal strategy, which decreases roughly exponentially throughout the bitplane. Coding performance achieved at the end of coding passes (marked with circles) commonly coincides with that achieved by the optimal strategy. Similar results are obtained for other bitplanes and codeblocks.

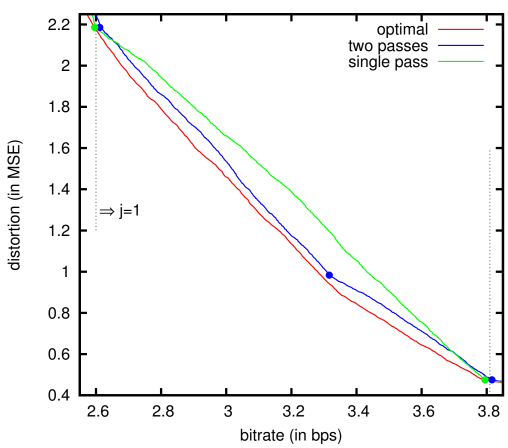
Figure 4 evaluates coding performance obtained for an entire image when using different codeblock sizes. In this case, only the CAD probability model is used, though results are similar for LAV. Truncation is allowed at the end of coding passes for the three-pass JPEG2000, two passes, and single pass strategies, and every 4 bytes for the optimal strategy. Clearly, the single pass scanning order that provides one coding pass per bitplane does not achieve competitive coding performance in any case. When the codeblock size is 16×16 or 32×32, results seem to suggest that two coding passes per bitplane are enough to achieve good coding performance. The performance of the optimal strategy is slightly better than that of JPEG2000, except for codeblocks of 16×16, which is degraded due to header overhead caused by the generation of too many truncation points.



(a) LAV probability model

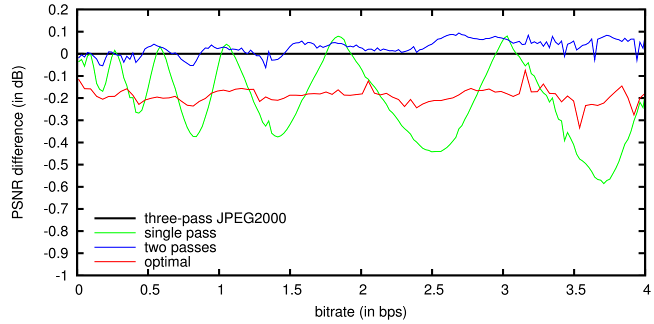


(b) CAD probability model

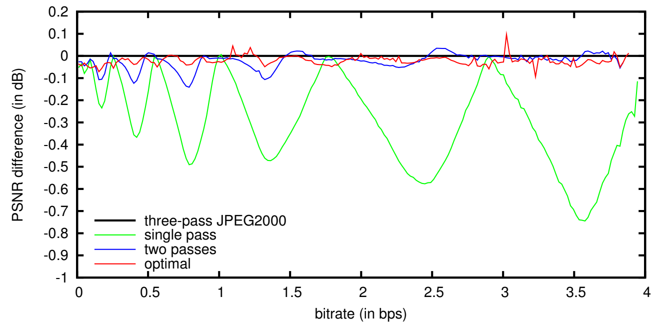


(c) detail of one bitplane (with LAV)

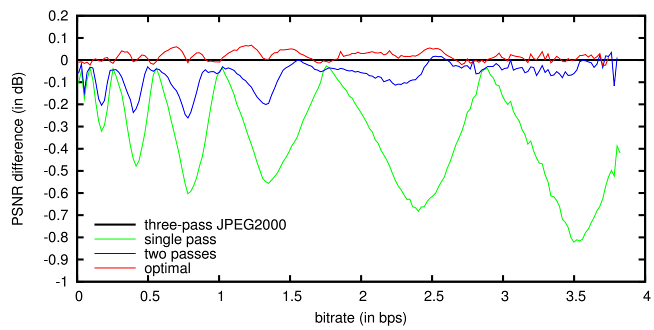
Fig. 3: Evaluation of the coding performance achieved by different scanning strategies when coding the same codeblock as in Figure 2.



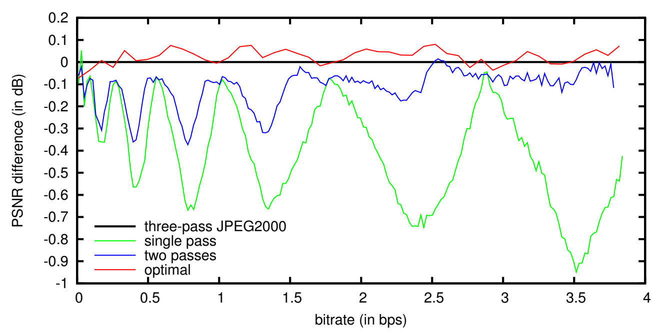
(a) codeblock size 16×16



(b) codeblock size 32×32



(c) codeblock size 64×64



(d) codeblock size 128×128

Fig. 4: Evaluation of the coding performance achieved by different scanning strategies when using different codeblock sizes, for the "Portrait" image. The irreversible 9/7 wavelet transform, and the CAD probability model are used.

IV. Hybrid Coding Strategy

A. General description

Some observations drawn from the previous study are:

- Significant bits should be emitted before refinement bits. Only at high bitplanes is it profitable for refinement to be interleaved with significant bits. Nonetheless, we note that few refinement bits are emitted at high bitplanes.
- The scanning order followed in significance coding should visit coefficients from the highest to the lowest φ (or $\hat{\varphi}$).
- The scanning order followed in refinement coding should visit coefficients from the lowest to the highest φ (or $\hat{\varphi}$).
- The size of the codeblock dictates the complexity of the scanning order required to obtain competitive coding performance. Simple scanning strategies serve for small codeblocks, whereas large codeblocks profit from elaborate scanning orders.
- Small codeblocks (up to 32×32) require at least two truncation points per bitplane, whereas larger codeblocks

require three or more truncation points.

- The main advantage of elaborate scanning orders is in providing more truncation points.
- Truncation points should lie on the convex hull of the rate-distortion function of the codeblock.
- Techniques that group the emission of several insignificant coefficients with a single symbol (e.g., the run mode of JPEG2000) do not produce a significant increase on coding performance compared to the use of precise probability models[4].
- Probability models have only minor influence on the efficacy of the scanning order, although they may impact the computational complexity of the coding engine.

Considering the points drawn from the previous analysis, the purpose of this section is to devise a scanning strategy with low computational complexity and competitive coding performance. The more complicated approaches to do so are perhaps the generation of more or less truncation points depending on the size of the codeblock, and to use a visiting sequence that depends on φ or other similar indicator. On the other hand, throughput assessments detailed in Section V point to a strategy that "evaluates" coefficients only once per bitplane. By this, we mean that the proposed strategy should avoid the use of context-based subbitplane coding passes (e.g., [7], [12], [23]), since a straightforward implementation of such scanning orders visits each coefficient in every coding pass in order to decide whether it is to be coded in that pass or not.

Our main insight to devise such a scanning order is to conceptually divide codeblocks into small sets of coefficients called cells. Cells provide a cost-effective representation of the codeblock. Our implementation uses cells of 8×8 coefficients, though other configurations may achieve similar results.

Most coding systems that employ bitplane coding require a step that computes the number of bitplanes M, or equivalently, the most significant bitplane (MSBP), denoted as $M-1$. The use of cells requires that the MSBP be calculated for each cell c. These quantities are denoted by $m[c]$. Clearly then, $M-1$ is the maximum $m[c]$ over all cells c in the codeblock to be encoded. This step has negligible computational costs. Cells are only used to code significant bits, which are emitted before refinement bits in each bitplane. As explained later, this can be accomplished easily with the aid of a single list containing locations of significant coefficients. From bitplane $M-1$ down to $m^\diamond+1$ (see below), the coder emits a binary symbol informing the decoder whether $m[c]$ is reached for cell c or not. This step does not enhance coding performance, though it reduces computational costs. Evidently, this step is not required for cell c when the current bitplane, say $j^\diamond$, is $j^\diamond < m[c]$. Indeed, this step is only carried out until bitplane $m^\diamond$ is reached, which can be in the range $m^\diamond \in \big[\max\{m[c]\}, \min\{m[c]\}\big]$. If $m^\diamond \neq \min\{m[c]\}$, there will be at least one bitplane of at least one cell for which a zero is coded at every location in that cell. In our implementation $m^\diamond$ is chosen for a codeblock as the most numerous MSBP (the mode of $m[c]$) of that codeblock. The value of $m^\diamond$ is transmitted at the beginning of coding.

[4]Though this point is not directly extracted from the analysis carried out in Section III, our experience indicates so.

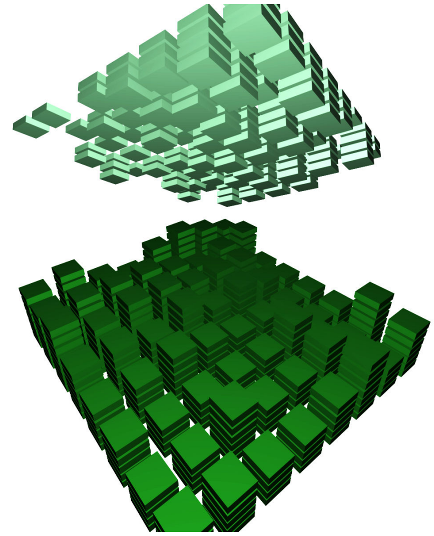


Fig. 5: Representation of the conceptual division of one codeblock into cells. Data belong to a codeblock of size 64×64 in subband HL_2 of the "Portrait" image.

TABLE I: Percentage of bitplanes in each cell skipped by the proposed algorithm. Results are reported on average per each resolution level (5 levels of irreversible 9/7 wavelet transform are used). Recall that the largest resolution level contains 75% of image coefficients, the next one 18.75%, and so on. Results are reported for the ISO 12640-1 corpus (images are 8 bit, gray-scale, size 2560×2048).

image	resolution level					
	0 (LL subband)	1	2	3	4	5
"Portrait"	4	10	22	25	27	30
"Cafeteria"	0	8	9	14	16	23
"Fruit Basket"	0	7	26	29	25	28
"Wine and Tableware"	8	8	29	35	31	30
"Bicycle"	0	10	14	19	27	35
"Orchid"	0	22	37	41	37	36
"Musicians"	4	8	16	20	23	23
"Candle"	4	8	12	13	14	22
AVERAGE	*2*	*10*	*21*	*24*	*25*	*28*

Figure 5 illustrates the conceptual division of one representative codeblock into cells. In this figure, each building (i.e., each pile of boxes in the same position) represents one cell, and each floor of the building (i.e., each box) represents one bitplane. Buildings are split into two parts: the higher part (in light green) represents those bitplanes that do not contain any significant coefficient in that cell; the lower part (in dark green) represents those bitplanes that contain at least one significant coefficient. The main advantage of the division of the codeblock into cells is that most bitplanes in the part above can be completely avoided by the scanning process. To demonstrate this fact, Table I shows the percentage of bitplanes that are *not* visited by the scanning order thanks to the use of cells, on average for each resolution level. On average, more than 25% of the bitplanes in each cell can be avoided.

At each bitplane, cells are sorted in list $\mathcal{G}_{cells}$ by the number of coefficients that became significant in previous bitplanes, say $q[c]$. Cells are visited following this order. That is, the cell that contains the largest number of significant coefficients is visited first. This may vary bitplane-by-bitplane. The intention of this sorting process is to first visit coefficients that are surrounded by many significant neighbors, and thus with (typically) high values of φ. Within each cell, coefficients are visited in raster mode. Allowing the truncation of the codestream at the end of the coding of each cell (or after a fixed number of cells) produces codestream segments from generally higher to lower distortion-length slopes in that bitplane. We note that scanning orders with fractional bitplanes provide truncation points at the end of each coding pass [7], [12], [23]. Small codeblocks have few cells, so few truncation points are provided in each bitplane. Large codeblocks contain many cells, providing many truncation points. Thus, cells provide more or less truncation points depending on the size of the codeblock. The sorting procedure has very low computational complexity.

Once a coefficient becomes significant, it is added to the end of a list of coefficients to be refined, referred to as $\mathcal{G}_{refs}$. The use of such a list is not new [2]. Here, its purpose is twofold. On the one hand, it saves computational time, especially at high bitplanes, since it avoids scanning all coefficients of the codeblock when only a few need refinement. On the other hand, it stores coefficients in the order that they become significant and allows visiting coefficients (roughly) from the lowest to the highest φ (as prescribed in Figure 2(a)). This can be seen by noting that φ is related to the magnitude of the coefficient [29], so coefficients that become significant at low bitplanes generally have lower φ than coefficients that become significant at high bitplanes. Thus, it is desirable to visit coefficients starting from the end of list $\mathcal{G}_{refs}$. Experimental evidence indicates that one truncation point for refinement coding is sufficient.

B. Algorithm

Our coding system uses a combination of CAD and LAV to adjust probabilities of emitted symbols, although the proposed scanning order is not influenced by the probability model. This combination of CAD and LAV is chosen here to illustrate the coding performance that can be achieved with this algorithm without impacting computational load. LAV is used for significance and refinement coding, whereas sign coding employs CAD. As it is formulated in [29], LAV implies a slight increment on computational complexity compared to CAD. In this work the computational gap between LAV and CAD for significance and refinement coding is removed through the simplified model described in Appendix A. Sign coding achieves very high coding performance when it is used with the context configuration deployed in JPEG2000, so that configuration is employed here. Symbols and their probabilities, or contexts, are feed to the arithmetic coder MQ [26]. The algorithm proceeds as follows:

Algorithm 1 Hybrid coder

1: emit $m^\diamond$
2: $Q \leftarrow 0$
3: **for** $j^\diamond \leftarrow M-1$ **to** 0 **do**
4:   $/*$ significance coding $*/$
5:   $Q' \leftarrow 0$
6:   **for** $c \leftarrow$ first cell in $\mathcal{G}_{cells}$ **to** last cell in $\mathcal{G}_{cells}$ **do**
7:     **if** $m^\diamond < j^\diamond$ AND $m[c] < j^\diamond$ **then**
8:       emit 0 with CAD(context 0)
9:     **else**
10:       **if** $m^\diamond < j^\diamond$ AND $m[c] = j^\diamond$ **then**
11:         emit 1 with CAD(context 0)
12:       **end if**
13:       **for** $w^* \leftarrow$ first insignificant coefficient in c **to** the last insignificant coefficient in c **do**
14:         emit $b_{j^\diamond}$ with LAV($\hat{\varphi}$)
15:         **if** $b_{j^\diamond} = 1$ **then**
16:           emit d with CAD(JPEG2000 context configuration)
17:           add w^* to $\mathcal{G}_{refs}$
18:           $q[c] \leftarrow q[c] + 1$
19:           $Q' \leftarrow Q' + 1$
20:         **end if**
21:       **end for**
22:     **end if**
23:   **end for**
24:   sort $\mathcal{G}_{cells}$ by $q[\cdot]$
25:   $/*$ refinement coding $*/$
26:   **for** $w^\star \leftarrow$ coefficient Q in $\mathcal{G}_{refs}$ **to** the first coefficient in $\mathcal{G}_{refs}$ **do**
27:     emit $b_{j^\diamond}$ with LAV($\hat{\varphi}$)
28:   **end for**
29:   $Q \leftarrow Q + Q'$
30: **end for**

The algorithm scans coefficients in the loop of lines 13-21. The conditional in line 7 avoids visiting those cells that have no significant coefficients. Truncation of the codestream is allowed at the end of each cell when the loop of lines 13-21 is executed. To reduce the number of truncation points in large codeblocks, the algorithm may restrict truncation points after the coding of every 2 or 4 cells, for instance. The proposed algorithm is named *hybrid* due to the diversity of techniques that it deploys.

We remark that the principal mechanisms that reduce computational load are the skipping of significance coding for many coefficients at high bitplanes, and the reduction of the number of coefficients visited in the refinement pass. The implementation of these mechanisms with simple data structures leads to low-complexity implementations of the hybrid algorithm.[5]

Figure 2(d) depicts the visiting sequence of coefficients followed by the hybrid algorithm. Both for significance and refinement coding, the algorithm is well-adapted to the data, visiting coefficients following an order similar to that followed by the optimal strategy.

C. Extension to 3D image coding

The work presented in [38] demonstrates that the LAV probability model can be extended to 3D image coding as well. The main idea behind the scheme introduced in [38]

[5]Our implementation of the hybrid codec is available at http://www.deic.uab.es/~francesc

is that redundancy among components of a 3D image can be exploited by the probability model instead of a transform. The proposed probability model is based on the same principles as LAV, but using the prior coefficient instead of the local average. Denote samples of a 3D image arising from the application of (only) a 2D transform to each spatial component 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. Each 2D component is then encoded as described above for LAV, but substituting the magnitude of the quantized "prior coefficient" for the local average $\hat{\varphi}$ when encoding $w_{z,y,x}$. The prior coefficient of $w_{z,y,x}$ is defined as $w_{z-1,y,x}$. Due to page constraints, we refer the reader to [38] for an extended description of the 3D image coding scheme.

V. Experimental results

A. Coding performance

The coding performance of the hybrid strategy is evaluated for the corpus ISO 12640-1 (images are 8 bit, gray-scale, size 2560×2048)[6]. Figures 6(a), 6(b), and 6(c) depict the difference in PSNR between the hybrid scanning order and JPEG2000 for different sizes of codeblock when using lossy compression. At low bitrates the coding performance of both strategies is almost the same, whereas at medium and high bitrates, the performance of the hybrid strategy is slightly better, especially for small codeblocks. We remark that the differences on coding performance are mostly *not* caused by the scanning order, but by the probability model. See in Figure 6(d) the results achieved when the hybrid algorithm employs CAD (as defined in JPEG2000) to code significant and refinement bits. Hybrid achieves virtually the same coding performance as that of JPEG2000. Figures 7(a), and 7(b) report the same evaluation for lossy-to-lossless compression. Results are similar as those achieved with lossy compression.

The coding performance of the hybrid algorithm when coding 3D images is assessed with two types of images: three AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) images of size 512×512 with 224 components and a bit-depth of 16 bits per sample (bps) that belong to the remote sensing field; and three Computed Tomography (CT) images of size 512×512 with 112 components and a bit-depth of 12 bps that belong to the medical community. Table II reports lossless results when these image are compressed with JPEG2000 and with the hybrid algorithm. The column "JPEG2000 2D" reports results when no transform is applied along the z axis, whereas column "JPEG2000 1D+2D" reports results when 5 levels of wavelet transform are applied along the z axis. As described above, hybrid does not use a transform along the z axis. In all cases, 5 levels of wavelet decomposition are applied spatially. Results indicate that hybrid achieves the best coding performance for CT images (slightly better than "JPEG2000 1D+2D"). For hyperspectral images, hybrid is significantly better than "JPEG2000 2D," and somewhat worse than "JPEG2000 1D+2D," though hybrid has significantly lower complexity as detailed below.

TABLE II: Evaluation of the lossless coding performance achieved by the hybrid strategy and JPEG2000. Results are reported as the bitrate of the compressed codestream, in bps.

	JPEG2000		hybrid
	2D	1D+2D	
AVIRIS - cuprite	7.01	5.28	5.74
AVIRIS - jasper	7.66	5.54	6.08
AVIRIS - lowAltitude	7.83	5.95	6.56
AVERAGE	*7.50*	*5.59*	*6.13*
CT - A	8.38	8.07	8.00
CT - B	8.41	8.07	8.04
CT - C	8.33	8.01	7.99
AVERAGE	*8.37*	*8.05*	*8.01*

B. Computational complexity

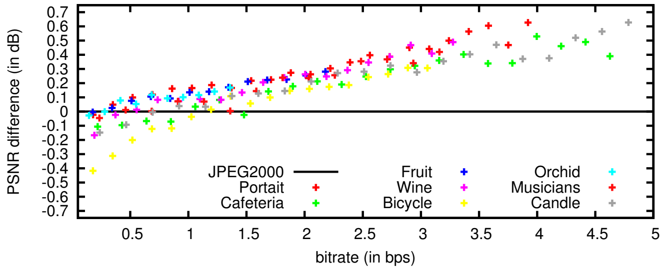
Achieving competitive coding performance commonly means high computational complexity. Hybrid is devised to achieve state-of-the-art coding performance with reduced complexity. The evaluation of computational costs is performed on an Intel Core i7 870 CPU at 2.93 GHz. All methods are implemented in our Java implementation BOI [39], and executed on a Java Virtual Machine v1.6 using GNU/Linux v2.6. Results report the CPU processing time spent by the bitplane coder. The implementation of the JPEG2000 context-adaptive approach uses several software optimizations as suggested in [26]. Hybrid uses a similar degree of optimization.

Figure 8 reports results when coding the images of the ISO 12640-1 corpus. This figure assesses JPEG2000, hybrid, and the single pass strategy. JPEG2000 uses its original CAD probability model for the arithmetic coder. Hybrid and the single pass strategy use the probability model described in Section IV. The single pass strategy is reported in this figure to indicate the minimum costs required by a bitplane image coder. On average, hybrid requires only 7% more computational time than the single pass strategy, whereas JPEG2000 requires 27% more time. Recall that the single pass strategy achieves significantly poorer coding performance than that achieved by hybrid, as reported in Figure 4 for the "Portrait" image. Similar results hold for the other images of the corpus.

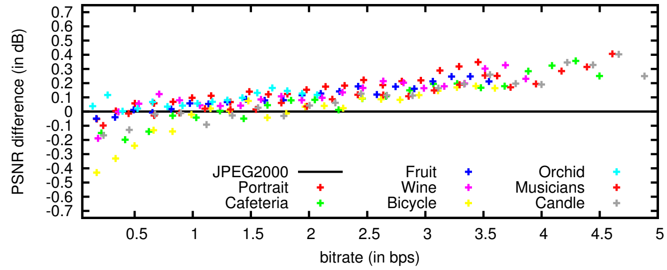
Commonly, the computational costs of the bitplane coder are partially attributed to the arithmetic coder. Internal columns depicted in Figure 8(a) report computational time when the arithmetic coder MQ is replaced by a raw coder. This is not practical, but gives an approximation of the complexity added by the arithmetic coder. Results suggest that hybrid with MQ is only slightly more computationally complex than JPEG2000 with*out* MQ. Profiling of the software modules indicates that the scanning order, the probability model, and the MQ coder in our implementation spend roughly 40%, 35%, and 25% of the overall computational time, respectively.

Figure 9 depicts the same evaluation as above for the coding of 3D images. For the three AVIRIS images, hybrid

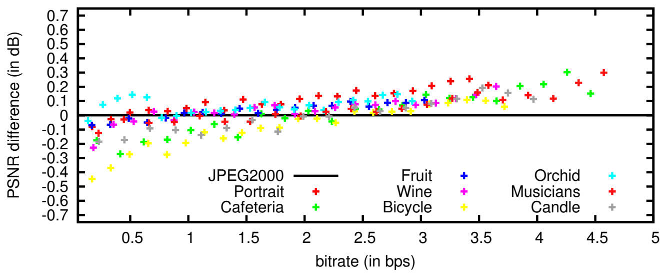
[6] Except when indicated, coding parameters are: 5 levels of irreversible 9/7, or reversible 5/3 wavelet transform, codeblock size of 64×64, single quality layer codestreams, no precincts. The dequantization procedure is carried out as defined in [30].



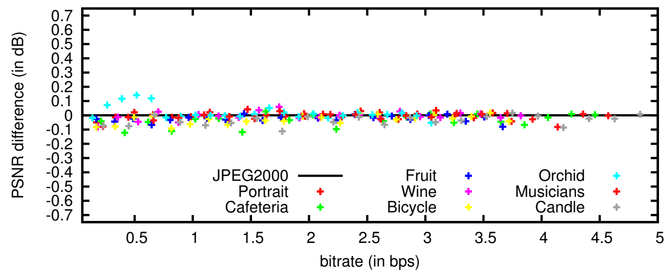
(a) codeblock size 16×16



(b) codeblock size 32×32

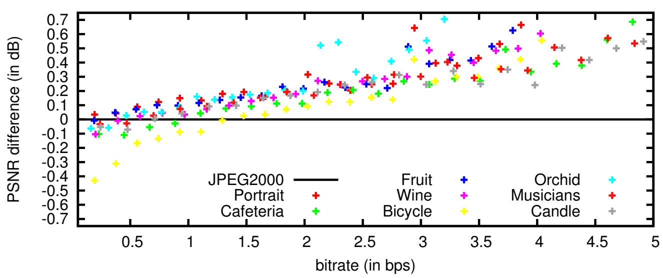


(c) codeblock size 64×64

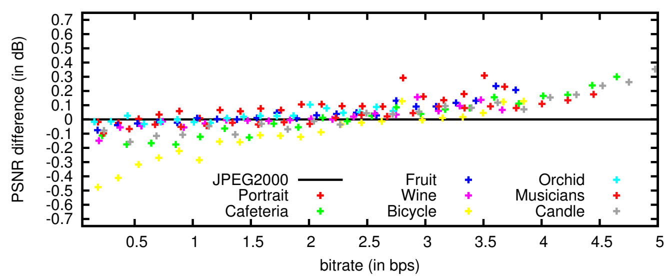


(d) codeblock size 64×64

Fig. 6: Evaluation of the lossy coding performance achieved by the hybrid scanning order compared to JPEG2000, for all images of the ISO 12640-1 corpus. (a), (b), and (c) report results when hybrid uses a combination of LAV and CAD probability models, with codeblocks of different size. (d) reports results when hybrid uses exclusively CAD.



(a) codeblock size 16×16



(b) codeblock size 64×64

Fig. 7: Evaluation of the lossy-to-lossless coding performance achieved by the hybrid scanning order compared to JPEG2000, for all images of the ISO 12640-1 corpus. Hybrid uses a combination of LAV and CAD probability models.

and JPEG2000 spend, respectively, 54% and 154% more computational time than the single pass strategy, on average. For the medical images, these percentages are 40% and 116%, respectively. In both cases the speedup of Hybrid with respect to JPEG2000 is approximately 1.6.

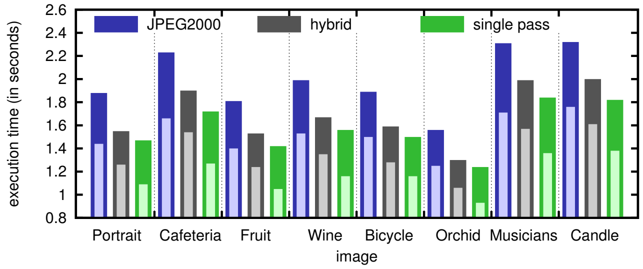
VI. Conclusions

Bitplane image coding is the core technology in many state-of-the-art coding schemes and standards. The interest raised in many different scenarios for such a technology is the main motivation behind this work to assess the efficacy of scanning orders deployed in coding engines. To do so, several mechanisms based on rate-distortion theory are proposed, namely, the distortion-length slope $\mathcal{S}'(w)$ to evaluate the transmission order in each bitplane, the local average $\hat{\varphi}$ to evaluate the visiting sequence of coefficients, and an optimal coding strategy that approximates the convex hull of the rate-distortion function of codeblocks.

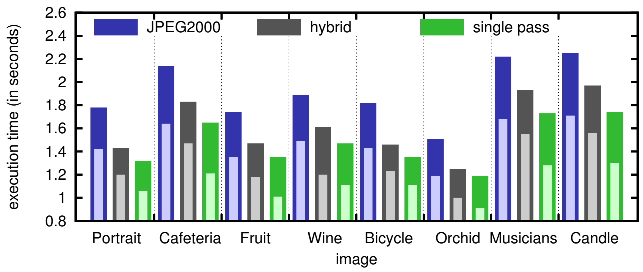
The evaluation of classic scanning orders reveals the efficacy of the techniques deployed in the bitplane coder from different perspectives. This may help to enhance bitplane image codecs with special purposes or requirements. To illustrate this point, a low complexity scanning order is proposed. The main insights used to devise the proposed strategy are taken from our previous analysis. Experimental evidence suggests that the proposed algorithm achieves competitive coding performance while reducing the use of computational resources.

Appendix A

The local average of coefficients is computed using the magnitude of partially transmitted coefficients, i.e., $\hat{\varphi} =$



(a) encoding



(b) decoding

Fig. 8: Evaluation of the computational costs of JPEG2000, hybrid, and a single pass strategy, for the corpus ISO 12640-1. (a) reports results for the encoder. (b) reports results for the decoder. External columns report execution time when using the arithmetic coder MQ, whereas internal columns report execution time when avoiding the use of MQ.

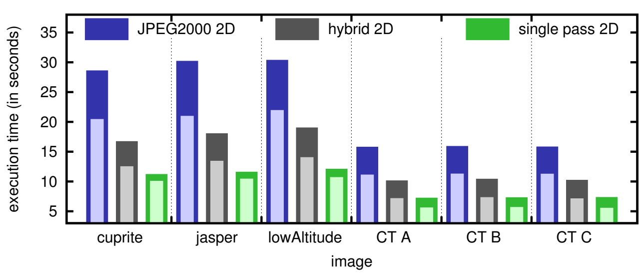


Fig. 9: Evaluation of the computational costs of JPEG2000, hybrid, and a single pass strategy, when decoding 3D images. External columns report execution time when using the arithmetic coder MQ, whereas internal columns report execution time when avoiding the use of MQ.

$\frac{1}{N}\sum_{n=1}^{N}|\hat{w}^n|$. This does not impact coding efficiency except at the highest bitplanes, when little information has been transmitted and most coefficients are still quantized as 0. This penalty can be avoided in practice by assumptions described in [29], which generally predict that the magnitude of $\hat{w}^n$ should be less than w. In that work, this is used to approximate the eight immediate neighbors of w as $\hat{w}'^n = \hat{w} * \beta$ if they are insignificant when computing the local average, with $\beta = 0.4$. To reduce computational costs, the hybrid algorithm only uses the four neighbors that are above, below, to the right, and left of w, and computes $\hat{w}'^n$ as $\hat{w}'^n = \hat{w}/2$, which can be implemented as a bit-wise operation. Experimental evidence suggests that coding performance is only slightly penalized compared to [29].

References

[1] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Trans. Image Process.*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[2] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, Jun. 1996.

[3] A. Zandi, J. D. Allen, E. L. Schwartz, and M. Boliek, “CREW: compression with reversible embedded wavelets,” in *Proc. IEEE Data Compression Conference*, Mar. 1995, pp. 212–221.

[4] X. Wu and J. hua Chen, “Context modeling and entropy coding of wavelet coefficients for image compression,” in *Proc. IEEE International Conference Acoustics, Speech, and Signal Processing*, vol. 4, Apr. 1997, pp. 3097–3100.

[5] R. W. Buccigrossi and E. P. Simoncelli, “Image compression via joint statistical characterization in the wavelet domain,” *IEEE Trans. Image Process.*, vol. 8, no. 12, pp. 1688–1701, Dec. 1999.

[6] X. Wu, “Context quantization with Fisher discriminant for adaptive embedded wavelet image coding,” in *Proc. IEEE Data Compression Conference*, Mar. 1999, pp. 102–111.

[7] E. Ordentlich, M. Weinberger, and G. Seroussi, “A low-complexity modeling approach for embedded coding of wavelet coefficients,” in *Proc. IEEE Data Compression Conference*, Apr. 1998, pp. 408–417.

[8] J. Li and S. Lei, “An embedded still image coder with rate-distortion optimization,” *IEEE Trans. Image Process.*, vol. 8, no. 7, pp. 913–924, Jul. 1999.

[9] S.-T. Hsiang, “Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling,” in *Proc. IEEE Data Compression Conference*, Mar. 2001, pp. 83–92.

[10] ——, “Highly scalable subband/wavelet image and video coding,” Ph.D. dissertation, Rensselaer Polytechnic Institue, Troy, NY, Jan. 2002.

[11] H. Danyali and A. Mertins, “Fully spatial and SNR scalable, SPIHT-based image coding for transmission over heterogenous networks,” *Journal of Telecommuncations and Information Technology*, pp. 92–98, Feb. 2003.

[12] D. Taubman, “High performance scalable image compression with EBCOT,” *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.

[13] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said, “Efficient, low-complexity image coding with a set-partitioning embedded block coder,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 11, pp. 1219–1235, Nov. 2004.

[14] N. Mehrseresht and D. Taubman, “A flexible structure for fully scalable motion-compensated 3-D DWT with emphasis on the impact of spatial scalability,” *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 740–753, Mar. 2006.

[15] 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.

[16] H. G. Lalgudi, A. Bilgin, M. W. Marcellin, and M. S. Nadar, “Compression of multidimensional images using JPEG2000,” *IEEE Signal Process. Lett.*, vol. 15, pp. 393–396, Apr. 2008.

[17] M. Farshchian, S. Cho, and W. A. Pearlman, “Optimal error protection for real-time image and video transmission,” *IEEE Signal Process. Lett.*, vol. 11, no. 10, pp. 780–783, Oct. 2004.

[18] Z. Wu, A. Bilgin, and M. W. Marcellin, “Error resilient decoding of JPEG2000,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 12, pp. 1752–1757, Dec. 2007.

[19] M. Dyer, S. Nooshabadi, and D. Taubman, “Design and analysis of system on a chip encoder for JPEG2000,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 215–225, Feb. 2009.

[20] A. Abrardo, M. Barni, E. Magli, and F. Nencini, “Error-resilient and low-complexity onboard lossless compression of hyperspectral images by means of distributed source coding,” *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 4, pp. 1892–1904, Apr. 2010.

[21] J. Oliver and M. P. Malumbres, “Low-complexity multiresolution image compression using wavelet lower trees,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 11, pp. 1437–1444, Nov. 2006.

[22] *Digital compression and coding for continuous-tone still images*, ISO/IEC Std. 10 918-1, 1992.

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

[24] *Image Data Compression*, Consultative Committee for Space Data Systems Std. CCSDS 122.0-B-1, Nov. 2005.

[25] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Process.*, vol. 1, no. 2, pp. 205–220, Apr. 1992.

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

[27] Z. Liu and L. J. Karam, "Mutual information-based analysis of JPEG2000 contexts," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 411–422, Apr. 2005.

[28] M. Cagnazzo, M. Antonini, and M. Barlaud, "Mutual information-based context quantization," *ELSEVIER Signal Processing: Image Communication*, vol. 25, no. 1, pp. 64–74, Jan. 2010.

[29] 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.

[30] 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.

[31] P. Simard, D. Steinkraus, and H. Malvar, "On-line adaptation in image coding with a 2-D tarp filter," in *Proc. IEEE Data Compression Conference*, Apr. 2002, pp. 23–32.

[32] C. Tian and S. S. Hemami, "An embedded image coding system based on tarp filter with classification," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, May 2004, pp. 17–21.

[33] 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.

[34] F. Auli-Llinas, "Local average-based model of probabilities for JPEG2000 bitplane coder," in *Proc. IEEE Data Compression Conference*, Mar. 2010, pp. 59–68.

[35] F. Auli-Llinas and M. W. Marcellin, "Rate-distortion optimized adaptive scanning order for bitplane image coding engines," in *Proc. IEEE Data Compression Conference*, Mar. 2011, pp. 163–172.

[36] Y. Yeung and O. C. Au, "Efficient rate control for JPEG2000 image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 3, pp. 335–344, Mar. 2005.

[37] 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.

[38] F. Auli-Llinas, M. W. Marcellin, J. Serra-Sagrista, and J. Bartrina-Rapesta, "Lossy-to-lossless 3D image coding through prior coefficient lookup tables," *ELSEVIER Information Sciences*, 2011, under revision.

[39] F. Auli-Llinas. (2011) BOI software. [Online]. Available: http://www.deic.uab.es/~francesc

Francesc Aulí-Llinàs (S'2006-M'2008) is a research fellow of the program Ramón y Cajal funded by the Spanish Government. He is currently working with the Group on Interactive Coding of Images of the Department of Information and Communications Engineering, at the Universitat Autònoma de Barcelona (Spain). He received the B.Sc. and B.E. degrees in Computer Management Engineering and Computer Engineering in 2000 and 2002, respectively, both from the Universitat Autònoma de Barcelona (Spain), and for which he was granted with two extraordinary awards of Bachelor (awards given to the first students of the promotion). In 2004 and 2006 he respectively received the M.S. degree and the Ph.D. degree (with honors), both in Computer Science from the Universitat Autònoma de Barcelona. Since 2002 he has been consecutively awarded with doctoral and postdoctoral fellowships in competitive calls. From 2007 to 2009 he carried out two research stages of one year each with the group of David Taubman, at the University of New South Wales (Australia), and with the group of Michael Marcellin, at the University of Arizona (USA). He is the main developer of BOI, a JPEG2000 Part 1 implementation that was awarded with a free software mention from the Catalan Government in April 2006. He is involved in several research projects granted by the European Union, the Spanish and the Catalan Governments. His research interests include a wide range of image coding topics, including highly scalable image and video coding systems, rate-distortion optimization, distortion estimation, and interactive transmission, among others.

Michael W. Marcellin (S'81-M'87-SM'93-F'02) was born in Bishop, California, on July 1, 1959. He graduated *summa cum laude* with the B.S. degree in Electrical Engineering from San Diego State University in 1983, where he was named the most outstanding student in the College of Engineering. He received the M.S. and Ph.D. degrees in Electrical Engineering from Texas A&M University in 1985 and 1987, respectively. Since 1988, Dr. Marcellin has been with the University of Arizona, where he holds the title of Regents' Professor of Electrical and Computer Engineering, and of Optical Sciences. His research interests include digital communication and data storage systems, data compression, and signal processing. He has authored or coauthored more than two hundred publications in these areas. Dr. Marcellin is a major contributor to JPEG2000, the emerging second-generation standard for image compression. Throughout the standardization process, he chaired the JPEG2000 Verification Model Ad Hoc Group, which was responsible for the software implementation and documentation of the JPEG2000 algorithm. He is coauthor of the book, *JPEG2000: Image compression fundamentals, standards and practice,* Kluwer Academic Publishers, 2002. Dr. Marcellin served as a consultant to Digital Cinema Initiatives (DCI), a consortium of Hollywood studios, on the development of the JPEG2000 profiles for digital cinema. Professor Marcellin is a Fellow of the IEEE, and is a member of Tau Beta Pi, Eta Kappa Nu, and Phi Kappa Phi. He is a 1992 recipient of the National Science Foundation Young Investigator Award, and a corecipient of the 1993 IEEE Signal Processing Society Senior (Best Paper) Award. He has received teaching awards from NTU (1990, 2001), IEEE/Eta Kappa Nu student sections (1997), and the University of Arizona College of Engineering (2000). In 2003, he was named the San Diego State University Distinguished Engineering Alumnus. Professor Marcellin is the recipient of the 2006 University of Arizona Technology Innovation Award. From 2001 to 2006, Dr. Marcellin was the Litton Industries John M. Leonis Professor of Engineering. He is currently the International Foundation for Telemetering Professor of Electrical and Computer Engineering at the University of Arizona.