Mutual Information-Based Analysis of JPEG2000 Contexts

Zhen Liu and Lina J. Karam, *Senior Member, IEEE*

***Abstract*—Context-based arithmetic coding has been widely adopted in image and video compression and is a key component of the new JPEG2000 image compression standard. In this paper, the contexts used in JPEG2000 are analyzed using the mutual information, which is closely related to the compression performance. We first show that, when combining the contexts, the mutual information between the contexts and the encoded data will decrease unless the conditional probability distributions of the combined contexts are the same. Given I, the initial number of contexts, and F, the final desired number of contexts, there are $S(I, F)$ possible context classification schemes where $S(I, F)$ is called the Stirling number of the second kind. The optimal classification scheme is the one that gives the maximum mutual information. Instead of using an exhaustive search, the optimal classification scheme can be obtained through a modified generalized Lloyd algorithm with the relative entropy as the distortion metric. For binary arithmetic coding, the search complexity can be reduced by using dynamic programming. Our experimental results show that the JPEG2000 contexts capture the correlations among the wavelet coefficients very well. At the same time, the number of contexts used as part of the standard can be reduced without loss in the coding performance.**

***Index Terms*—Context-based arithmetic coding, JPEG2000, mutual information.**

I. Introduction

DATA compression is the process of reducing the amount of data required to represent a given quantity of information. Data is a means by which the information is conveyed. Yet, data≠information. In a lot of cases, there is redundancy within the data. The basis for compression is redundancy removal. In digital image compression, three types of redundancy are commonly exploited. These are interpixel redundancy, psycho-visual redundancy, and statistical redundancy. Corresponding to these redundancies removal, a typical transform-based lossy image compression scheme has three stages: a reversible transform stage, an irreversible quantization stage, which is usually controlled by rate allocation, and a lossless entropy-coding stage. Recently, context-based artithmetic coding has been widely adopted in the final entropy-coding stage for image and video compression [1]–[13]. In particular, context-based arithmetic coding is a key component of the new JPEG2000 image compression standard.

Manuscript received April 30, 2003; revised March 5, 2004. This work was supported in part by the National Science Foundation under Grant CCR-9733897. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Thrasyvoulos N. Pappas.

Z. Liu is with the Qualcomm, Inc., San Diego, CA 92121-1714 USA (e-mail: zhenl@qualcomm.com).

L. J. Karam is with the Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287-5706 USA (e-mail: karam@asu.edu).

Digital Object Identifier 10.1109/TIP.2004.841199

In theory, the contexts provide a model for probability estimation of each possible message to be coded, and the arithmetic coder translates the estimated probabilities into bits. The "codeword" that the arithmetic coder assigns to each possible message consists of a number of bits that is needed to distinguish the half-open subinterval determined by the corresponding message from all other possible subintervals in $[0.0, 1.0)$. The more probable message is represented using a larger subinterval and, therefore, results in a shorter codeword. In practice, the probability is estimated incrementally and the subinterval that is associated with the current message, is refined incrementally using the probability of individual symbols, with bits outputted as each symbol comes in. The goal of context modeling is to provide the statistical probability information to the coder. Essentially, it is the quality of the context model that determines the compression efficiency. In JPEG2000, 17 contexts are specified to efficiently code the significant, sign, and refinement information. This research is motivated by the question why these 17 contexts are selected. In this paper, we try to answer this question and analyze the JPEG2000 contexts using the mutual information.

This paper is organized as follows. In Section II, we first show that a very important problem in wavelet image coder design is how to model and utilize the various within and cross-subband correlations left among the wavelet transformed coefficients. Context modeling and adaptive arithmetic coding is one approach to capture these correlations and this approach has been widely adopted [7]–[10]. In Section III, an overview of entropy coding is given and an expression of the total entropy-coding efficiency is derived. In Section IV, we propose that the mutual information between the contexts and the encoded data can be used to measure the context modeling optimality. When combining contexts, the mutual information will decrease unless the conditional probability distributions of the combined contexts are the same. Instead of using exhaustive search, the optimal classification can be obtained through a modified generalized Lloyd algorithm (GLA) with the relative entropy as the distortion metric. For binary arithmetic coding, dynamic programming can be implemented to reduce the search complexity. Since we are dealing with binary random variables in the JPEG2000 standard, dynamic programming is then applied to study the performance of JPEG2000 contexts. Experimental results are reported in Section V. Finally, we conclude the paper in Section VI.

II. Context Modeling for Wavelet Image Compression

Since its introduction as a tool for signal representation, the wavelet transform has become very popular in the image-coding community. It has such desirable features as space-frequency

(a) (b)

Fig. 1. Illustration of various within-subband and cross-subband correlations within the wavelet transformed coefficients. (a) Original 512×512 Lena image. (b) Wavelet transformed coefficients after three levels of wavelet decomposition using $9/7$ filters [14].

localization, energy compaction, and multiresolution analysis. Many state-of-the-art image coders [1]–[10] employ the *discrete wavelet transform* (DWT) in their algorithm, which is also employed in the new JPEG2000 image compression standard.

In the early wavelet image coders [14]–[16], the DWT coefficients were assumed to be independent. The histogram of the DWT coefficients can be modeled by a *generalized Gaussian distribution* (GGD). This implies that the DWT coefficients are sparse, with only a small number of large coefficients and a large number of small coefficients. Based on this observation, early wavelet image coders were mainly concerned with designing better quantization and bit allocation strategies. The drawback is apparent in that those near zero-valued wavelet coefficients, which convey little information, must be represented and coded, which can consume a large portion of the bit budget. Although these types of coders provide superior visual quality by eliminating the blocking effect in comparison to block-based image coders such as JPEG, their objective performance measured by the *peak signal-to-noise ratio* (PSNR) increases only moderately.

At the same time, a simple check of the wavelet transform coefficients reveal that the wavelet transform can not totally decorrelate real-world signals. There are still various dependencies between coefficients in different subbands. The lowest frequency subband looks like the original image, and most of the energy in the high-frequency subband is concentrated in the edge areas that correspond to large transitions in the original image as shown in Fig. 1.

A lot of research have been done to characterize these dependencies and model the wavelet coefficients. In [17], a GGD with autoregressive dependencies between neighboring coefficients (both within and across scales) is used to model the wavelet coefficients. In [18], the wavelet coefficients are modeled by a generalized Laplacian distribution; the local neighbors are used to estimate the model parameters. In [19], an explicit conditional probability model is built to capture these within and cross-subband correlations; this model is then used to build an *embedded predictive wavelet image coder* (EPWIC). In [20] and [21], a *hidden Markov tree* (HMT) is used to capture the joint *probability density function* (PDF) of the wavelet coefficients. The non-Gaussian PDF of wavelet coefficients are modeled as a two-component Gaussian mixture. The components are labeled by a hidden state signifying whether the coefficient is small or large. By linking these hidden states across scales in a Markov tree, the cross-subband correlations are captured.

These statistical properties and their exploitation are very important for the image coder design. The success of recent wavelet image coders can be mainly attributed to the innovative strategies for data organization and representation that exploit these statistical dependencies one way or the other.

In Shapiro's *embedded zerotree* (EZW) [1] and in Said and Pearlman's *set partitioning in hierarchical tree* (SPIHT) [2], a zerotree structure is introduced to exploit the cross-subband similarity of wavelet coefficients. Xiong *et al.*'s *space-frequency quantization* (SFQ) [3] goes one step forward by incorporating the zerotree quantizer with the scalar quantizer and using an iterative algorithm to reach the optimal rate allocation between the two types of quantizers. In Servetto *et al.*'s *morphological representation of wavelet data* (MRWD) [4], morphological operations are used to exploit the within-subband clustering of the wavelet coefficients. Chai *et al.*'s *significant-linked connected component analysis* (SLCCA) [5] strengthens MRWD by exploiting not only the within-subband clustering of wavelet coefficients but also cross-subband similarity in the significant fields. In Joshi *et al.*'s [6], the within-subband clustering property is exploited to classify the blocks into several classes within each subband and code them at different rates using *arithmetic-coded trellis-coded quantization* (ACTCQ); the cross-subband similarity is exploited to code the classification maps.

At the same time, in Wu's *embedded conditional entropy coding of wavelet coefficients* (ECECOW) [22], ECECOW

using Fisher discriminant [7], Chrysafis's *context-based entropy coding* (C/B) [23], Li's *rate-distortion optimized embedding* (RDE) [9], Taubman's *embedded block coding with optimized truncation* (EBCOT) [8], it is shown that these within-subband and cross-subband dependencies can be effectively utilized by using context modeling and adaptive arithmetic coding. In fact, in all the above-mentioned image-coding algorithms, context-based arithmetic coding is adopted in the final entropy-coding stage. Although context-based entropy coding is widely used, contexts are usually formed heuristically.

In ECECOW, high-order context modeling is used. It uses not only the already coded neighboring coefficient's energy level, but also the spatial structure to form the context template C. The context template formation also depends on the subband orientation. This high-dimension context template C is then projected into a lower dimensional space using a linear projector. Quantization is then applied on the projected space. The projector used in [22] is determined by linear regression. In [7], Fisher discriminant is used. The quantization scheme used in [7] is based on dynamic programming. In C/B [23], the context template C is based on the energy level of 12 neighboring coefficients in the current subband and the spatially corresponding coefficients in the parent subband. A linear projector is used to project the high-dimension C into a scalar. Lloyd–Max quantization is then applied in the projected space to obtain a finite and sufficiently small number of contexts. In RDE [9], 128 contexts are formed using the significance status of six spatially neighboring coefficients in the current subband and the spatially corresponding coefficients in the parent subband.

In JPEG2000, a two-tier coding approach is adopted. In tier-1 coding, after dc level shifting, DWT, and quantization, the samples in each subband are partitioned into code blocks. Each code block is then independently bitplane coded from the most significant bitplane (MSB) to the least significant bitplane (LSB). Each bitplane is fractionally coded using three coding passes. In the three passes, four primitives are used to code the sample's value in the current bitplane (0 or 1) and the sample's sign (positive or negative) when the sample becomes significant. In the significance propagation pass, the zero-coding (ZC) primitive is used to code the current bitplane value of those samples that are insignificant and have at least a significant neighbor. In the magnitude refinement pass, the magnitude-refinement (MR) primitive is used to code the current bitplane value of those samples that have already become significant in the previous bitplane. In the cleanup pass, a combination of ZC and run-length coding (RLC) primitives are used to code the current bitplane value of the remaining samples. If a sample becomes significant, the sign-coding (SC) primitive is used to code the sample's sign immediately. In all cases, a binary valued symbol is coded using context-based arithmetic coding. In this way, an embedded bitstream is generated for each code block. At the same time, the rate increase and the distortion reduction associated with each coding pass is recorded. This information is then used in the post compression rate control to determine each coding block's contribution to different quality layers. Given the rate allocation result, the final bitstream is formed in tier-2 coding. In embedded bitplane coding, context-based binary arithmetic coding is extensively used. In JPEG2000, 17 contexts are specified to code the significant, sign, and refinement information. A fairly natural question to ask is why these 17 contexts are selected. Why not other classification schemes? In addition, since the goal of JPEG2000 is to create a unified compression standard (lossy and lossless) for different types of still images with different characteristics (natural, scientific, medical, military, text, and compound), do the 17 chosen contexts work very well for all these different diverse requirements? To answer these questions, Section III first gives an overview of basic entropy-coding concepts.

III. Entropy-Coding Analysis

Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ be a random process that takes on, at each time i, one value from a finite alphabet of possible values. Let message $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be a realization of such random process where each symbol x_i, is an observation of the corresponding discrete random variable X_i.

When entropy coding such a message $\mathbf{x}$, the minimum possible coded length in bits is $-\log_2 p(\mathbf{x}) = -\log_2 p(x_1 x_2 \ldots x_n)$. The lower bound of the average number of bits that will be used to code the messages generated by random process $\mathbf{X}$ is given by the entropy $H(\mathbf{X}) = H(X_1 X_2 \ldots X_n)$. To achieve this lower bound, we need to calculate the probabilities of the whole message and code jointly the entire message.

Such global probabilities are very difficult to calculate and the complexity grows exponentially with the message length. This procedure needs a very large buffer and incurs a very long coding delay. To solve these issues in practice, one approach is to break the coding procedure into a series of steps as indicated by the chain rule of the entropy

$$H(X_1 X_2 \ldots X_n) = \sum_{i=1}^{n} H(X_i | X^{i-1}), \quad \text{where } X^{i-1} = X_1 X_2 \ldots X_{i-1}. \tag{1}$$

This implies that entropy coding can be performed sequentially one symbol at a time. The corresponding minimum coded length can be expressed as $-\sum_{i=1}^{n} \log_2 p(X_i | x^{i-1})$, where $x^{i-1} = x_1 x_2 \ldots x_{i-1}$. In this way, the entire preceding message forms a context for predicting X_i. This approach does not help unless the conditional probability $p(X_i | x^{i-1})$ is easier to calculate than the overall probability $p(X_1 X_2 \ldots X_n)$. In most cases, this conditional probability is also very difficult to get or is unknown. So, we need to approximate the conditional probabilities. One way of approximation is to estimate the probabilities based on a finite number of previous symbols in the message. A more general way of approximation is to assign each possible string x^{i-1} to some conditioning class; in this case, the conditional probability can be formally expressed as $p(X_i | x^{i-1}) \approx p(X_i | f(x^{i-1}))$, where $f(\cdot)$ is the function that maps each possible x^{i-1} to a conditioning context. The range of the values that function $f(\cdot)$ returns is the context set $\tilde{C}$.

Therefore, sequential entropy coding can be separated into two parts: a modeler and a coder. At each time instance i, the modeler tries to estimate $p(X_i | f(x^{i-1}))$, the probability of the next symbol to be coded based on the observed context. Because the contexts are formed by the already coded symbols and

the entropy coding is lossless, the same context is also available at the decoder, so no side information need to be transmitted. Given the estimated $\hat{p}(X_i|f(x^{i-1}))$, an ideal entropy coder places $-\log_2(\hat{p}(X_i = k|f(x^{i-1})))$ bits onto its output stream if $X_i = k$ actually occurs. This ideal codeword length can be achieved by using arithmetic coding [24]–[26].

The average number of bits that will be used to code X_i is given by

$$\sum_{k=1}^{N} p(X_i = k|f(x^{i-1})) \log_2 \left(\frac{1}{\hat{p}(X_i = k|f(x^{i-1}))} \right) \quad (2)$$

$$= \sum_{k=1}^{N} p(X_i = k|f(x^{i-1})) \log_2 \left(\frac{1}{p(X_i = k|f(x^{i-1}))} \frac{p(X_i = k|f(x^{i-1}))}{\hat{p}(X_i = k|f(x^{i-1}))} \right) \quad (3)$$

$$= H(p(X_i|f(x^{i-1}))) + D(p(X_i|f(x^{i-1})) \,||\, \hat{p}(X_i|f(x^{i-1}))) \quad (4)$$

where $D(\cdot \,||\, \cdot)$ is the relative entropy between two distributions, and is also known as the Kullback–Leibler distance and is calculated as follows:

$$D(p(x) \,||\, q(x)) = \sum_x p(x) \log_2 \frac{p(x)}{q(x)}. \quad (5)$$

From information theory [27], $D(p(x) \,||\, q(x)) \geq 0$ with equality if and only if $p(x) = q(x)$. Therefore, for efficient coding, we hope the modeler can give an accurate estimate of $p(X_i|f(x^{i-1}))$. Otherwise, using the estimated distribution incurs a penalty of $D(p(X_i|f(x^{i-1})) \,||\, \hat{p}(X_i|f(x^{i-1})))$ on the average code length.

The estimated probabilities $\hat{p}(X_i|f(x^{i-1}))$ can be precomputed through training and are available to both the encoder and decoder before the actual coding starts. In this case, a static model is used. Alternatively, these estimates can be computed and updated on the fly based on the past coded symbols. This is the adaptive model. Practical applications generally require adaptive, on-line estimation of these probabilities either because sufficient statistical knowledge is not available or because these statistics are time varying. One very natural way of estimating the probabilities online is to count the number of times each symbol has occurred under a certain context. The estimates at the encoder are updated with each encoded symbol, and the estimates at the decoder are updated with each decoded symbol. This universal modeling approach requires no *a priori* knowledge about the data to be coded, and the coding can be completed in only one pass. So, it is widely adopted. In addition, it is shown in [26] that an adaptive model can perform only a little bit worse than the best possible static model (one whose statistics are derived from the message itself, which requires two pass coding and transmission of side information), and an adaptive model may perform much better than a static model if the message to be coded is different from what the static model is expecting.

We expect these probability distributions to be nonuniform with one or a few of the symbols having much higher probability than the others, which allows prediction based on the observed context. In fact, entropy coding achieves compression just by exploiting this nonuniformity or prediction property. The more accurate the prediction, the better the compression performance.

In summary, the process of entropy coding a message is usually done incrementally. Conditional probabilities are estimated incrementally using some conditioning classes, and these are then fed into an incremental entropy encoder (usually arithmetic). Upon receiving the first few bits of the encoded bit stream, the decoder can start decoding the message using the same conditioning classes and conditional probabilities as the encoder, finally the whole message is reconstructed.

IV. Context Formation Based on Mutual Information Maximization

From the previous sections, we can see that statistical context modeling in the form of probability estimation is the heart of entropy coding. Ultimately, it is the model quality that determines the compression. The central task of the modeler is to estimate the conditional probability, and this is usually accomplished in an adaptive manner.

Because the symbols forming x^{i-1} are all discrete, x^{i-1} can only take a finite number of values. Yet, not every possible value of the past sequence x^{i-1} needs to define a distinct context. The purpose of the function $f(\cdot)$ is to extract the important relevant information that x^{i-1} has about the symbol x_i. The values $\tilde{C}$ that the function $f(\cdot)$ can return are also discrete and finite. So, $f(\cdot)$ is essentially a classification function. Those x^{i-1} that have the same relevant information about x_i should be classified into the same context.

The minimum achievable rate for entropy coding is given by $H(p(X_i|f(x^{i-1})))$. Our objective is to develop a classification scheme that can minimize this conditional entropy. From information theory [27], we know that the conditional entropy is given by

$$H(X|Y) = H(X) - I(X;Y) = H(Y) - I(X;Y). \quad (6)$$

So, minimizing the conditional entropy corresponds to the maximization of the mutual information $I(X;Y)$.

Proposition 1: The mutual information will decrease if contexts are combined unless the conditional probability distributions of the combined contexts are the same.

This proposition can be explained intuitively. Entropy is the uncertainty of a single random variable. Mutual information is the reduction of this uncertainty due to another random variable. This reduction is due to the correlation between two random variables. So, given the knowledge of one random variable, we can predict the other. By merging contexts, this inductive inference capability decreases; thus, the amount of information one random variable provides about the other also decreases.

For a formal proof, one can use the data processing inequality, the convexity of the relative entropy, the concavity of the entropy, or the log sum inequality. A sketch proof is given below.

Proof: Without loss of generality, let the random variable Y be the data to be coded, let the random variable X be the context, let $x_k, k = i \ldots j$ be the contexts that are combined to form a new context x_{cij}, and let p_{x_k} be the probability that context x_k occurs. Then, the mutual information reduction is

obtained by subtracting the MI that is obtained after context combining from the original MI (before any context combining) and is given by

$$\begin{aligned}\mathrm{MI}_{\mathrm{red}}(i \leftrightarrow j) &= p_{x_{cij}} H(p(Y \mid x_{cij})) - \sum_{k=i}^{j} p_{x_k} H(p(Y \mid x_k)) \\ &= \sum_{k=i}^{j} p_{x_k} D(p(Y \mid x_k) \,\|\, p(Y \mid x_{cij})) \end{aligned} \tag{7}$$

where

$$p_{x_{cij}} = \sum_{k=i}^{j} p_{x_k}$$

and

$$p(Y \mid x_{cij}) = \sum_{k=i}^{j} \frac{p_{x_k}}{p_{x_{cij}}} p(Y \mid x_k).$$

From the nonnegativity of $D(\cdot \,\|\, \cdot)$, we get that $\mathrm{MI}_{\mathrm{red}}(i \leftrightarrow j)$ is always bigger or equal to 0. It is 0 when $p(Y \mid x_k) = p(Y \mid x_{cij})$ for $k = i \ldots j$. Therefore, the MI will decrease unless the conditional probability distributions of the combined contexts are the same.

From Proposition 1, it may seem that the mapping function $f(\cdot)$ should be one to one so that we can have as many contexts as the different values for x^{i-1}. This way, we can squeeze as much information out of the available data as possible and we can have the minimum conditional entropy or the maximum mutual information. However, in practice, since the number of symbols to be coded is finite, and the conditional probability is estimated on the fly, we will not generally have enough symbols to get an accurate estimate for each context. This will cause the context dilution problem, which is also called the zero-frequency problem and best match problem in [26]. In addition, too many contexts increase the algorithm's complexity.

To avoid context dilution and lower the complexity, we need to form a finite and sufficiently small number of contexts. Given I, the initial number of contexts, and F, the final desired number of contexts, there are $S(I, F)$ possible classification schemes. $S(I, F)$ is called the Stirling number of the second kind. For $I > F \geq 2$, $S(I, F)$ can be calculated using [28]

$$\begin{aligned} S(I, F) &= S(I-1, F-1) + F \times S(I-1, F) \\ &= \sum_{k=0}^{F-1} (-1)^k \frac{(F-k)^I}{k!(F-k)!}. \end{aligned} \tag{8}$$

The optimal classification scheme is the one that gives the maximum MI. We can search over the $S(I, F)$ possible classification schemes to find out the optimal one [28]. □

Proposition 2: In the optimal context classification, if context x_i is classified into group m and $x_{cn}, 1 \leq n \leq F$ are the grouped context, then $D(p(Y \mid x_i) \,\|\, p(Y \mid x_{cm})) \leq D(p(Y \mid x_i) \,\|\, p(Y \mid x_{cn}))$.

This optimality condition must hold for all contexts. If it is not satisfied, then the classification scheme is not optimal. This optimality condition can be explained intuitively. The relative entropy $D(p \,\|\, q)$ is a measure of the coding efficiency reduction by assuming that the distribution is q when the true distribution is p. In our context classification problem, the true distribution determined by each context is replaced by the distribution determined by the grouped context. In order to maximize the coding efficiency, the optimal classification scheme should classify the context into the group that gives the minimum relative entropy.

TABLE I
EIGHT NEIGHBORS CLASSIFIED INTO THREE GROUPS: H, V, AND D. X DENOTES THE CURRENT SAMPLE

D_0	V_0	D_1
H_1	X	H_0
D_2	V_1	D_3

Proof: It suffices to show that if context x_i is classified into group m, and $D(p(Y \mid x_i) \,\|\, p(Y \mid x_{cn})) \leq D(p(Y \mid x_i) \,\|\, p(Y \mid x_{cm}))$, then a better classification scheme is obtained by putting x_i into group n. In the following, the new grouped context including x_i is denoted by x_{cn+i}

$$p_{x_i} D(p(Y \mid x_i) \,\|\, p(Y \mid x_{cm})) \tag{9}$$

$$\geq p_{x_i} D(p(Y \mid x_i) \,\|\, p(Y \mid x_{cn})) \tag{10}$$

$$\geq p_{x_i} D(p(Y \mid x_i) \,\|\, p(Y \mid x_{cn+i})) + p_{x_{cn}} D(p(Y \mid x_{cn}) \,\|\, p(Y \mid x_{cn+i})). \tag{11}$$

It can be shown that subtracting (11) from (10) equals $(p_{x_i} + p_{x_{cn}}) D(p(Y \mid x_{cn+i}) \,\|\, p(Y \mid x_{cn}))$, which is always greater than or equal to zero.

Instead of using exhaustive search, the context classification can be optimized using a general iterative algorithm based on a modified version of the GLA. The GLA algorithm has been applied in many standard quantizer designs. Here, the distortion metric is changed to one based on the Kullback–Leibler (relative entropy) distance given by (5). Starting with an initial classification of the I contexts into F groups and the corresponding F grouped context, the algorithm can reclassify the I contexts according to the minimum Kullback–Leibler distance. Given the new classification, the F grouped contexts are recalculated. Given the new F grouped contexts, the I contexts can be reclassified. This process is repeated until the MI converges to a stable value. At each step, the algorithm either reduces the MI reduction or keep the MI unchanged. Therefore, after a certain number of iterations, the algorithm is guaranteed to converge. It is shown in [29] that the GLA together with stochastic relaxation techniques can be used to obtain a globally optimal solution. □

Proposition 3: If Y is a binary random variable, the contexts can be ordered according to $p(y \mid X)$, where $y = 1$ or 0, in a one-dimensional space. In the optimal classification scheme, only successively adjacent contexts can be classified into the same group.

Proof: For simplicity, we just need to show that, for three contexts $x_1\, x_2\, x_3$ with $p(y \mid x_1) \leq p(y \mid x_2) \leq p(y \mid x_3)$, merging x_1 and x_2 or x_2 and x_3 is better than merging x_1 and x_3. The relative entropy is a measure of the distance between two distributions. Therefore, both $D(p(Y \mid x_1) \,\|\, p(Y \mid x_2))$ and $D(p(Y \mid x_2) \,\|\, p(Y \mid x_3))$ are less than or equal to $D(p(Y \mid x_1) \,\|\, p(Y \mid x_3))$. Following the same reasoning used in proving Proposition 2, we get that the mutual information

TABLE II
CLASSIFICATION MAP FOR THE ZC CONTEXTS USED IN THE JPEG2000 STANDARD.x DENOTES A "DON'T CARE" ENTRY

LL and LH subbands (vertical high pass)			HL subband (horizontal high pass)			HH subband (diagonal high pass)		Context label
$\sum H$	$\sum V$	$\sum D$	$\sum H$	$\sum V$	$\sum D$	$\sum(H+V)$	$\sum D$	
2	x	x	x	2	x	x	≥ 3	8
1	≥ 1	x	≥ 1	1	x	≥ 1	2	7
1	0	≥ 1	0	1	≥ 1	0	2	6
1	0	0	0	1	0	≥ 2	1	5
0	2	x	2	0	x	1	1	4
0	1	x	1	0	x	0	1	3
0	0	≥ 2	0	0	≥ 2	≥ 2	0	2
0	0	1	0	0	1	1	0	1
0	0	0	0	0	0	0	0	0

reduction caused by the merging of contexts x_1 and x_2, or x_2 and x_3 is less than that caused by merging contexts x_1 and x_3.

Given Proposition 3, the number of possible classification schemes for the binary case is reduced to

$$\frac{(I-1)(I-2)\dots(I-F+1)}{(F-1)!}. \tag{12}$$

In the case of binary arithmetic coding, since the conditional probability is reduced to one dimension, dynamic programming can be used to speed up the optimal context classification search. For this purpose, we first order the I contexts according to $p(y\,|\,X), y = 1$ or 0. If we number the ordered contexts from left to right as 1 to I, and if we denote the MI reduction due to optimal classification of context $i\dots j$ into m groups by $\mathrm{MI}_{\mathrm{red}}(i:j \Rightarrow m)$, then the basis of the dynamic programming algorithm is the following relationship:

$$\mathrm{MI}_{\mathrm{red}}(1:I \Rightarrow F) = \min_{i=1\mathrm{to}I-F+1}(\mathrm{MI}_{\mathrm{red}}(1 \leftrightarrow i) + \mathrm{MI}_{\mathrm{red}}(i+1:I \Rightarrow F-1)) \tag{13}$$

where $\mathrm{MI}_{\mathrm{red}}(i:j \Rightarrow 1) = \mathrm{MI}_{\mathrm{red}}(i \leftrightarrow j)$.

During the initialization, we precompute the $\mathrm{MI}_{\mathrm{red}}(i \leftrightarrow j)$ for $i = 1\dots I, j = i\dots I$ and store the results in a lookup table. The dynamic programming then proceeds to compute the values in the matrix shown in (14), at the bottom of the page.

The elements in the matrix (14) are computed from left to right, top to bottom. The results obtained in row $n-1$ are used in calculating row n. The final value $\mathrm{MI}_{\mathrm{red}}(1:I \Rightarrow F)$ obtained is the solution to our problem. □

V. ANALYSIS OF JPEG2000 CONTEXTS

Context-based binary arithmetic coding is a key component in the JPEG2000 image compression standard. The high-compression efficiency of the JPEG2000 algorithm is in part due to the careful selection of contexts.

In this work, the proposed context formation based on mutual information maximization (Section IV) is used to study the JPEG2000 contexts. Five standard images, Barbara (512×512), bike (2048×2560), cmpnd1 (512×768), hotel (720×576), and us (512×448), are used as training images. Another nine images, aerial2 (720×1024), bike3 (781×919), cafe (2048×2560), cmpnd2 (1024×1400), finger (512×512), goldhill (720×576), Lena (512×512), tools (1524×1200), and woman (2048×2560), are used for testing and comparing the coding performance using the JPEG2000 standard contexts and the contexts obtained by applying the proposed dynamic programming method for context classification to the data samples collected from the training images. The Barbara and Lena images are widely used for image-coding performance comparison. The other 12 images are from the JPEG2000 test image set.

A. ZC Contexts

In zero coding, the nearby 8 neighbors' significance status is used to form the context template C. The 8 neighbors are classified into three group as shown in Table I: *horizontal neighbors* (H), *vertical neighbors* (V), and *diagonal neighbors* (D). Each neighbor can take two states: significant or insignificant. So, the context template C can take 256 possible values. These 256 values are classified into nine coding contexts according to which subband (LL, LH, HL, and HH) the sample comes from. The detailed classification scheme is shown in Table II. This classification scheme can be partly explained intuitively as follows. If we view the high-pass filtering as a local edge detector, we will expect the HL subband (horizontally high pass) to contain mostly vertical edge information. Thus, the vertically adjacent samples will exhibit strong correlations. This explains the emphasis on the vertical neighbors for the HL subband contexts. Similarly, for the LH subband, which contains mostly horizontal

$$\begin{bmatrix} \mathrm{MI}_{\mathrm{red}}(1:I \Rightarrow 2) & \mathrm{MI}_{\mathrm{red}}(2:I \Rightarrow 2) & \cdots & \mathrm{MI}_{\mathrm{red}}(I-2:I \Rightarrow 2) & \mathrm{MI}_{\mathrm{red}}(I-1:I \Rightarrow 2) \\ \mathrm{MI}_{\mathrm{red}}(1:I \Rightarrow 3) & \mathrm{MI}_{\mathrm{red}}(2:I \Rightarrow 3) & \cdots & \mathrm{MI}_{\mathrm{red}}(I-2:I \Rightarrow 3) & \\ \vdots & \vdots & & & \\ \mathrm{MI}_{\mathrm{red}}(1:I \Rightarrow F) & & & & \end{bmatrix}. \tag{14}$$

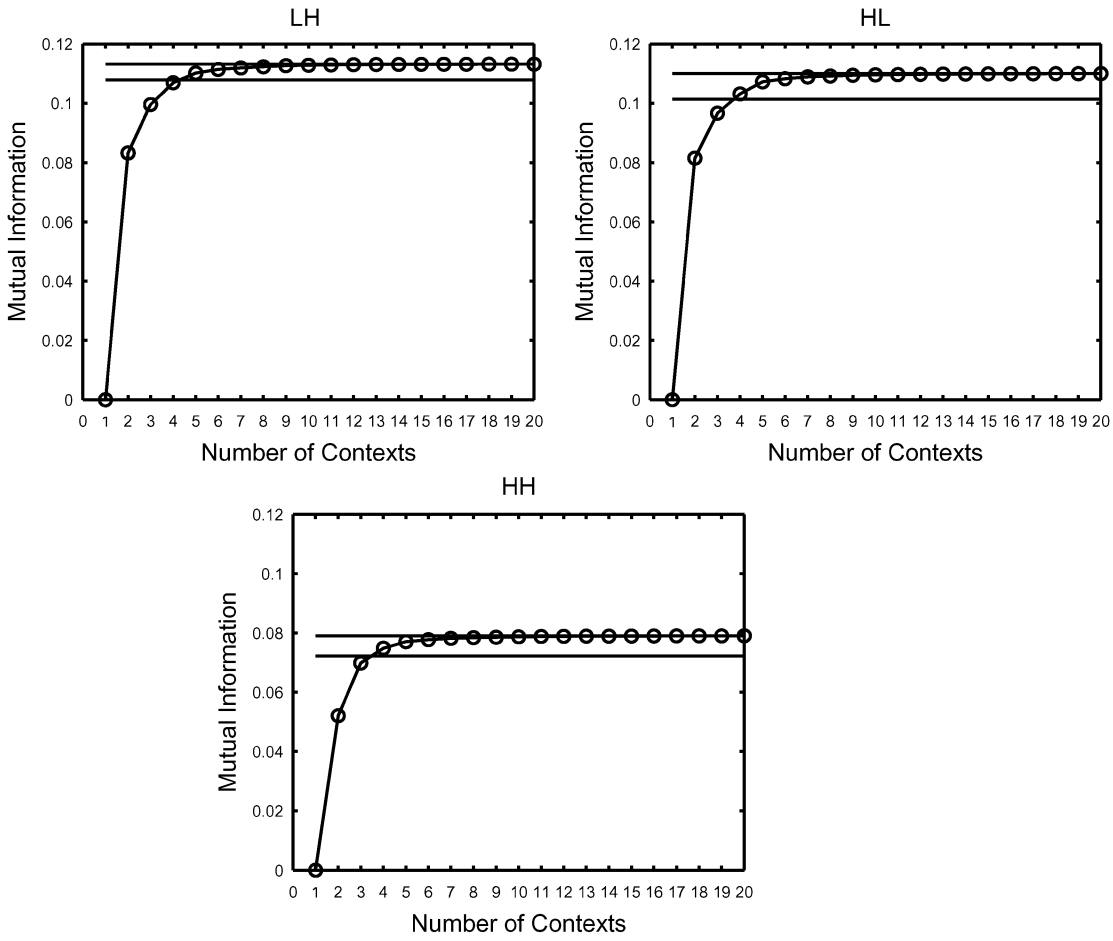

Fig. 2. Context formation based on MI maximization for ZC contexts for subband LH, HL, and HH. Upper horizontal line: maximum MI using 256 contexts. Lower horizontal line: MI obtained using the standard nine ZC contexts of JPEG2000. Curve: MI obtained by optimally classifying the 256 contexts into F groups, where F varies from 1 to 20.

TABLE III
MI CONTRIBUTION FROM THE NINE ZC CONTEXTS IN JPEG2000

	0	1	2	3	4	5	6	7	8	Sum
LH	0.04258	0.00922	0.00004	0.00014	0.00091	0.00393	0.00236	0.03330	0.01538	0.10787
HL	0.04034	0.00883	0.00011	0.00072	0.00075	0.00593	0.00106	0.03116	0.01243	0.10131
HH	0.03285	0.00011	0.00191	0.00462	0.00095	0.00848	0.00008	0.01086	0.01237	0.07222

edge information, more emphasis is put on the horizontal neighbors. However, this does not clarify how the number of contexts is determined and how the contexts are chosen.

As in JPEG2000, three context classification schemes are derived for the ZC contexts using the proposed context classification (discussed in Section IV): one for the LH direction, which corresponds to the data samples collected from all the LH subbands of different levels of the discrete wavelet transform; one for the HL direction; and one for the HH direction. We code the five training images using the JPEG2000 JASPER [30] implementation. At the same time, we count the occurrences of the symbol (0, 1) for each possible C value. These symbols' occurrences are first divided by the corresponding image size and then added up for different DWT levels in the same direction and different images. In this way, we get a 256×2 joint probability distributions for each direction at a specific compression ratio. The different compression ratios that were included in this experiment are 128:1, 64:1, 32:1, 16:1, 8:1 with both $5/3$ and $9/7$ lifting wavelet transform, and lossless compression with $5/3$ lifting. In this way, the statistical changes over different compression ratios and over different wavelet transforms are taken into consideration. The obtained 11 joint probability distributions are equally combined to get a final joint probability distribution which are then used by the dynamic programming algorithm to find the optimal context classification. Fig. 2 shows

TABLE IV
MI OBTAINED BY USING 2, 4, 8 NEIGHBORS IN FORMING THE ZC CONTEXT TEMPLATE WITHOUT ANY CONTEXT COMBINING, AND MI OBTAINED USING THE JPEG2000 NINE ZC CONTEXTS

	LH	HL	HH
2 neighbors(H)	0.077755	0.031250	0.043668
2 neighbors(V)	0.039658	0.072059	0.033822
4 neighbors(H+V)	0.101300	0.091232	0.064069
4 neighbors(D)	0.035937	0.027207	0.034424
8 neighbors	0.113239	0.110055	0.079107
JPEG2000 9 contexts	0.107871	0.101313	0.072223

TABLE V
COMPRESSED FILE SIZE (IN BYTES) IN THE LOSSLESS-CODING MODE FOR THE TEST IMAGES USING THE NEW ZC CONTEXT CLASSIFICATION SCHEME BASED ON MI MAXIMIZATION. JASPER: NINE STANDARD ZERO-CODING JPEG2000 CONTEXTS. 1C–9C: NEW ZC CONTEXTS OBTAINED BASED ON MI MAXIMIZATION

	JASPER	1C	2C	3C	4C	5C	6C	7C	8C	9C
aerial2	394277	396092	394892	394517	394279	**394259**	394279	394323	394470	394502
bike3	417559	420501	418256	417664	417375	**417339**	417355	417466	417573	417703
caf	3510080	3536238	3516115	3512103	3510612	3510368	**3510001**	3510353	3511034	3511584
cmpnd2	386041	393802	387510	386217	**385553**	385808	386005	386318	386517	386767
finger	185757	**185498**	185544	185589	185615	185591	185669	185723	185726	185746
goldhill	238990	239593	238820	238806	**238774**	238833	238865	238975	239011	239119
lena	141509	141594	141409	141376	**141362**	141483	141458	141490	141538	141577
tools	1274042	1281301	1275668	1274768	1274031	**1273876**	1273955	1274366	1274342	1274685
woman	2959173	2960599	2958309	**2957187**	2958374	2958017	2958608	2959316	2959727	2960491
Sum	9507428	9555218	9516523	9508227	9505975	**9505574**	9506195	9508330	9509938	9512174

TABLE VI
AVERAGE PSNR OF NINE TEST IMAGES FOR DIFFERENT COMPRESSION RATIOS AND DIFFERENT WAVELET TRANSFORMS USING THE NEW ZC CONTEXT CLASSIFICATION SCHEME BASED ON MI MAXIMIZATION. JASPER: NINE STANDARD ZC JPEG2000 CONTEXTS. 1C–9C: NEW ZC CONTEXTS OBTAINED BASED ON MI MAXIMIZATION

		JASPER	1C	2C	3C	4C	5C	6C	7C	8C	9C
5/3	8:1	33.12	33.00	33.09	33.10	33.12	33.12	33.12	33.11	33.11	33.11
5/3	16:1	28.84	28.74	28.84	28.85	28.85	28.83	28.83	28.83	28.83	28.82
5/3	32:1	25.52	25.41	25.51	25.50	25.52	25.52	25.52	25.51	25.51	25.52
5/3	64:1	23.05	22.99	23.04	23.03	23.04	23.05	23.05	23.05	23.04	23.04
5/3	128:1	21.26	21.21	21.26	21.26	21.26	21.26	21.25	21.27	21.26	21.25
9/7	8:1	33.58	33.43	33.53	33.55	33.56	33.56	33.57	33.57	33.56	33.56
9/7	16:1	29.32	29.18	29.29	29.30	29.31	29.31	29.32	29.31	29.32	29.31
9/7	32:1	25.96	25.83	25.95	25.94	25.96	25.95	25.95	25.95	25.95	25.95
9/7	64:1	23.45	23.39	23.46	23.43	23.45	23.44	23.45	23.44	23.43	23.44
9/7	128:1	21.58	21.56	21.57	21.56	21.57	21.58	21.57	21.57	21.57	21.57

the MI obtained for different numbers of contexts. In Fig. 2, the horizontal axis shows the number of contexts. The vertical axis shows the mutual information. The upper horizontal line is the MI if all the initial 256 contexts are used; this corresponds to no context combining. The lower horizontal line corresponds to the MI obtained if the JPEG2000's context classification scheme is used. The solid curve is the MI obtained by optimally classifying the 256 contexts into F contexts using the proposed MI-based context classification method, where F varies from 1 to 20. We can get this curve by subtracting the first column of matrix (14) from the MI obtained without context combining. Because the MI increases at a very slow speed after $F \geq 20$, we only show the MI obtained for $F \leq 20$ in Fig. 2. It can be seen that by using only three or four contexts, we can obtain the same MI as the JPEG2000 which uses nine contexts. Also, it can be seen that the MI with nine or ten contexts is almost equal to the MI with 256 contexts.

Table III shows the MI contribution from the nine contexts when the JPEG2000 context classification is applied. The context labels used in Table III are from Table II. We can see that the largest three MI contributions are from the zero, eight, and seven contexts as expected. If the 8 neighbors are all insignificant, it is highly probable that the sample will also be insignificant. This is the reason why the significance identification in the bitplane coding is divided into the forward significance propagation pass and cleanup pass in the JPEG2000 standard. The large MI contributions from contexts eight and seven confirm the emphasis on the horizontal neighbor in forming the contexts for the LH subband, and the vertical neighbor in forming the contexts for the HL subband. A natural question is whether it is sufficient to use just the horizontal and vertical neighbors in forming the LH and HL subband contexts and use just the diagonal neighbors in forming the HH subband contexts. Table IV shows the MI obtained by using 2, 4, 8 neighbors in forming the context template without any context combining. For N neighbors, we then get 2^N contexts. For comparison purposes, the MI obtained using the nine JPEG2000 ZC contexts are also listed. From Table IV, one can see that for the LH direction, the MI obtained by just using horizontal neighbors is much larger than that obtained by just using the vertical neighbor. However, the vertical neighbor do provide some additional information about the symbol to be coded, which is evidenced from the fact that the 4 neighbors (H+V) MI is almost 0.03 larger than just using the horizontal neighbors. At the same time, the four horizontal and vertical neighbors give much larger MI than the four diagonal neighbors. For the HL direction, a similar observation can be made. For the HH direction, one unexpected observation is that

TABLE VII
CLASSIFICATION MAP FOR THE SC CONTEXTS USED IN THE JPEG2000 STANDARD

$H_0 + H_1$	$V_0 + V_1$	Context label	Sign flipping
> 0	> 0	4	No
> 0	$= 0$	3	No
> 0	< 0	2	No
$= 0$	> 0	1	No
$= 0$	$= 0$	0	No
$= 0$	< 0	1	Yes
< 0	> 0	2	Yes
< 0	$= 0$	3	Yes
< 0	< 0	4	Yes

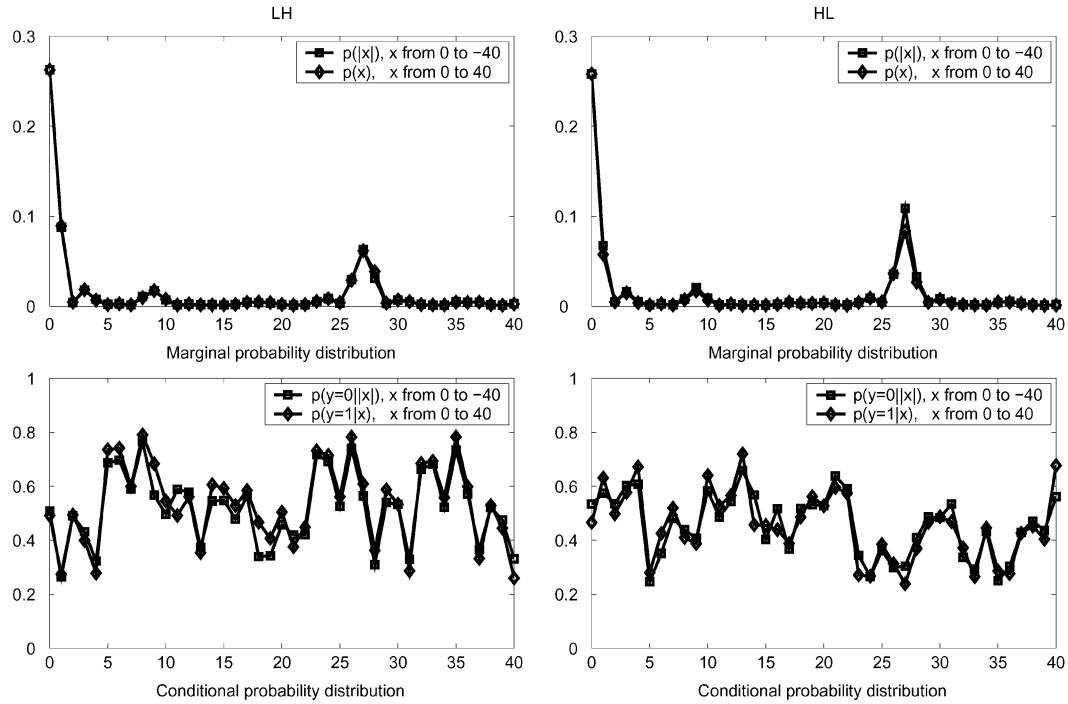

Fig. 3. Marginal probability distribution $p(X)$ and conditional probability distribution $p(y|X)$ for the (left) LH and (right) HL subbands, obtained from training images for sign coding.

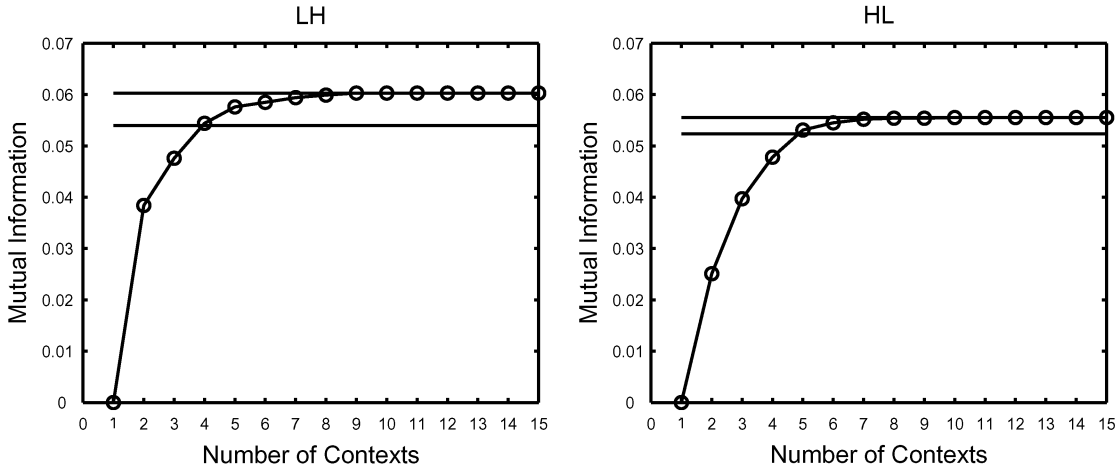

Fig. 4. Context formation based on MI maximization for sign coding. Upper horizontal line: maximum MI using 81 contexts. Lower horizontal line: MI obtained using the standard five SC contexts of JPEG2000. Curve: MI obtained by optimally classifying the 81 contexts into F groups, where F varies from 1 to 15.

the MI obtained by using the four horizontal and vertical neighbors is larger than that using the four diagonal neighbors. This indicates that more emphasis should be put in the horizontal and vertical neighbors for forming the ZC context for the HH subband. Compared with the MI obtained with 8 neighbors, the four horizontal and vertical neighbors can provide a similar MI. However, in order to get an MI comparable to the one obtained with the JPEG2000 nine ZC contexts, we do need 8 neighbors in forming the context template.

To illustrate the coding performance using the new computed contexts, the JPEG2000 JASPER implementation [30] is modified. Instead of using the nine ZC contexts specified in the standard, they are replaced by the F contexts that are computed using the proposed MI-based method, where F is varied from 1 to 9. Table V shows the compressed file size in the lossless-coding mode for the test image set. The smallest compressed file size from 1C to 9C is highlighted in bold font. These results generally fit our expectation. With the increase of context number F, the compression efficiency first increases. As the number of contexts increase further, the compression efficiency decreases which is due to context dilution. The best result is obtained by using four or five contexts. One unexpected observation is that for the fingerprint image, the compression result without using any context (1C case) is the best.

For lossy coding, Table VI shows the average PSNR over the nine test images at different compression ratios for both the $9/7$ and $5/3$ lifting wavelet transforms. From Table VI, we can see that the PSNR results are very close for all values of F, where F is the number of contexts used. The worst results shown in column 1C, which are obtained without any context modeling,

TABLE VIII
MI CONTRIBUTION FROM THE FIVE SC CONTEXTS IN JPEG2000

	0	1	2	3	4	Sum
LH	0.005851	0.015700	0.011560	0.020348	0.000545	0.054004
HL	0.000005	0.020746	0.019305	0.011962	0.000331	0.052349
HH	0.000178	0.000448	0.000141	0.000657	0.000032	0.001456

TABLE IX
CLASSIFICATION MAP FOR THE MAGNITUDE-REFINEMENT CONTEXTS USED IN THE JPEG2000 STANDARD. X DENOTES A "DON'T CARE" ENTRY

$\sum H + \sum V + \sum D$	First refinement for this coefficient	Context label
x	false	2
≥ 0	true	1
0	true	0

are usually within 0.2 dB of the best results. By using two contexts, we can get the compression results that is within 0.02 dB of the JPEG2000 nine contexts. Therefore, for lossy coding, two contexts are enough to get a comparable compression efficiency as the nine ZC JPEG2000 contexts.

B. SC Contexts

In JPEG2000, after a sample is identified as significant, its sign is coded immediately. The horizontal and vertical neighbors' significance status and sign are used in forming the context template. Each neighbor can take three values: significant positive $(+1)$, significant negative (-1), and insignificant (0). Therefore, the context template can take 81 different values. In JPEG2000, these 81 values are then classified into five coding contexts according to the sign of $H_0 + H_1$ and $V_0 + V_1$, where H_0, H_1, V_0, V_1 are the horizontal and vertical neighbors defined in Table I. The detailed classification scheme is shown in Table VII. If the sign flipping is yes, then the sign is flipped prior to encoding.

Fig. 3 shows the obtained conditional probability distribution $p(y\,|\,X)$ and marginal probability distribution $p(X)$ from the five training images for the LH and HL directions. The horizontal axis is x, the value that the context template can take. x is calculated as $x = V_0 \times 27 + V_1 \times 9 + H_0 \times 3 + H_1$, therefore, $-40 \leq x \leq +40$. In Fig. 3, the probability distribution for $x \leq 0$ is flipped. One can see that the marginal probability distribution $p(|x|), -40 \leq x \leq 0$ and $p(x), 0 \leq x \leq 40$ are almost the same, and that the conditional probability distribution $p(y = 1\,|\,x), 0 \leq x \leq 40$ and $p(y = 0||x|), -40 \leq x \leq 0$ are very similar. These properties confirm sign flipping. In the $p(X)$ plot, there are peaks at $|x| = 0, 27, 9, 3, 1$. The peaks at 27 and 1 are higher than the peaks at 9 and 3. This is due to the code block scan pattern used in JPEG2000.

Fig. 4 shows the MI obtained by applying the proposed context classification method to the probability distribution of Fig. 3. The upper horizontal line is the MI obtained without any context combining. The lower horizontal line is the MI obtained with the JPEG2000 five SC contexts. The curve is the MI obtained by optimally classifying the 81 contexts into F groups, where F is varied from 1 to 15. From Fig. 4, one can conclude that we do need five contexts for efficient sign coding. Table VIII shows the MI contribution from the five JPEG2000 contexts. The context labels used in Table VIII are from Table VII. We can see that the MI contribution from the number 0 and four contexts are small. The small MI contribution is out of expectation since, originally, one would think that, if $H_0 + H_1$ and $V_0 + V_1$ are of the same sign, it would be highly probable that the considered symbol will follow that sign. From Table VIII, it can be seen that this assumption does not hold. In addition, the MIs for the HH subband are small compared to the MIs of the LH and HL subbands. Therefore, in Figs. 3 and 4, we just show LH and HL subband information.

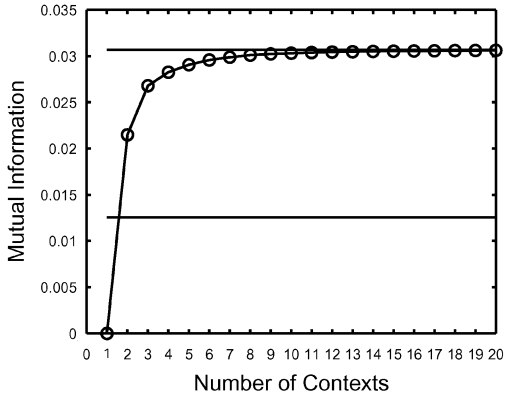

Fig. 5. Context formation based on MI maximization for MR coding. Upper horizontal line: maximum MI using 512 contexts. Lower horizontal line: MI obtained using the standard three MR contexts of JPEG2000. Curve: MI obtained by optimally classify the 512 contexts into F groups, where F is from 1 to 20.

TABLE X
MI CONTRIBUTION FROM THE THREE MR CONTEXTS IN JPEG2000

0	1	2	Sum
0.002773	0.003075	0.006702	0.012551

C. MR-Coding Contexts

In JPEG2000, the contexts that are used to code the MR information, are determined by the summation of the significance status of the 8 neighbors and whether the bits being coded are the first refinement bits. Therefore, the context template can take 512 different values. These 512 values are classified, in JPEG2000, into three coding contexts according to Table IX.

One MR context classification scheme is derived by applying the proposed context classification to the training data collected from the training images at different compression ratios. The

TABLE XI
COMPRESSED FILE SIZE (IN BYTES) IN THE LOSSLESS-CODING MODE FOR THE TEST IMAGES USING THE NEW MAGNITUDE-REFINEMENT CONTEXT CLASSIFICATION SCHEME BASED ON MI MAXIMIZATION. JASPER: THREE STANDARD JPEG2000 MAGNITUDE REFINEMENT CONTEXTS. 1C–5C: NEW MAGNITUDE REFINEMENT CONTEXTS OBTAINED BASED ON MI MAXIMIZATION

	JASPER	1C	2C	3C	4C	5C
aerial2	394277	394119	**394053**	394171	394206	394340
bike3	417559	417434	**417302**	417396	417497	417559
caf	3510080	3508788	**3508224**	3508533	3509428	3510447
cmpnd2	386041	387073	386175	385734	**385651**	385664
finger	185757	**185697**	185769	185856	185857	185947
goldhill	238990	238902	**238957**	239025	239070	239108
lena	141509	141429	**141424**	141494	141473	141482
tools	1274042	1273735	**1273510**	1273671	1274011	1274315
woman	2959173	**2958236**	2958398	2958704	2959716	2960567
Sum	9507428	9505413	**9503812**	9504584	9506909	9509429

MI obtained are shown in Fig. 5. The upper horizontal line is the MI with the original 512 contexts. The lower horizontal line is the MI with the JPEG2000 three MR contexts. The curve is the MI obtained by optimally classifying the 512 contexts into F groups, where F varies from 1 to 20. From Fig. 5, one can see that by using two contexts, we can obtain much better MI than JPEG2000. The compression results also confirm this finding. Table X shows the MI contribution from the three MR contexts in JPEG2000. Table XI shows the compressed file size in the lossless-coding mode for the test image set. The smallest file size in each row is highlighted in bold font. One can see that, for most images, the contexts derived from the proposed method give better compression results than JPEG2000. For six out of nine images, the result in the 2C column is better than JPEG2000. These results tell us that the JPEG2000 MR context is not very effective. Similar trends are observed for lossy coding.

VI. CONCLUSION

From the results presented in Section V, it can be concluded that the ZC and SC contexts in JPEG2000 capture the correlation among the wavelet coefficients very well. In fact, from Figs. 2 and 4, the MI of the JPEG2000 (lower horizontal line) is not far from the maximum possible MI (upper horizontal line). However, the proposed method allows us to achieve similar coding performance as JPEG2000 using a smaller number of contexts and may result in a lower-coding complexity.

REFERENCES

[1] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal 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, Apr. 1996.

[3] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. Image Process.*, vol. 6, no. 5, pp. 677–693, May 1997.

[4] S. D. Servetto, K. Ramchandran, and M. T. Orchard, "Image coding based on a morphological representation of wavelet data," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1161–1174, Sep. 1999.

[5] B. B. Chai, J. Vass, and X. Zhuang, "Significance-linked connected component analysis for wavelet image coding," *IEEE Trans. Image Process.*, vol. 8, no. 6, pp. 774–784, Jun. 1999.

[6] R. L. Joshi, H. Jafarkhani, J. H. Kasner, T. R. Fischer, N. Farvardin, M. W. Marcellin, and R. H. Bamberger, "Comparison of different methods of classification in subband coding of images," *IEEE Trans. Image Process.*, vol. 6, no. 11, pp. 1473–1486, Nov. 1997.

[7] X. Wu, "Context quantization with fisher discriminant for adaptive embedded wavelet image coding," in *Proc. Data Compression Conf.*, 1999, pp. 102–111.

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

[9] 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.

[10] W. Berghorn, T. Boshamp, M. Lang, and H. O. Peitgen, "Context conditioning and run-length coding for hybrid, embedded progressive image coding," *IEEE Trans. Image Process.*, vol. 10, no. 12, pp. 1791–1800, Dec. 2001.

[11] B. J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3D set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 8, pp. 1374–1387, Dec. 2000.

[12] D. Taubman and Z. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Process.*, vol. 3, no. 9, pp. 572–588, Sep. 1994.

[13] J. Vass, B. B. Chai, K. Palaniappan, and X. Zhuang, "Significant-linked connected component analysis for very low bit rate wavelet video coder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 630–647, Jun. 1999.

[14] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Process.*, vol. 1, no. 4, pp. 205–230, Apr. 1992.

[15] P. Sriram and M. W. Marcellin, "Image coding using wavelet transform and entropy-constrained trellis-coded quantization," *IEEE Trans. Image Process.*, vol. 4, no. 6, pp. 725–733, Jun. 1995.

[16] R. L. Joshi, V. J. Crump, and T. R. Fischer, "Image subband coding using arithmetic coded trellis coded quantization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 8, pp. 515–523, Dec. 1995.

[17] E. P. Simoncelli, "Statistical models for images: Compression, restoration, and synthesis," in *Proc. 31st Asilomar Conf. Signals, Systems, and Computers*, Nov. 1997, pp. 673–678.

[18] 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 *Data Compression Conf.*, 1997, pp. 221–230.

[19] 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.

[20] J. K. Romberg, H. Choi, and R. G. Baraniuk, "Bayesian tree-structured image modeling using wavelet-domain hidden Markov models," *IEEE Trans. Image Process.*, vol. 10, no. 7, pp. 1056–1068, Jul. 2001.

[21] H. Choi and R. G. Baraniuk, "Multiscale image segmentation using wavelet-domain Hidden Markov models," *IEEE Trans. Image Process.*, vol. 10, no. 9, pp. 1309–1321, Sep. 2001.

[22] X. Wu, "High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression," in *Proc. 31st Asilomar Conference on Signals, Systems, and Computers*, Nov. 1997, pp. 1378–1382.

[23] C. Chrysafis and A. Ortega, "Efficient context-based entropy coding for lossy wavelet image compression," in *Data Compression Conf.*, 1997, pp. 241–250.
[24] W. B. Pennebaker, J. L. Mitchell, G. Langdon, and R. B. Arps, "An overview of the basic principles of the Q-coder adaptive binary arithmetic coder," *IBM J. Res. Develop.*, vol. 32, pp. 717–726, Nov. 1988.
[25] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520–541, Jun. 1987.
[26] T. C. Bell, J. G. Cleary, and I. H. Witten, *Text Compression*. Upper Saddle River, NJ: Prentice-Hall, 1990.
[27] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
[28] D. L. Kreher and D. R. Stinson, *Combinatorial Algorithms*. Boca Raton, FL: CRC, 1999.
[29] K. Zeger, J. Waisey, and A. Gersho, "Globally optimal vector quantization design by stochastic relaxation," *IEEE Trans. Signal Process.*, vol. 40, no. 2, pp. 310–322, Feb. 1992.
[30] (2000). [Online]. Available: http://www.ece.ubc.ca/mdadams/jasper/
[31] A. K. Jain, *Fundamentals of Digital Image Processing*. Upper Saddle River, NJ: Prentice-Hall, 1989.
[32] A. Deever and S. S. Hemami, "What's your sign?: Efficient sign coding for embedded wavelet image coding," in *Data Compression Conf.*, 2000, pp. 273–282.
[33] J. Liu and P. Moulin, "Analysis of interscale and intrascale dependencies between image wavelet coefficients," presented at the *Int. Conf. Image Processing*, 2000.
[34] [Online]. Available: http://www.jpeg.org/CDs15444.htm
[35] P. G. Howard and J. S. Vitter, "Arithmetic coding for data compression," *Proc. IEEE*, no. 6, pp. 857–865, Jun. 1994.

Zhen Liu received the B.S. and M.S. degrees in telecommunications from the Nanjing University of Posts and Telecommunication, Nanjing, China, and the Ph.D. degree in electrical engineering from Arizona State University (ASU), Tempe, 1995, 1998, and 2003, respectively.

He is currently a Senior Engineer with the Digital Signal Processing Department, Qualcomm, Inc., San Diego, CA. His research interests are in the areas of image and video compression, transmission, and communication. From 1999 to 2003, he was a Graduate Research Assistant in the Image, Video, and Usability Laboratory, ASU, and a Graduate Teaching Assistant with the Electrical Engineering Department, ASU. He was an intern at HP Laboratories and the Ricoh California Research Center where he worked on compound document compression during the summers of 2000 and 2003, respectively.

Lina J. Karam (S'91–M'95–SM'03) received the B.E. degree in electrical engineering from the American University of Beirut, Beirut, Lebanon, and the M.S. and Ph.D. degrees in electrical engineering from the Georgia Institute of Technology, Atlanta, in 1989, 1992, and 1995, respectively.

She is currently an Associate Professor with the Department of Electrical Engineering, Arizona State University, Tempe. From 1991 to 1995, she was a Graduate Research Assistant with the Graphics, Visualization, and Usability Center and then with the Department of Electrical Engineering, Georgia Institute of Technology. In 1992, she was with Schlumberger Well Services working on problems related to data modeling and visualization. In 1994, she was with the Signal Processing Department, AT&T Bell Laboratories, working on problems of video coding. Her research interests are in the areas of image and video processing, image and video coding, error-resilient source coding, and digital filtering.

Dr. Karam is the recipient of an NSF CAREER Award. She served as Chair of the IEEE Communications and Signal Processing Chapters in Phoenix, AZ, in 1997 and 1998. She also served as an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING from 1999 to 2003. She is an elected member of the IEEE Circuits and Systems (CAS) Society's DSP Technical Committee and of the IEEE Signal Processing (SP) Society's IMDSP Technical Committee and a voting member of the IEEE SP Society's Conference Board. She is also a member of the IEEE Signal Processing and Circuits and Systems Societies.