Stationary Probability Model for Microscopic Parallelism in JPEG2000

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

***Abstract*—Parallel processing is key to augmenting the throughput of image codecs. Despite numerous efforts to parallelize wavelet-based image coding systems, most attempts fail at the parallelization of the bitplane coding engine, which is the most computationally intensive stage of the coding pipeline. The main reason for this failure is the causality with which current coding strategies are devised, which assumes that one coefficient is coded after another. This work analyzes the mechanisms employed in bitplane coding and proposes alternatives to enhance opportunities for parallelism. We describe a stationary probability model that, without sacrificing the advantages of current approaches, removes the main obstacle to the parallelization of most coding strategies. Experimental tests evaluate the coding performance achieved by the proposed method in the framework of JPEG2000 when coding different types of images. Results indicate that the stationary probability model achieves similar coding performance, with slight increments or decrements depending on the image type and the desired level of parallelism.**

***Index Terms*—Parallel architectures, JPEG2000, probability models, bitplane image coding.**

I. Introduction

FOR the last two decades, most image coding systems have relied on context-adaptive approaches to determine probabilities of symbols emitted by coding engines. In such approaches, the context of a coefficient is defined through some characteristics of its neighbors. At the start of coding, the (conditional) probabilities of all symbols emitted in each context are set to some fixed initial values. Typically, this initialization assumes that the symbols are equiprobable (uniform) within each context. However, these initial distributions need not be uniform, and can differ from context to context. As the coding process evolves, the symbol probabilities are adjusted depending on the coded data, i.e., increasing (or decreasing) as symbols appear more (or less) frequently. Emitted symbols are fed to an entropy coder, which employs their probabilities in its compression process.

The effectiveness of context-adaptive approaches stems from two main mechanisms. The first is the discrimination of symbol probabilities imposed by the contexts. Contexts permit the encoding of the same symbol using different probabilities that depend on the context in which it is emitted [1]–[3]. The second mechanism behind the effectiveness of context-adaptive approaches is the ability to adapt to varying local statistics. In general, probabilities are adjusted considering the data coded, rapidly increasing the probabilities of those symbols that have appeared recently [4], [5].

In addition to being computationally non-complex, the main advantages of context-adaptive approaches are high compression efficiency and the separation of modeling and coding, allowing a single entropy coding strategy. Additionally, the use of adaptive probability models avoids a pre-processing step to collect statistics, since statistics are collected at the same time that data are coded. Context-adaptive technology has become prevalent in image and video coding, and has been introduced in many compression standards such as JPEG2000 [6], AVC [7], HEVC [8], and JBIG [9].

Despite their popularity, context-adaptive approaches have two major drawbacks. The first is poor performance when short data sequences are coded. In JPEG2000, for instance, spatial scalability is achieved by coding rectangular sets of wavelet coefficients (called codeblocks) independently. The smaller the codeblock, the finer the scalability, though then adaptive strategies may not have sufficient data to adjust the probabilities reliably, thus sacrificing compression performance. The second drawback is the lack of opportunities for parallelism. This is mainly caused by the fact that adaptive procedures are intrinsically causal systems in which the probability of the current symbol depends on all symbols previously coded. In practice, this compels coefficients to be coded sequentially. In addition to adaptivity, certain context structures may also force sequential coding or decoding, since the characteristics of coefficients just before the current one may also be needed to determine its context. These issues are significant for microscopically parallel architectures, since they prevent the simultaneous processing of (all) coefficients.

Parallel processing continues to gain importance in many fields [10], [11] due to the advent of multi-core architectures and (general-purpose) graphics processing units (GPUs). Highly parallel algorithms can significantly accelerate their sequential counter-parts, which opens new possibilities for a wide range of applications [12]–[14] that deal with multimedia content or require fast, or real-time, processing.

To the best of our knowledge, there are few wavelet-based image coding systems that do not employ context and/or adaptive strategies. The early codec proposed in [15], [16] determines probabilities through a characterization of wavelet data that does not require adaptiveness. In the same vein, the strategy introduced in [17] employs a joint probability density function (pdf) of wavelet coefficients and their local averages through which probabilities of individual symbols can be integrated before coding begins. Unfortunately, the principal objective in [17] was the exploration of context models that

Dr. Francesc Aulí-Llinàs is with the Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, Spain (phone: +34 935811861; fax: +34 935813443; e-mail: fauli@deic.uab.cat). Prof. Michael W. Marcellin is with the Department of Electrical and Computer Engineering, University of Arizona, USA (e-mail: marcellin@ece.arizona.edu).
This work has been partially supported by the Spanish Government (MINECO), by FEDER, and by the Catalan Government, under Grants RYC-2010-05671, TIN2012-38102-C03-03, and 2009-SGR-1224.

enhance coding performance, without considering parallelism. Hence some contexts employ adaptive mechanisms.

The purpose of the work reported here is to explore strategies for the parallelization of the coding stage of wavelet-based image codecs. The strategy proposed herein is aimed at the finest level of parallelism possible, i.e., the parallel processing of individual coefficients. This is referred to as microscopic parallelism [18]. In modern codecs, the main issue that prevents the achievement of this level of parallelism is the adaptive probability model. In what follows, we introduce a stationary probability model that overcomes the drawbacks of adaptive mechanisms while preserving most advantages of modern codecs. The main idea is to establish probability estimates of symbols beforehand, by employing a training set of images. These probability estimates are kept in a lookup table (LUT) that is shared by the encoder and the decoder.

The proposed strategy is viable for any context-based coding system such as [19]–[23]. For concreteness, the proposed scheme is described and tested within the framework of JPEG2000, without sacrificing any of the advanced features of the standard. In addition to the static probability model, a slight modification of JPEG2000 context formation is posed as a means to increase parallelism opportunities. Experimental results suggest that the proposed strategies achieve similar coding performance to that of conventional context-adaptive approaches, with slight increments or decrements depending on the type of image being coded. As additional assets, the proposed method significantly improves compression efficiency when fine spatial scalability is needed, removes the machinery needed to realize context-adaptive mechanisms, and may help to simplify implementation in hardware architectures.

We remark here that context-adaptive arithmetic coding is sometimes viewed as a (single) entropy coding algorithm. In this work however, we take the more common point of view [16], [17], [24], [25] that modeling (e.g., context formation and probability estimation) is separate from entropy coding (e.g., arithmetic coding). The goal of the research presented here is then the exploration of probability models and contexts that admit highly parallel implementation. Previous work [26]–[29] has focused on highly parallel implementation of the wavelet transform. Our future work will focus on highly parallelizable entropy coding techniques such as [30], [31].

The remainder of the paper is structured as follows. Section II briefly reviews the context-adaptive approach employed in JPEG2000. Section III introduces the stationary probability model and the coding strategies proposed in this work. The coding performance of the proposed method is assessed in Section IV employing four corpora containing different types of images. The last section summarizes this research and provides concluding remarks.

II. Overview of JPEG2000 context-adaptive mechanisms

A. Context formation

After a wavelet transform, JPEG2000 partitions wavelet subbands into rectangular codeblocks. The maximum number of coefficients within a codeblock is 4096, with the width and

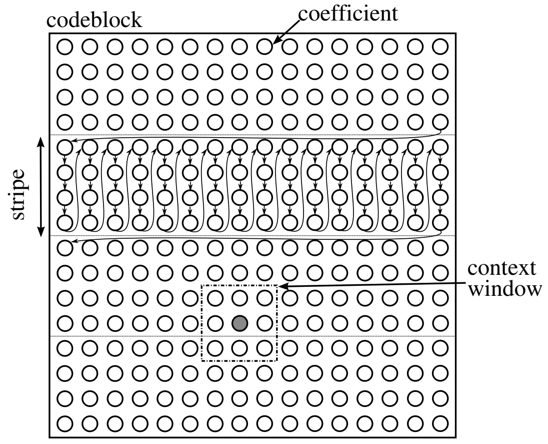


Fig. 1: Illustration of the scanning order and the context window defined in JPEG2000.

the height of the codeblock being any power of two equal to or greater than 2^2. Typically, large codeblocks (e.g., 64×64) yield the best coding performance in terms of mean squared error. However, codeblocks of size 32×32 or smaller have important applications [19], [32]. Codeblocks of size 16×16 or smaller are not commonly used because of lower compression efficiency. If not for this shortcoming, smaller codeblocks would be highly desirable in some cases. For example, see the visual masking discussion in [18, Ch. 16.1].

Each codeblock is coded independently by means of a bitplane coding strategy. Let $[b_{M-1}, b_{M-2}, ..., b_1, b_0]$, $b_i \in \{0, 1\}$ be the binary representation of an integer υ which represents the magnitude of the index obtained by quantizing wavelet coefficient ω, with M being a sufficient number of bits to represent all coefficients. The collection of bits b_j from all coefficients is called a bitplane. Bitplane coding strategies code bits from the most significant bitplane $M-1$ to the least significant bitplane 0. The first non-zero bit of the binary representation of υ is denoted by b_s and is referred to as the significance bit. The sign of the coefficient, denoted by $d \in \{+, -\}$, is coded immediately after b_s, so that the decoder can begin to approximate ω as soon as possible. The bits $b_r, r < s$ are referred to as refinement bits.

JPEG2000 employs three coding passes in each bitplane, called the significance propagation pass (SPP), the magnitude refinement pass (MRP), and the cleanup pass (CP). Each bit in a bitplane is coded in one of these coding passes. The SPP codes bits from non-significant coefficients that are likely to become significant in the current bitplane, while the MRP codes refinement bits from coefficients that were significant in previous bitplanes. The CP codes any remaining bits in the bitplane, i.e., those from non-significant coefficients that were not coded in the SPP. Each coding pass visits coefficients in a scanning order that is organized in stripes. A stripe is defined as four consecutive rows of coefficients (see Fig. 1). Within each stripe, coefficients are scanned from top to bottom in each column, from the leftmost to the rightmost column.

Each symbol emitted by the coding engine is associated to one of the 19 contexts defined by the standard [18]. Contexts $\phi \in \{0, ..., 8\}$ are devoted to significance coding in SPP and

CP. Contexts $\phi \in \{10, ..., 14\}$ are employed to code sign bits, also in SPP and CP. Contexts $\phi \in \{15, 16, 17\}$ are associated to refinement bits in MRP. CP also employs context $\phi = 9$ to code bits emitted during a special coding mode called run mode. In run mode, only one bit is emitted per four consecutive coefficients. This mode is activated only when it is likely that the four coefficients will not become significant in the current bitplane. Context $\phi = 18$ is reserved to code the position of the first coefficient that was found to be significant during the run mode, if any.

The contexts devoted to significance coding are defined as a function of the significance state of the 8 immediate neighbors of the coefficient. These neighbors are illustrated in Fig. 1 as those within the context window. The significance state of a neighbor is 1 if its significance bit has already been coded, and 0 otherwise. This clearly includes all coefficients that became significant in bitplanes higher than the current. It also includes the coefficients that become significant in the current bitplane –and that appear earlier in the scanning order than the current coefficient. This last observation has important implications for parallelization.

Similarly, the contexts devoted to sign coding are defined as a function of the sign of already significant neighbors. Contexts employed for refinement bits employ the significance state of the neighbors and the bitplane number of the refinement bit, relative to that of the significance bit.

B. *Adaptive mechanisms*

The state machine depicted in Fig. 2 is employed to adapt symbol probabilities in JPEG2000. Each dot in the figure is a state. In each state, estimated symbol probabilities are given by the vertical axis of the figure. All emitted symbols are binary, so their probabilities are expressed in the range $[0.5, 1)$ as the probability of the most probable symbol (MPS). The entropy coder dynamically changes which symbol is considered to be most probable when the context is in a state with probability 0.5 and the least probable symbol (LPS) is coded. This is represented in the figure by the dotted red arrows of the bottom-left states. All states in the figure have a blue and a red outgoing arrow that represent state transitions when the MPS and LPS are coded, respectively. The transitions only occur when the internal registers of the arithmetic coder reach a particular state, which happens once every few symbols. The current state, as well as the MPS, is maintained separately for each context. At the start of coding, all contexts except $\phi \in \{0, 9, 18\}$ are set to the bottom-leftmost state of the figure. Contexts $\phi \in \{0, 9\}$ are initialized to states with higher probabilities and with MPS $= 0$, since these contexts begin coding mostly zeroes. Context $\phi = 18$ is set to a special state with only self-transitions and constant probability 0.5 (bottom-right corner of the figure).

For a given context, symbol probabilities rise through the state following one of the three vertical paths formed by the blue lines in Fig. 2. The two leftmost paths are starting mechanisms that are employed solely in the beginning of the coding process to rapidly increase the probability when the MPS is coded repeatedly. The coding of an LPS lowers the probability and, if the context is associated with a state along one of the two leftmost paths, the next state is selected along the next path to the right. Eventually, the state is associated with the rightmost path, along which probabilities rise and fall more smoothly.

The dotted lines in Fig. 3 depict the estimated probabilities for the symbols emitted with context $\phi = 1$ when coding each bitplane of one codeblock of one image. Specifically, data correspond to one codeblock in the high vertical-, low horizontal-frequency subband of the first decomposition level of the "Portrait" image (ISO 12640-1 corpus, 8 bit, gray-scale). The irreversible 9/7 wavelet transform is employed. Each subfigure reports the probabilities employed to code symbols emitted in a different bitplane. The vertical axis is the probability of the MPS, whereas the horizontal axis is the index of the coefficient corresponding to the symbol coded, according to the scanning order discussed previously. This results in all subfigures having the same width corresponding to the number of coefficients in the codeblock, even though there are a different number of symbols in each bitplane. At the start of coding (leftmost part in the highest subfigure) probabilities rise and drop rapidly due to the aforementioned starting mechanisms. Probabilities fluctuate more smoothly in subsequent bitplanes as they are adapted to the coded data. Similar behavior is observed for other contexts, codeblocks, and images.

III. Stationary probability model

A. *Main insights*

As previously discussed, the main drawbacks of context-adaptive approaches are poor coding performance when fine spatial scalability is required, and the provision of few opportunities for parallelism. These drawbacks are mainly caused by the adaptation mechanisms, and can be eliminated by means of a stationary model of probabilities. The proposed probability model employs two main insights. The first is derived via observation of Fig. 3. Note from this figure that the probabilities employed to code the symbols emitted –within the same bitplane– tend to trend around the average value for that bitplane. These averages are illustrated by the straight lines in each subfigure. The more important differences are found between the average probabilities of different bitplanes. At the highest bitplane, the average probability is nearly 0.99, whereas at the lowest bitplanes, it falls to around 0.5.

This empirical evidence suggests that the adaptive mechanisms of conventional approaches are effective to adjust probabilities between bitplanes, though little gain is obtained from adjusting them within the same bitplane. Our hypothesis is that adaptation can be omitted for symbols emitted in the same bitplane without penalizing coding performance. The proposed method uses constant probability for symbols emitted in the same bitplane, but varies this constant value from bitplane to bitplane. This amounts to using the probabilities depicted by the straight lines in Fig. 3. As discussed below, this method provides more opportunities for parallelism and enhances coding performance when using codeblocks of small size.

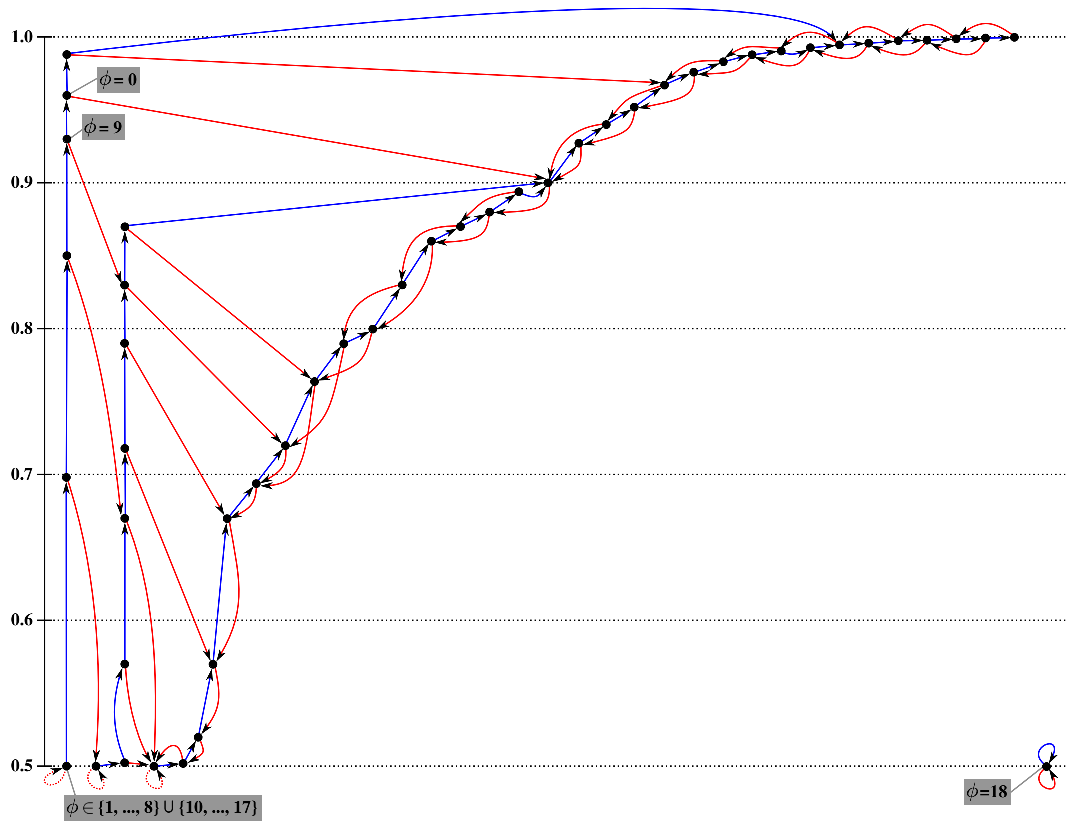


Fig. 2: State machine employed to adjust symbol probabilities in JPEG2000. Dots represent states, whereas red and blue arrows are transitions between states when coding the most and the least probable symbol, respectively. The probability of each state is represented by the vertical axis of the figure. The gray boxes indicate initial states of JPEG2000 contexts.

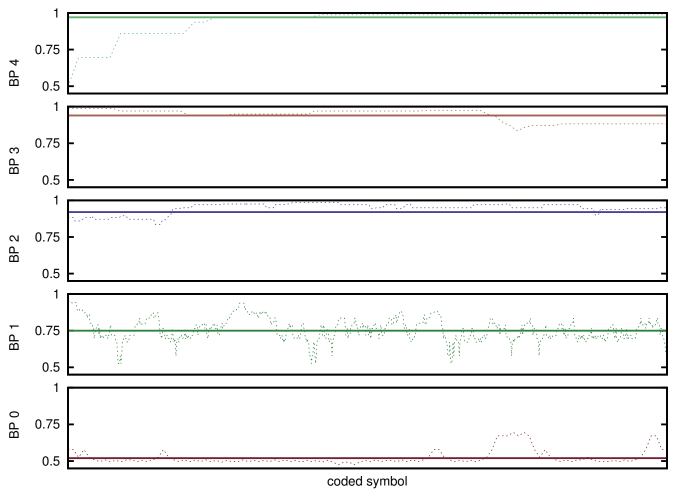


Fig. 3: Estimated symbol probabilities. Dotted lines depict the probabilities employed to code symbols emitted with context $\phi = 1$. The vertical axis in each subfigure is the probability of the MPS, whereas the horizontal axis is the index of the symbol coded. The solid straight line in each subfigure is the average probability in each bitplane.

The second insight behind the proposed method is that, in addition to avoiding symbol by symbol adaptivity, a highly parallel coding scheme must avoid a pre-processing step to collect data statistics of the image to be coded. To this end, we take advantage of the typical statistical behavior of the data produced by a wavelet transform, which is similar for images of the same type. For example, the experimental results provided below utilize four different types of images classified as natural, aerial, hyperspectral AVIRIS, and XRAY angiography. These image types belong to different fields and are captured with different sensors. As observed in other works [16], [17], [33], the data produced after transforming images of the same type with the same wavelet filter-bank are statistically similar. We exploit this fact to generate a single lookup table (LUT) for each image type. This LUT contains the probability estimate for each context, subband, and bitplane. The LUT is assumed to be precomputed off-line, and known by both the encoder and the decoder, so there is no need of an image-to-image pre-processing step to collect statistics nor to transmit the LUT as side information. Nonetheless, the size of the LUT is in practice so small that its transmission would represent a negligible increase to the length of the codestream (see Section IV).

B. *LUT formation*

The probability estimates needed to populate the LUTs are determined as follows. First, the images of a suitable training set are transformed with the appropriate wavelet transform. The coefficients of each subband are then quantized with a step size computed as the L_2-norm of the synthesis basis vectors of the subband. This produces a signal with energy gain factor of 1 in all subbands. This is a common strategy in JPEG2000, though other quantization step sizes may also be used with the proposed approach. The conditional probability mass function (pmf) of the resulting quantization indices υ is then computed for all symbols that can be emitted by the coding engine, as described in the following paragraphs.

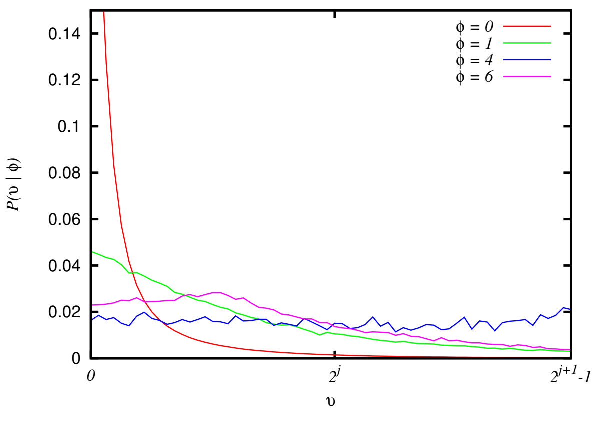
$P_{S_j}(\upsilon \mid \phi)$ denotes the pmf for the significance contexts employed in SPP and CP at bitplane j. It is computed for each wavelet subband using the data from all images in the training set. The subband to which the pmf belongs is not reflected in the notation for simplicity. The support of this pmf is $\{0, ..., 2^{j+1}-1\}$ since it contains quantization indices that were not significant in bitplanes greater than j. Fig. 4(a) depicts $P_{S_j}(\upsilon \mid \phi)$ for one wavelet subband. Only the pmfs for contexts $\phi \in \{0, 1, 4, 6\}$ are depicted to avoid cluttering the figure. Note that the pmf obtained for context $\phi = 0$ has a Laplace-like shape since this context is employed for those coefficients that do not have any significant neighbor. Thus, most of the coefficients coded with this context have magnitudes near 0. Coefficients with contexts $\phi \in \{1, 4, 6\}$ have 1, at least 2, and at least 3 significant neighbors, respectively, so the pmf for these contexts is more uniform.

$P_{R_j}(\upsilon \mid \phi)$ denotes the pmf for the refinement contexts employed in MRP. Its support is $\{2^{j+1}, ..., 2^{j+2}-1\}$ for $\phi \in \{15, 16\}$ since these contexts are solely employed to code the first refinement bit of the quantized coefficients. The support of $P_{R_j}(\upsilon \mid \phi)$ for $\phi = 17$ is $\{2^{j+2}, ..., 2^{M}-1\}$ since this context is employed to code the remaining refinement bits. Fig. 4(b) depicts the pmf obtained for these contexts using the same wavelet subband as that employed for $P_{S_j}(\upsilon \mid \phi)$ above. These pmfs are Laplace-like, though they have different shapes depending on the context.

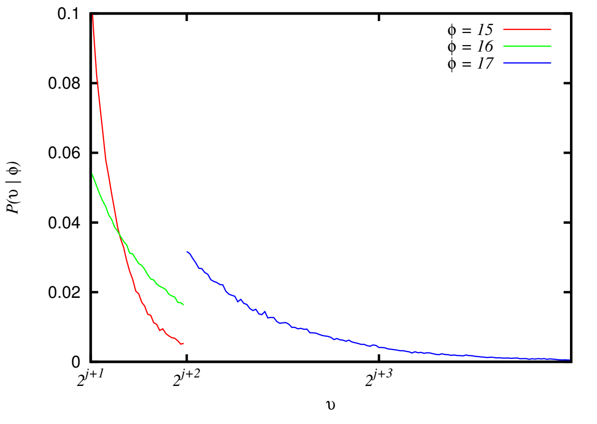
$P_{D_j}(d \mid \phi)$ denotes the pmf for the contexts employed to code the signs of coefficients that become significant in SPP and CP. Accordingly, the support of this pmf is binary. $P_{N_j}(r \mid \phi)$ denotes the pmf for context $\phi = 9$, employed in the run mode of CP. Its support is also binary. Context $\phi = 18$ does not require a pmf since it always employs probability 0.5.

Once the pmfs are computed, the probability estimates used to populate the LUTs are generated by integrating the pmfs to obtain the probabilities of emitting 0 or 1 in the corresponding contexts. Denote the probability that bit b_j is 0 during significance coding by $P_{sig}(b_j = 0 \mid \phi), \phi \in \{0, ..., 8\}$. This probability is determined from the corresponding pmf according to

$$P_{sig}(b_j = 0 \mid \phi) = \frac{\sum_{\upsilon=0}^{2^j-1} P_{S_j}(\upsilon \mid \phi)}{\sum_{\upsilon=2^j}^{2^{j+1}-1} P_{S_j}(\upsilon \mid \phi)} = \frac{\sum_{\upsilon=0}^{2^j-1} P_{S_j}(\upsilon \mid \phi)}{1} = \sum_{\upsilon=0}^{2^j-1} P_{S_j}(\upsilon \mid \phi). \tag{1}$$



(a)



(b)

Fig. 4: Pmfs obtained for the high vertical-, low horizontal-frequency subband of the first decomposition level produced by the irreversible 9/7 wavelet transform. Results are for the corpus of natural images described in Section IV. (a) depicts $P_{S_j}(\upsilon \mid \phi)$ for $\phi \in \{0, 1, 4, 6\}$ and $j = 5$, whereas (b) depicts $P_{R_j}(\upsilon \mid \phi)$ for $\phi \in \{15, 16, 17\}$ and $j = 4$.

Similarly, probabilities for refinement bits are denoted by $P_{ref}(b_j = 0 \mid \phi)$ and are determined as

$$P_{ref}(b_j = 0 \mid \phi) = \sum_{\upsilon=2^{j+1}}^{2^{j+1}+2^j-1} P_{R_j}(\upsilon \mid \phi) \tag{2}$$

when $\phi \in \{15, 16\}$ and as

$$P_{ref}(b_j = 0 \mid \phi) = \sum_{\upsilon=2^{j+2}}^{2^{j+2}+2^j-1} P_{R_j}(\upsilon \mid \phi) + \sum_{\upsilon=2^{j+2}+2^{j+1}}^{2^{j+2}+2^{j+1}+2^j-1} P_{R_j}(\upsilon \mid \phi) + \sum_{\upsilon=2^{j+3}}^{2^{j+3}+2^j-1} P_{R_j}(\upsilon \mid \phi) + ... \tag{3}$$

when $\phi = 17$. Note that Equation (3) sums up all subintervals in which the refinement bit $b_r = 0$.

Probabilities for sign coding and run mode can be taken directly from their pmfs as $P_{D_j}(d = + \mid \phi), \phi \in \{10, ..., 14\}$ and as $P_{N_j}(r = 0 \mid \phi)$, respectively.

One LUT per subband is created containing the probability estimates for each context in each bitplane. The LUT is accessed as $\mathcal{L}[j][\phi]$ with j denoting the bitplane and ϕ the context. Fig. 5 depicts the probability estimates determined for one wavelet subband. Each subfigure represents one of the four corpora employed in the experimental results presented below. The horizontal axis of each subfigure is labeled by bitplane. For each bitplane, 18 vertical bars depict the probability for each of the 18 contexts employed to code symbols. The vertical bars are classified by color depending on the coding function of its corresponding context (i.e., significance, refinement, sign, and run mode). The vertical axis is the probability of the MPS. Note that probability estimates are significantly different for each image type. For instance, the probability estimates for the two lowest bitplanes of Fig. 5(c) are near 0.5 for all but one context, whereas in Fig. 5(b) they are significantly greater than 0.5 for several contexts. Note also that the probability of context $\phi = 0$ (leftmost bar in each bitplane) in Fig. 5(b), is high at the highest bitplanes, medium at the medium bitplanes, and high again at the lowest bitplanes. Contrarily, for the other image types, this probability estimate is highest at the highest bitplanes and lowest at the lowest bitplane. Similar patterns can be observed for other contexts. This variability indicates that LUTs must be computed bitplane by bitplane and image type by image type to achieve high compression efficiency.

C. Parallelism analysis

Parallelization of the JPEG2000 data coding stage can be considered at three different levels: codeblock, coding pass, and coefficient. Codeblock parallelism refers to the use of different execution threads to code different codeblocks. Parallelism at the coding pass level would indicate that the coding passes within a codeblock are executed simultaneously. Coefficient parallelism would imply that coefficients within a codeblock are processed in parallel. In [18][Ch. 12.4.2], parallelism external to the codeblock is referred to as macroscopic parallelism, whereas intra-codeblock parallelism is referred to as microscopic parallelism. Of the two, highly parallel architectures like GPUs are more suited to microscopic parallelism, especially the finest grain parallelism at the coefficient level.

Table I enumerates the levels of parallelism that can be achieved with the proposed method and with different coding strategies compliant with JPEG2000. As seen in the table, JPEG2000 without invoking any of its specialty coding variations can only achieve codeblock parallelism. This macroscopic parallelism is easy to implement because there are no data dependencies among codeblocks. This level of parallelism is also achieved by all other coding strategies evaluated.

From the point of view of probability model adaptation and context formation, coding pass parallelism can be achieved in JPEG2000 through invoking its RESET coding variation [18]. This coding variation re-initializes all contexts at the beginning of each coding pass, eliminating any dependence of probabilities in the current coding pass on probabilities in previous coding passes.[1] In the encoder, coding pass parallelism is then achieved straightforwardly since all probabilities depend only on symbols previously coded in the current coding pass. Furthermore, all bits of all quantization indices are readily available in the encoder, permitting the formation of contexts in all coding passes.

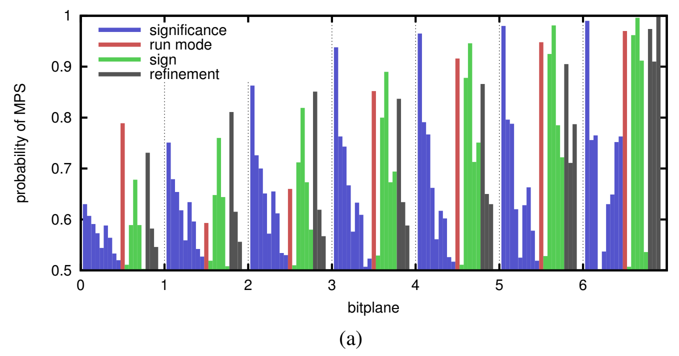
TABLE I: Evaluation of the levels of parallelism achievable at the encoder and the decoder when using different coding strategies in the framework of JPEG2000. An asterisk besides a check mark indicates that tight synchronization is required to achieve the corresponding level of parallelism.

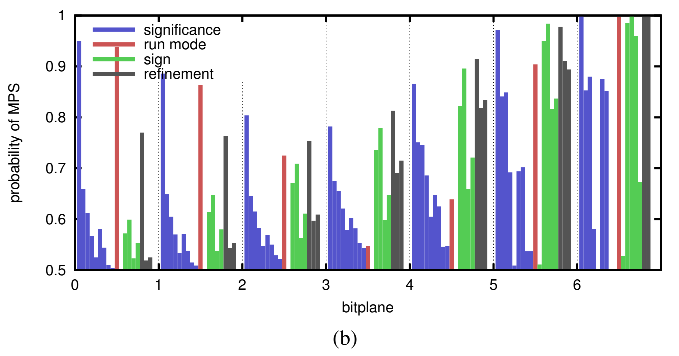
			parallelism level		
			cblk.	cod. pass	coef.
COMPLIANT	JP2	enc.	✓	✗	✗
		dec.	✓	✗	✗
	JP2 +RESET	enc.	✓	✓	✗
		dec.	✓	✗	✗
	JP2 +RESET +CAUSAL	enc.	✓	✓	✗
		dec.	✓	✓*	✗
NON COMPLIANT	stationary	enc.	✓	✓	✓
		dec.	✓	✗	✗
	stationary +CAUSAL	enc.	✓	✓	✓
		dec.	✓	✓*	✗
	stationary +ctxt. mod.	enc.	✓	✓	✓
		dec.	✓	✗	✓

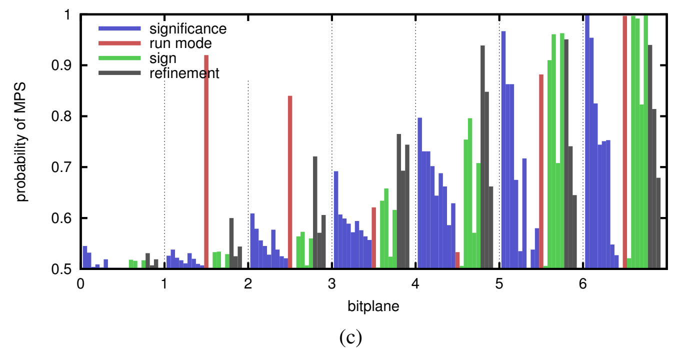
In the decoder, the probability adaptation is also easily implemented in parallel, by coding pass. However coding pass parallelization is complicated by the context formation process, which requires certain bits from the eight neighboring coefficients to be decoded before the context can be formed for the current bit. The bits needed from the neighboring coefficients include all bits from previous bitplanes, those from the previous coding passes of the current bitplane, and those from the previous coefficients scanned in the current coding pass. Parallel coding pass decoding is thus only possible if a tightly synchronized delay of (slightly more than) one stripe is introduced between the decoding of subsequent coding passes. However, the number of coding passes is commonly higher than the number of stripes, rendering impractical such a strategy.

JPEG2000 overcomes this barrier to coding pass parallelism via the CAUSAL coding variation. When this variation is in use, the context formation process considers only neighbors within the same stripe. In other words, when the context window shown in Fig. 1 trespasses stripe boundaries, all coefficients from trespassed stripes are assumed to be insignificant. Since there are then no dependencies between stripes, the decoder can parallelize coding passes by introducing a delay of two columns (within the same stripe) between the threads

[1] From the point of view of the (arithmetic) entropy coder, further measures are required. This is discussed in more detail below.







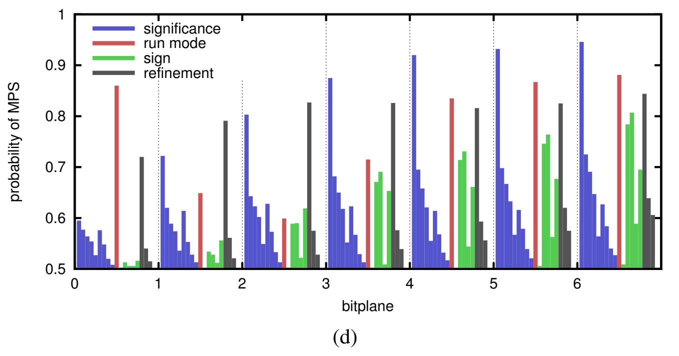


Fig. 5: Lookup tables for the four corpora used in Section IV. Data correspond to the low vertical-, high horizontal-frequency subband of the first decomposition level produced by the irreversible 9/7 wavelet transform. Subfigures correspond to (a) natural images, (b) aerial images, (c) XRAY angiographies, and (d) hyperspectral AVIRIS images. Probabilities for the least significant seven bitplanes are depicted.

that process two consecutive coding passes. This ensures that all required neighboring bits have been decoded in time to form contexts in the current coding pass.

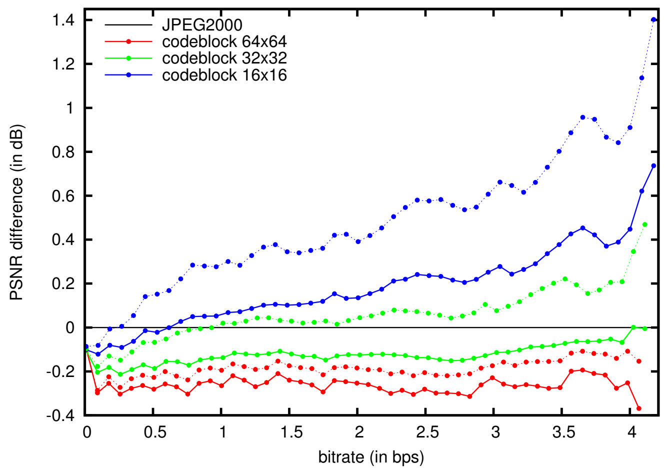
Even when the coding variations discussed above are employed, JPEG2000 cannot provide parallelism at the coefficient level because of the adaptive probability model. On the other hand, the stationary probability model described before permits parallelism at both the coding pass and coefficient levels at the encoder. This is true even when the CAUSAL coding variation is not employed.[2] At the decoder, neither of these levels of parallelism are possible in the absence of the CAUSAL coding variation due to the context formation approach. As before, use of the CAUSAL coding variation allows coding pass parallelism in the decoder through the tight synchronization strategy described before. However, coefficient level parallelism is still not possible.

Parallelism at the coefficient level in the decoder can only be achieved if the stationary probability model is accompanied by a slight modification of the context formation process in JPEG2000. As noted above, the formation of the contexts considers coefficients that become significant in previous coding passes and those that are scanned before the current coefficient in the current coding pass. Coefficients cannot be processed in parallel unless this condition is removed. The last strategy mentioned in Table I modifies the context formation of JPEG2000 by computing the significance state considering only the bits reconstructed in previous coding passes. This enables coefficient parallelism at the decoder by synchronizing the threads that process (all) coefficients to code each coding pass in a synchronized manner, i.e., one thread coding a coefficient cannot code the next coding pass until all its neighbors have finished coding the current coding pass.

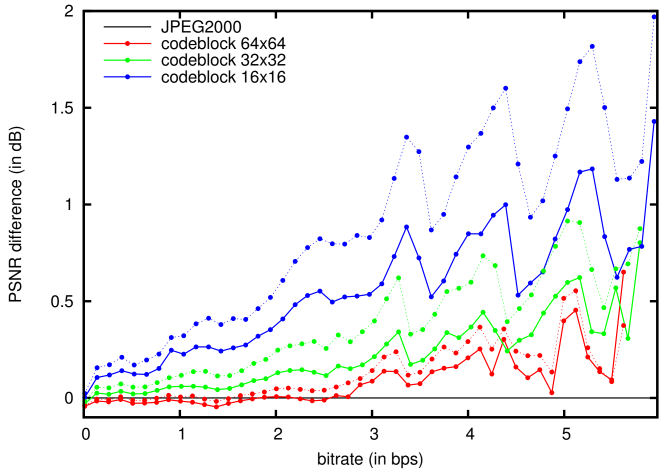
As noted in footnote 1 and as described in more detail in [18], the use of coding pass parallelism also requires the RESTART coding variation, which causes the arithmetic coder to terminate its compressed bitstream at the end of each coding pass. This eliminates dependencies between the bitstreams of different coding passes. As mentioned previously, this suffices for coding pass level parallelism, but not for coefficient level parallelism. As discussed in the Introduction, achieving coefficient level parallelism will require changing or replacing the entropy coding procedure. In this paper, we focus on modeling (probability models and context formation) to support parallelization, and remark that the potential gains of a fully coefficient parallel implementation of the bitplane coding stage, including both modeling and coding, are significant. Implementations of the wavelet transform that employ coefficient level parallelism, for instance, achieve speedups of 10 or more with respect to a sequential implementation [29]. Similar accelerations may be possible for the bitplane coding stage.

Though not originally devised for parallelism purposes, the

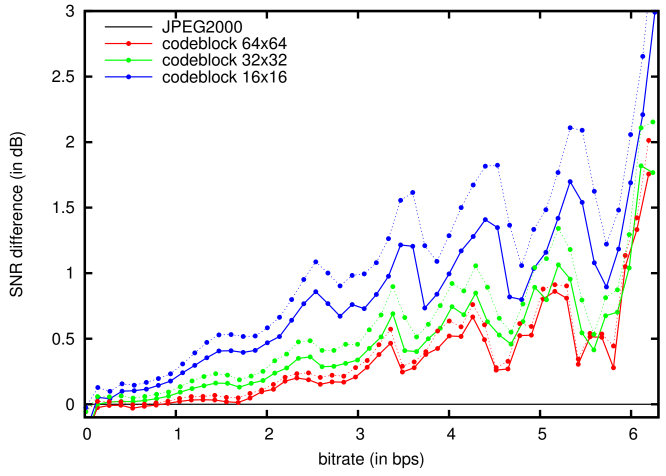
[2]In the case of the stationary probability model, the RESET coding variation has no meaning, since probabilities are not adapted.



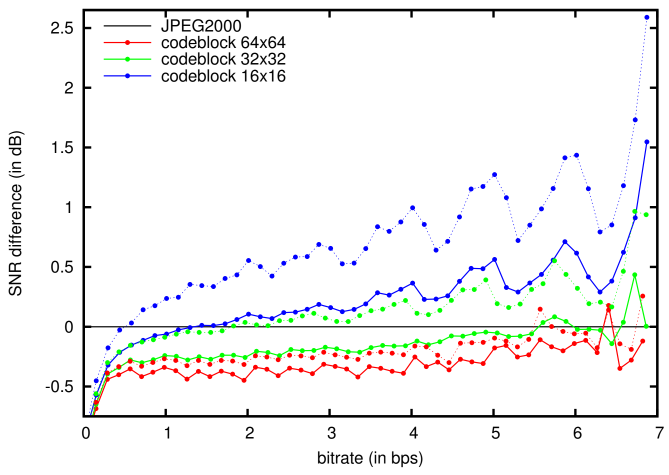
(a)



(b)



(c)



(d)

Fig. 6: Evaluation of the coding performance achieved by the proposed stationary probability model as compared to JPEG2000 (solid plots) and to JPEG2000 with the RESET coding variation (dotted plots). Each subfigure reports the performance achieved for one image from a specific corpus: (a) natural, (b) aerial, (c) XRAY, and (d) AVIRIS.

BYPASS coding variation may also be employed to parallelize certain coding passes. When this coding variation is in use, the bits emitted by the coding engine in some coding passes are not fed to the arithmetic coder but are directly transmitted in raw mode. Parallelization of the coding of these raw coding passes is straightforward since they neither require context formation nor probability models. However, we note that using BYPASS alone does not facilitate parallelism in general, since in most coding passes adaptive probabilities are still used.

IV. Experimental results

Four corpora of images are employed to evaluate the proposed strategies. The first corpus consists of the eight ISO12640-1 images (2048×2560, gray scale, 8 bits per sample (bps)). The second is composed of four aerial images provided by the Cartographic Institute of Catalonia, covering vegetation and urban areas (7200×5000, gray scale, 8 bps). The third corpus has three XRAY angiography images from the medical community (512×512 with 151 components, 12 bps). The last corpus contains three AVIRIS (Airbone Visible/Infrared Imaging Spectrometer) hyperspectral images provided by NASA (512×512 with 224 components, 16 bps).

The results reported in this section employ the unmodified MQ arithmetic coder as used in JPEG2000. This provides an equal footing for the comparison of different context and probability modeling strategies. As mentioned previously, a fully parallelized implementation of JPEG2000 would require other ingredients such as a parallelized wavelet transform and parallelized entropy coder. Parallelizable entropy coding may result in further changes to compression efficiency, likely similar to the RESET and the CAUSAL coding variations (around 2% in general). However, due to the separation between modeling and coding, these effects can be considered separately. Thus, the focus here is on the gains or losses that can be obtained via stationary probability modeling, all other things being equal.

Fig. 6 reports the lossy coding performance achieved for one image from each corpus. The LUT employed in each case is constructed using all images of the same corpus except the image that is evaluated. Coding images of a different type from that used to construct a given LUT may decrease the coding performance significantly. Constructing a single LUT based on images from all corpora also degrades performance obtained for each corpus individually. On the other hand, the size of the LUT is typically quite small. For example, for the four corpora described above, the LUTs represent 0.05, 0.008, 0.007, and 0.004, respectively. Thus, the overhead that would be incurred by including a specialized LUT in a codestream would be negligible.

The JPEG2000 lossy mode, which uses the irreversible 9/7 wavelet transform, is employed for all results reported in Fig. 6. The proposed context formation modification is not used. The horizontal axis of each subfigure is the rate at which the images are coded, whereas the vertical axis is the peak signal-to-noise ratio (PSNR) difference achieved between the proposed method and that of JPEG2000. Values above 0 indicate that the proposed method achieves higher PSNR than that of JPEG2000. Solid plots compare the proposed method to JPEG2000 without any coding variations, whereas dotted plots compare the proposed method to JPEG2000 using the RESET coding variation. Results are reported for three different codeblock sizes.

For natural and AVIRIS images, results indicate that the stationary probability model achieves slightly inferior coding performance to that of JPEG2000 (without coding variations) when the codeblock size is 32×32 or larger. For smaller codeblock sizes, the stationary probability model achieves significantly higher PSNR than that of JPEG2000. This is caused by the adaptive mechanisms of JPEG2000 not having enough data to adjust probabilities reliably for small codeblocks, while the proposed method does not depend on the amount of data coded per codeblock. For the aerial and XRAY images, the proposed method achieves higher coding performance than that of JPEG2000 at most rates, even for codeblocks of size 64×64. The gain obtained by the proposed model is significant, being of 2 dB or more at some rates.

When the RESET mode is in use, the gains achieved by the proposed method increase as the codeblock size is decreased. Indeed, the proposed method becomes superior for all images when the RESET mode is employed for codeblock sizes of 32×32 or smaller. It is worth noting that the relative improvement when the RESTART variation is employed is due to changes in JPEG2000, and not to changes in the proposed scheme (see footnote 2). Specifically, adaptive probabilities penalize JPEG2000 coding performance further due to the lack of data coded in each individual coding pass. Results are similar for other images of the corpora. Specific results for the CAUSAL coding variation are not reported. Typical penalties incurred by this mode are less than 0.01 bps.

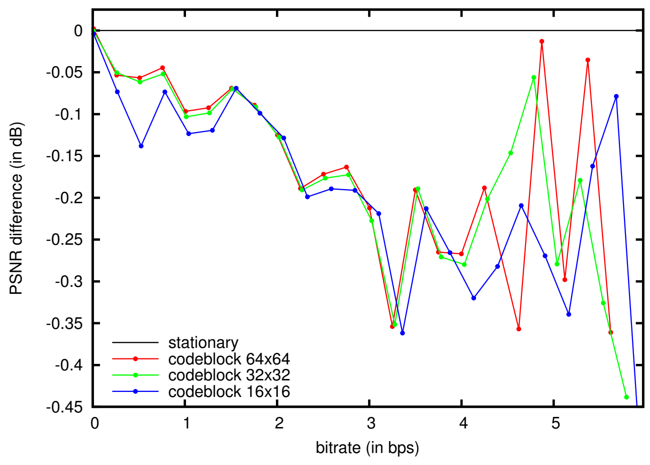
The next test assesses coding performance for the lossless mode of JPEG2000. This mode employs the reversible 5/3 wavelet transform and codes all bitplanes of the resulting (integer) transform coefficients. Five different codeblock sizes are employed. Table II reports the results achieved for each of the four corpora. Results for JPEG2000 are computed by averaging the rate required to losslessly compress each image of a corpus. Results for JPEG2000 with RESET and for the proposed stationary probability model are reported as the difference in bps between the evaluated strategy and JPEG2000. Negative values indicate that the codestream generated for the indicated strategy is shorter than that of JPEG2000. As expected, the RESET coding variation always produces positive values (increases the length of the JPEG2000 codestream).

TABLE II: Evaluation of lossless coding performance. Results are averaged over all images in each corpus.

	cblk. size	*JP2*	*JP2 +RESET*	*stationary*
ISO 12640-1	64×64	**4.71** bps	+0.02	+0.04
	64×32	**4.73** bps	+0.03	+0.02
	32×32	**4.76** bps	+0.05	**0.00**
	32×16	4.82 bps	+0.07	**-0.04**
	16×16	4.92 bps	+0.09	**-0.10**
	max diff.	0.21	0.28	**0.07**
Aerial	64×64	5.80 bps	+0.02	**-0.04**
	64×32	5.82 bps	+0.03	**-0.05**
	32×32	5.85 bps	+0.05	**-0.07**
	32×16	5.91 bps	+0.08	**-0.11**
	16×16	6.01 bps	+0.11	**-0.16**
	max diff.	0.21	0.3	**0.09**
XRAY	64×64	6.40 bps	+0.01	**-0.14**
	64×32	6.42 bps	+0.02	**-0.16**
	32×32	6.45 bps	+0.04	**-0.17**
	32×16	6.51 bps	+0.06	**-0.21**
	16×16	6.61 bps	+0.08	**-0.26**
	max diff.	0.21	0.28	**0.11**
AVIRIS	64×64	7.19 bps	+0.02	**-0.05**
	64×32	7.21 bps	+0.04	**-0.07**
	32×32	7.24 bps	+0.06	**-0.09**
	32×16	7.30 bps	+0.09	**-0.12**
	16×16	7.40 bps	+0.12	**-0.18**
	max diff.	0.21	0.31	**0.08**

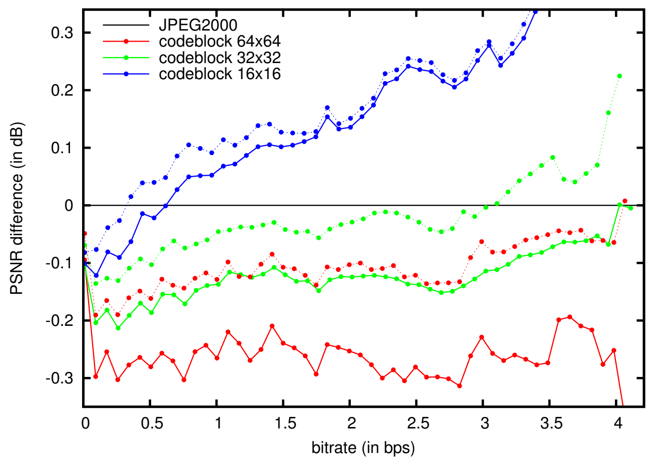
The best result in each row of the table is emphasized with bold font. The proposed method achieves the best results in all cases except when natural images are compressed using large codeblock sizes. The images with the greatest differences come from the XRAY corpus, for which the proposed method decreases the length of the codestream by up to 0.26 bps. It is worth emphasizing the significance of these results: the proposed method provides more parallelism than the RESET variation while enhancing coding performance. In addition, it penalizes the use of small codeblocks less than the other methods presented in the table. This can be seen in the last row corresponding to each corpus, which reports the difference between the codestream generated when using codeblocks of size 64×64 and that generated with codeblocks of 16×16. The differences for the stationary probability model are half or less of those for JPEG2000.

The modification of the context formation process as de-



(a)

Fig. 7: Coding performance for the proposed stationary probability model when the context formation modification is used. Results are for the same aerial image used for Fig. 6(b).



(a)

Fig. 8: Coding performance for the proposed stationary probability model when context $\phi = 9$ (run mode) uses stationary (solid lines) and adaptive (dotted lines) probabilities, as compared to JPEG2000. Results are for the same natural image used for Fig. 6(a).

scribed in Section III permits coefficient level parallelism at the decoder, but degrades coding performance somewhat. Fig. 7 reports the difference between the PSNR achieved by the proposed method with and without the context modification. The degradation in performance is similar for all codeblock sizes, being approximately 0.2 and 0.4 dB, respectively for medium and high rates. Results vary slightly depending on the image type, though decreases in PSNR are never more than 0.5 dB.

As a final remark, we point out that, as discussed in previous sections, the adaptive mechanisms employed by JPEG2000 are most effective to adjust probabilities between bitplanes. Empirical evidence reveals that this statement holds for all contexts except $\phi = 9$ (i.e., that employed in the run mode of CP). For this context, some gain can be had by allowing probabilities to adapt within a bitplane. This can be seen in Fig. 8, which depicts the coding performance achieved by the proposed method when only context $\phi = 9$ employs adaptive probabilities, for the same image reported in Fig. 6(a). For the purpose of comparison, the figure also reports the performance for the unmodified stationary model, i.e., the same results reported in Fig. 6(a). The unmodified strategy is shown by the solid lines in the figure, while the dotted lines report the performance achieved when context $\phi = 9$ employs adaptive probabilities. Note that for codeblocks of size 64×64 and 32×32, the gain in PSNR can be significant. Evidently, adopting this strategy would forfeit the advantages of the stationary model.

V. Conclusions

The aim of this work is to explore coding strategies for the parallelization of the data coding stage of modern wavelet-based image codecs. The main obstacle to achieve the finest level of parallelism, in which all coefficients are processed in parallel, is the context-adaptive mechanism employed to determine probabilities of emitted symbols. We overcome this obstacle with a stationary probability model that, instead of adjusting probabilities as symbols are emitted, establishes probability estimates beforehand. With this probability model, parallelism at the coefficient level is attainable.

Evaluation of the stationary probability model indicates that it achieves similar coding performance as that obtained with conventional context-adaptive approaches when medium and large codeblock sizes are employed. For small codeblock sizes, the proposed model significantly improves the coding performance over classical approaches.

References

[1] Y. Yoo, A. Ortega, and B. Yu, “Image subband coding using context-based classification and adaptive quantization,” *IEEE Trans. Image Process.*, vol. 8, no. 12, pp. 1702–1715, Dec. 1999.

[2] D. Taubman, “Remote browsing of JPEG2000 images,” in *Proc. IEEE International Conference on Image Processing*, Sep. 2002, pp. 229–232.

[3] F. Auli-Llinas and M. W. Marcellin, “Scanning order strategies for bitplane image coding,” *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1920–1933, Apr. 2012.

[4] 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.

[5] M. Slattery and J. Mitchell, “The Qx-coder,” *IBM Journal of Research and Development*, vol. 42, no. 6, pp. 767–784, Nov. 1998.

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

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

[8] *High Efficiency Video Coding Standard*, International Telecommunication Union Std. H.265, 2013.

[9] *Information technology - Lossy/lossless coding of bi-level images*, ISO/IEC Std. 14 492, 2001.

[10] Y.-K. Chen, C. Chakrabarti, S. Bhattacharyya, and B. Bougard, “Signal processing on platforms with multiple cores: Part 1 - overview and methodologies,” *IEEE Signal Process. Mag.*, vol. 26, no. 6, pp. 1–2, Nov. 2009.

[11] J. Sole, R. Joshi, N. Nguyen, T. Ji, M. Karczewicz, G. Clare, F. Henry, and A. Duenas, “Transform coefficient coding in HEVC,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1765–1777, Dec. 2012.

[12] Y.-K. Chen, C. Chakrabarti, S. Bhattacharyya, and B. Bougard, “Signal processing on platforms with multiple cores: Part 2 - applications and designs,” *IEEE Signal Process. Mag.*, vol. 27, no. 2, pp. 1–2, Mar. 2010.

[13] X. Peng, J. Xu, Y. Zhou, and F. Wu, “Highly parallel line-based image coding for many cores,” *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 196–206, Jan. 2012.

[14] L. Santos, E. Magli, R. Vitulli, J. F. Lopez, and R. Sarmiento, “Highly-parallel GPU architecture for lossy hyperspectral image compression,” *IEEE J. Sel. Topics Appl. Earth Observations Remote Sens.*, vol. 6, no. 2, pp. 670–681, Apr. 2013.

[15] S. M. LoPresto, K. Ramchandran, and M. T. Orchard, “Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework,” in *Proc. IEEE Data Compression Conference*, Mar. 1997, pp. 221–230.

[16] 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.

[17] 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.

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

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

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] M. J. Weinberger, J. J. Rissanen, and R. B. Arps, “Applications of universal context modeling to lossless compression of gray-scaled images,” *IEEE Trans. Image Process.*, vol. 5, no. 4, pp. 575–586, Apr. 1996.

[25] 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.

[26] T.-T. Wong, C.-S. Leung, P.-A. Heng, and J. Wang, “Discrete wavelet transform on consumer-level graphics hardware,” *IEEE Trans. Multimedia*, vol. 9, no. 3, pp. 668–673, Apr. 2007.

[27] C. Tenllado, J. Setoain, M. Prieto, L. Pinuel, and F. Tirado, “Parallel implementation of the 2D discrete wavelet transform on graphics processing units: Filter bank versus lifting,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, no. 3, pp. 299–310, Mar. 2008.

[28] J. Matela, “GPU-Based DWT acceleration for JPEG200,” in *Annual Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, Jan. 2009, pp. 136–143.

[29] W. J. van der Laan, A. C. Jalba, and J. B. Roerdink, “Accelerating wavelet lifting on graphics hardware using CUDA,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 1, pp. 132–146, Jan. 2011.

[30] S. Chen, S. Chen, and S. Sun, “P3-CABAC: A nonstandard tri-thread parallel evolution of CABAC in the manycore era,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 6, pp. 920–924, Jun. 2010.

[31] W. Hu, J. Wen, W. Wu, Y. Han, S. Yang, and J. Villasenor, “Highly scalable parallel arithmetic coding on multi-core processors using LDPC codes,” *IEEE Trans. Commun.*, vol. 60, no. 2, pp. 289–294, Feb. 2012.

[32] “Digital cinema system specification,” Tech. Rep. Version 1.2, 2008. [Online]. Available: http://www.dcimovies.com

[33] 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*, vol. 239, no. 1, pp. 266–282, Aug. 2013.

Francesc Aulí-Llinàs (S’06-M’08) is a Ramón y Cajal Fellow with the Department of Information and Communications Engineering, Universitat Autònoma de Barcelona, Spain. He received the B.Sc., B.E., M.Sc., and Ph.D. degrees in computer science from Universitat Autònoma de Barcelona in 2000, 2002, 2004, and 2006, respectively. Since 2002, he has been consecutively funded 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 D. Taubman, University of New South Wales, Australia, and with the group of M. Marcellin, University of Arizona, USA. He develops and maintains BOI, a free-source JPEG2000 implementation. In 2000 and 2002, he received two awards of Bachelor given to the first students of the promotion. In 2006, he was awarded with a free software mention from the Catalan Government for the development of BOI. In 2013, he was awarded with a distinguished R-Letter given by the IEEE Communications Society for a paper co-authored with M. Marcellin. He is reviewer for various magazines and symposiums and has authored numerous papers in the top journals and conferences. His current research interests include a wide range of image coding topics, including highly scalable image and video coding systems, rate-distortion optimization techniques, parallel architectures for image and video coding, distortion estimation, embedded quantization, and interactive image and video transmission.

Michael W. Marcellin (S’81-M’87-SM’93-F’02) 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. Dr. Marcellin is a major contributor to JPEG2000, the emerging second-generation standard for image compression. He is coauthor of the book, *JPEG2000: Image compression fundamentals, standards and practice,* Kluwer Academic Publishers, 2002. Dr. 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.