# Context-Based Entropy Coding of Block Transform Coefficients for Image Compression

Chengjie Tu, *Student Member, IEEE,* and Trac D. Tran, *Member, IEEE*

***Abstract*—It has been well established that state-of-the-art wavelet image coders outperform block transform image coders in the rate-distortion (R-D) sense by a wide margin. Wavelet-based JPEG2000 is emerging as the new high-performance international standard for still image compression. An often asked question is: how much of the coding improvement is due to the transform and how much is due to the encoding strategy? Current block transform coders such as JPEG suffer from poor context modeling and fail to take full advantage of correlation in both space and frequency sense. This paper presents a simple, fast, and efficient adaptive block transform image coding algorithm based on a combination of prefiltering, postfiltering, and high-order space–frequency context modeling of block transform coefficients. Despite the simplicity constraints, coding results show that the proposed coder achieves competitive R-D performance compared to the best wavelet coders in the literature.**

***Index Terms*—Adaptive entropy coding, block transform, context modeling, DCT, image coding, JPEG, postfiltering, prefiltering.**

## I. Introduction

An image coding algorithm generally involves a transformation to compact most of the energy of the input image into a few transform coefficients which are then quantized and entropy encoded. The two popular transformation approaches for image compression are the block transform and the wavelet transform. The block-based approach partitions the input image into small nonoverlapped blocks; each of them is then mapped into a block of coefficients via a particular block transform usually constructed from local cosine/sine bases. Most popular amongst block transforms for visual data is the type-II discrete cosine transform (DCT) [1] and the block size is commonly set to 8 × 8. Block DCT coding is the basis of many international multimedia compression standards from JPEG for still images to the MPEG family for video sequences. Low-complexity and good energy compaction within a data block are two main reasons why the DCT has been popular. Unfortunately, at low bit rates, block-based systems suffer from the notorious blocking artifacts, i.e., discontinuities at the block boundaries resulting from reconstruction mismatches.

Unlike the block transform approach, the wavelet transform approach treats the input signal globally and avoids blocking artifacts by employing overlapped basis functions. (To process high-resolution images, practical implementations partition the input into large data blocks, called tiles, and transform them independently.) The wavelet transform can be interpreted as an iteration of a two-channel filter bank with a certain degree of regularity on its lowpass output. Part of the beauty and power of wavelets is their elegance: complicated tilings of the time–frequency plane can be easily achieved by merely iterating a simple two-channel decomposition. Furthermore, a coarse approximation together with detailed wavelet components at different resolutions allows fast execution of many DSP applications such as image browsing, database retrieval, scalable multimedia delivery, etc.

Since the introduction of the embedded zerotree wavelet (EZW) compression algorithm by Shapiro in 1993 [2], wavelet coding technology has advanced significantly. State-of-the-art wavelet coding algorithms, such as Said and Pearlman's set partitioning in hierarchical trees (SPIHT) [3], Chrysafis and Ortega's context-based entropy coding (C/B) [4], Wu's embedded conditional entropy coding of wavelet coefficients (ECECOW) [5] and ECECOW with context quantization guided by Fisher discriminant (FD) [6], Hong and Ladner's group testing for wavelets (GTW) [7], and especially Taubman's embedded block coding with optimized truncation (EBCOT) [8]—the framework for the current state of the JPEG2000 image compression standard [9], give the best R-D performance in the literature. In the meantime, the progress for block transform image coding is limited and the R-D performance gap between the best wavelet coding algorithm and the best block transform coding algorithm is quite large.

The success of wavelet coding technology is mainly a result of advanced context modeling and adaptive entropy coding of wavelet coefficients. Good context-based entropy coding well exploits the nature of wavelet coefficients such as multiresolution structure, parent–children relationship, zero clustering, and sign correlation. On the contrary, context based entropy coding of block transform coefficients has not received much attention. The standard method for coding quantized block transform coefficients is still JPEG's zigzag scanning and runlength coding technology, whose efficiency at least suffers from: 1) 2-D data is encoded in a 1-D manner; correlation between coefficients in the same block has not been fully exploited; 2) data blocks are coded independently (except for DC prediction) and correlation between blocks has been mostly ignored; and 3) too many run-level combinations result in suboptimal entropy coding.

Recently, there have been several block-transform-based image coders which use zerotree wavelet coding algorithms such as EZW and SPIHT to encode block transform coefficients

Manuscript received December 17, 2001; revised July 11, 2002. This work was supported by the NSF under CAREER Grant CCR-0093262. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nasir Memon.

The authors are with the Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: cjtu@jhu.edu; trac@jhu.edu).

Digital Object Identifier 10.1109/TIP.2002.804279

1057-7149/02$17.00 © 2002 IEEE

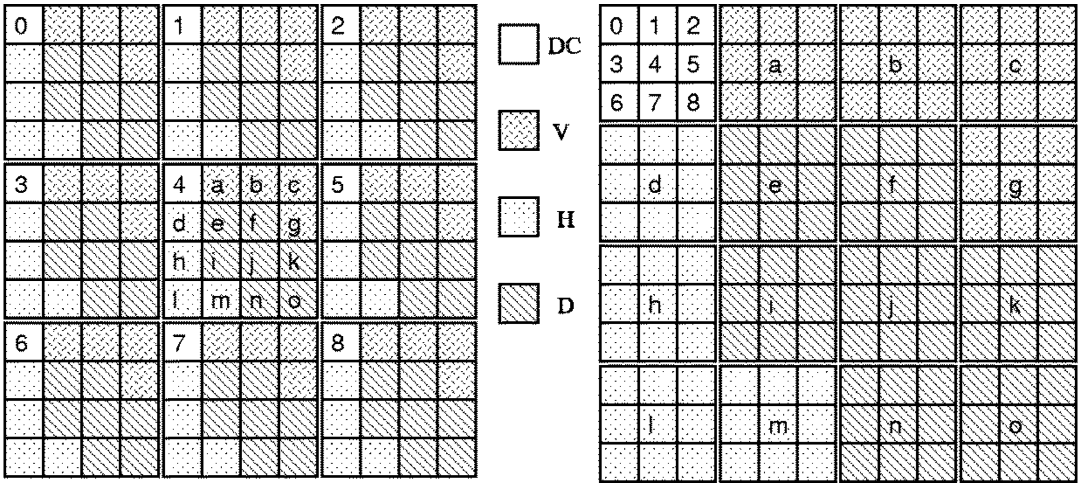


Fig. 1. Space–frequency relationship between coefficients in (left) blocks and (right) subbands.

by reordering them into familiar wavelet structures [10]–[12]. These block transform coders significantly outperform the traditional JPEG coder in the R-D sense. When correlation between blocks is taken into account by employing lapped transforms, competitive or even better compression compared to wavelet coders can be achieved on a wide range of images at a wide range of bit rates. Notice that the zerotree structure is actually a high-order context model. The significant improvement over JPEG is mainly a result of better context based entropy coding of coefficients. Although these coders exhibit the power of context-based entropy coding of block transform coefficients, they have not been optimized for block transforms. Wavelet-specific features are imposed and special characteristics of block transforms are ignored.

In this paper, we present a novel adaptive block transform image coding algorithm based on context-based entropy coding of block transform coefficients (CEB). Benefiting from high-order context modeling and adaptive entropy coding optimized for block transforms, CEB demonstrates superior R-D performance. CEB comes with two flavors: local CEB (L-CEB) and embedded CEB (E-CEB). L-CEB transforms, quantizes, and compresses an image block by block sequentially in place without buffering others. It is geared toward resource-constrained applications with high speed and very low memory requirement such as those in mobile handheld devices. E-CEB compresses an image bitplane by bitplane just like other progressive coders. All transform coefficients are buffered; however, each bitplane is still coded block by block. E-CEB offers rate and distortion scalability. E-CEB also allows better context modeling and thus achieves better compression, especially at high bit rates.

The paper is organized as follows. The next section describes the space–frequency subband representation of block transform coefficients and how simple prefiltering and postfiltering can take into account correlation between blocks to maximize coding efficiency. In Section III, we show that a coefficient is highly correlated not only with its block neighbors, but also with its subband neighbors. Based on this observation, Section IV discusses effective context based entropy coding of block transform coefficients. The generic CEB algorithm and specific examples of L-CEB and E-CEB implementations follow in Sections V and VI, respectively. In Section VII, we present extensive coding results and compare them with those of other state-of-the-art algorithms.

## II. Representation of Block Transform Coefficients

In this paper, a block transform refers to a separable linear mapping of an $M \times M$ block of image pixels into an $M \times M$ block of transform coefficients. We shall use exclusively the 8 × 8 type-II DCT [1] ($M = 8$) which has been proven to be a good tradeoff between complexity and coding performance. An $M$-point block transform can also be interpreted as an $M$-band filter bank whose filters are simply the transform's basis functions.

### A. Space–Frequency Relationship

The $M$-point DCT maps a $KM \times LM$ input image into a $K \times L$ grid of $M \times M$ coefficient blocks. Under the aforementioned filter bank perspective, the coefficients can be reordered into an $M \times M$ grid of $K \times L$ subbands. The coefficient at position $(i, j)$ $(0 \leq i, j < M)$ in block $(x, y)$ $(0 \leq x < K,$ $0 \leq y < L)$ is located at position $(x, y)$ in subband $(i, j)$. In other words, subband $(i, j)$ collects all coefficients at the $(i, j)$ position from every block. These coefficients represent the same frequency component of the entire image. On the other hand, block $(x, y)$, which gathers all coefficients at the same spatial location $(x, y)$ from every subband, represents different frequency components of a local spatial region. Dominating features of an image are often more obvious in subbands than in blocks. Fig. 1 demonstrates the two different representations of coefficients: the block representation and the subband representation.

Subbands can be divided into four categories: a DC subband, V subbands containing vertical features, H subbands containing horizontal features, and D subbands containing diagonal features. These correspond approximately to the LL subband, LH subbands, HL subbands, and HH subbands in the familiar hierarchical wavelet structure. As depicted in Fig. 1, subband (0, 0) at the upper left corner is the DC subband; subbands with $i > 2j$

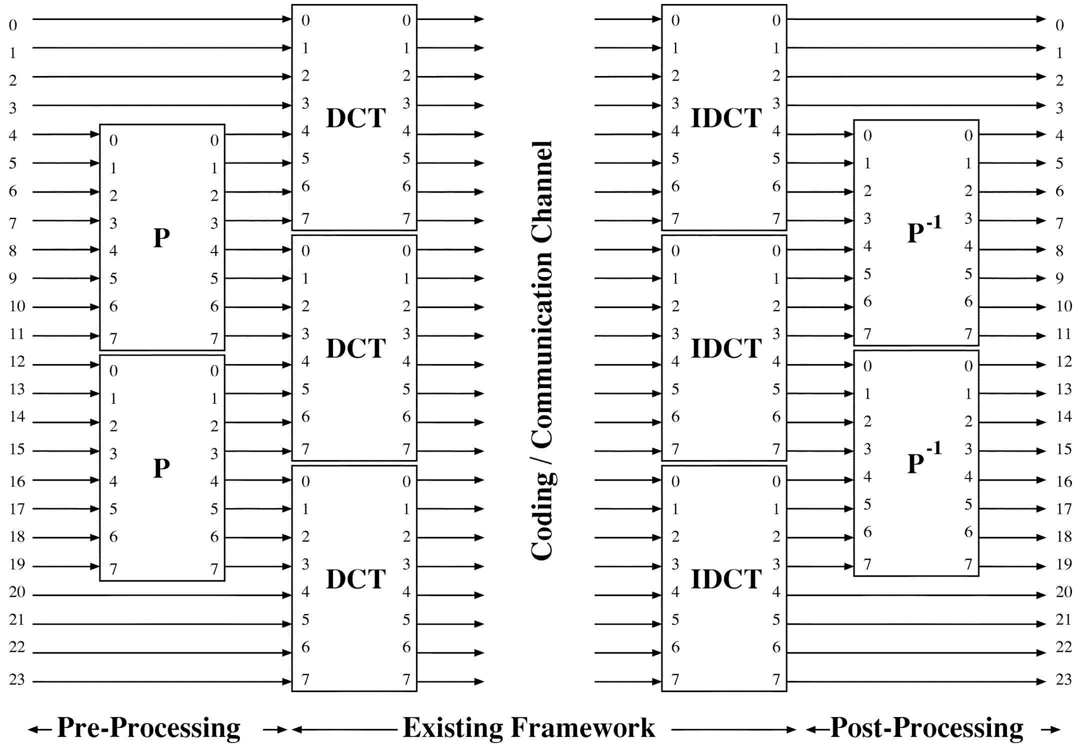


Fig. 2. Prefiltering and postfiltering at block boundaries.

are classified as V subbands; subbands with $j > 2i$ are classified as H subbands; and the remaining are D subbands.

Let $C_{x,y}^{i,j}$ denote the transform coefficient at position $(i, j)$ in block $(x, y)$, or in other words, the coefficient at position $(x, y)$ in subband $(i, j)$. Coefficient $C_{x,y}^{i,j}$ has two types of neighbors: block neighbors in space and subband neighbors in frequency. For example, in Fig. 1, the coefficient labeled "4" has block neighbors "$a$"–"$o$" and subband neighbors "0," "1," "2," "3," "5," "6," "7," and "8."

### B. Prefiltering and Postfiltering

To avoid blocking artifacts and to improve coding efficiency, but still retaining all attractive features of block-based techniques, we rely on prefiltering and postfiltering along the block boundaries of the traditional block-based coding scheme. Our prefiltering and postfiltering framework is illustrated in Fig. 2. The prefilter $\mathbf{P}$ processes the block boundaries, taking away correlation between blocks. The preprocessed samples are then fed to the DCT to be transformed and encoded as usual. At the decoder side, $\mathbf{P}^{-1}$ serves as the postfilter, reconstructing the data in an overlapping manner, eliminating blocking artifacts. Both prefiltering and postfiltering can be performed block-wise locally just like the DCT. For 2-D image data, prefiltering is performed in a separable fashion, followed by the common separable DCT decomposition. Postfiltering is performed separably as well.

The block prefilter consists of two stages of butterflies and a matrix $\mathbf{V}$ between them

$$\mathbf{P} = \frac{1}{2}\begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix} \tag{1}$$

where $\mathbf{I}$, $\mathbf{J}$, and $\mathbf{0}$ are the $M/2 \times M/2$ identity matrix, the reversal matrix, and the null matrix, respectively. The corresponding postfilter is

$$\mathbf{P}^{-1} = \frac{1}{2}\begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^{-1} \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix}. \tag{2}$$

The matrix $\mathbf{V}$ controls prefiltering and postfiltering. The combination of prefiltering/postfiltering and the DCT can be shown to generate a family of invertible linear phase lapped transforms [13]. It should be noted that if we set $\mathbf{V} = \mathbf{I}$ then $\mathbf{P} = \mathbf{P}^{-1} = \mathbf{I}$, i.e., prefiltering/postfiltering is turned off, and we are back to a traditional block-DCT-based coding system. At image boundaries, extending the image symmetrically followed by prefiltering or postfiltering is equivalent to bypassing prefiltering or postfiltering altogether [13].

The detailed design of fast and efficient prefiltering/postfilters is described in another publication [14]. The prefilter $\mathbf{P}$ used in this paper depicted on the left of Fig. 3 is taken directly from [14]. The corresponding postfilter $\mathbf{P}^{-1}$ is shown on the right. The filters are designed to be of low computational complexity without losing much (if not any) coding efficiency. In particular, we choose $\mathbf{V}$ such that the postfilter is multiplierless.

Fig. 4 demonstrates the effect of prefiltering on an input image: the preprocessed image becomes very blocky; however, each block fed into the DCT is smooth. The prefilter extracts the correlation between adjacent blocks, hence increasing the DCT's effectiveness in energy compaction. The postfilter is the exact inverse of the prefilter. It operates on the signal between blocks as a smooth interpolator, eliminating blocking artifacts.

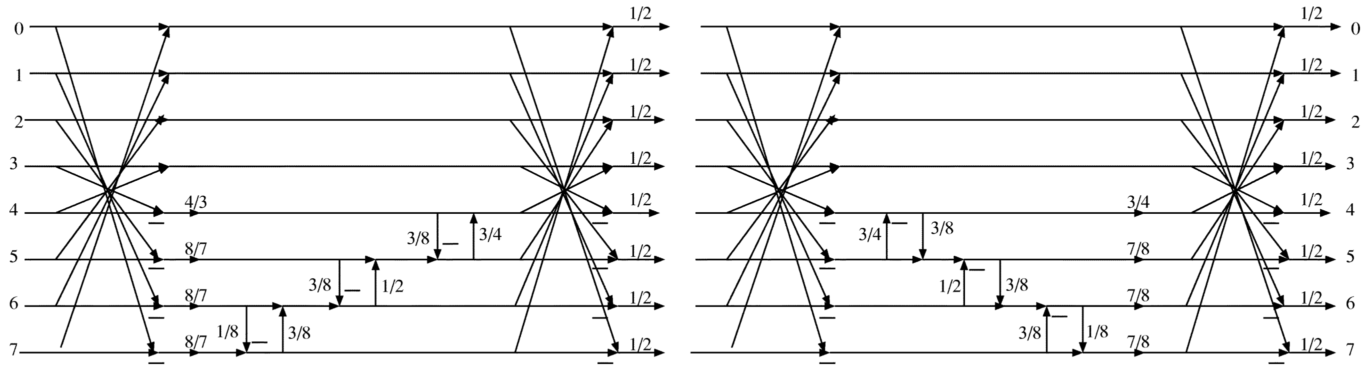


Fig. 3. (Left) Prefilter $\mathbf{P}$ and (right) the postfilter $\mathbf{P}^{-1}$.

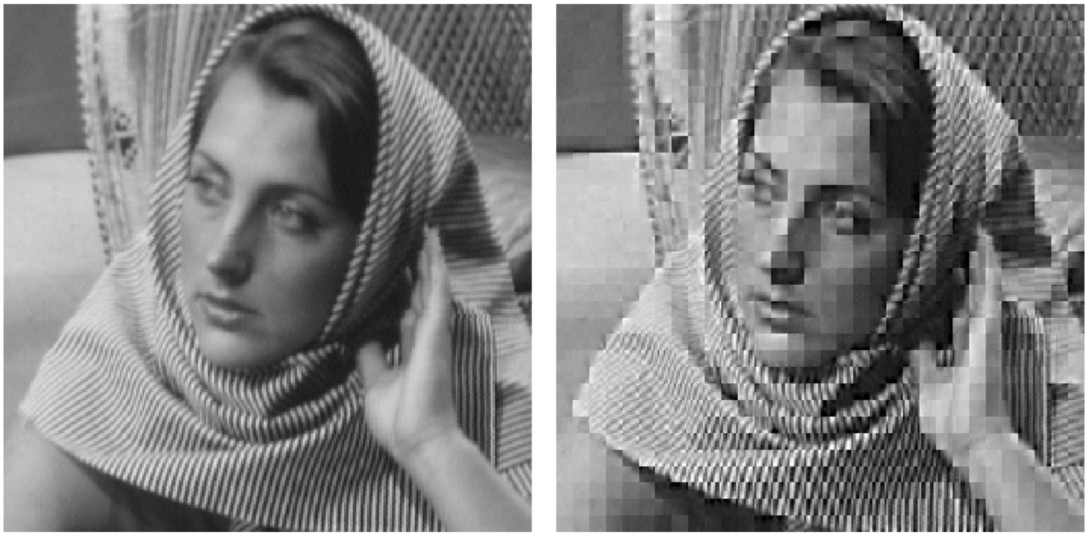


Fig. 4. Demonstration of the prefiltering effect. Left: original image. Right: prefiltered image before DCT transformation.

## III. Intrablock and Interblock Correlation

### A. Intrablock Correlation

The most well-known feature of a coefficient block in a natural image is that the magnitudes of AC coefficients decrease as the frequency increases. Besides, although the basis functions corresponding to different coefficients of a block are generally orthogonal or near orthogonal, two coefficients in a block are not independent. A coefficient is still correlated with its block neighbors. We call this correlation among neighbors within a local block intrablock (or intersubband) correlation.

However, algorithms exploiting only intra-block correlation are not expected to yield state-of-the-art R-D performance since:

1) intrablock correlation, especially for 8 × 8 block size, is limited to a small local block and only reflects the short-range correlation;
2) correlation is weak since coefficients at different locations correspond to different basis functions;
3) most coefficients are quantized to zero and, thus, not much information can be exploited.

### B. Interblock Correlation

Block transform subbands and wavelet subbands demonstrate similar characteristics. A block transform coefficient is strongly correlated with its subband neighbors, especially in low-frequency subbands. We call this type of correlation interblock (or intrasubband) correlation. Generally, interblock correlation is much stronger than intrablock correlation. Fig. 5 shows the block and subband representation of 8 × 8 DCT coefficients of the 512 × 512 Lena image. Spatial features are much more visually obvious in subbands than in blocks. The DC subband is a thumbnail image as is the lowpass $LL$ wavelet subband. Each of the other subbands is a bandpass-filtered thumbnail of Lena. Each subband contains a part of the global information of the image. Vertical (horizontal) features are easily recognized in the $V(H)$ subbands just as with the $LH(HL)$ wavelet subbands.

Exploiting interblock correlation is the key to eliminating frequency redundancy. This is a weakness of most block coding algorithms. For example, except for DPCM coding of DC coefficients, JPEG [15] completely ignores interblock correlation. Fully taking advantage of both intrablock and interblock correlation in the transforming stage with prefiltering/postfiltering and in the coding stage with advanced context-based adaptive entropy coding is the key to a vastly improved block-DCT-based image coder.

### C. Sign Correlation

It has been recently observed that sign correlation exists among wavelet coefficients. Many researchers have explored

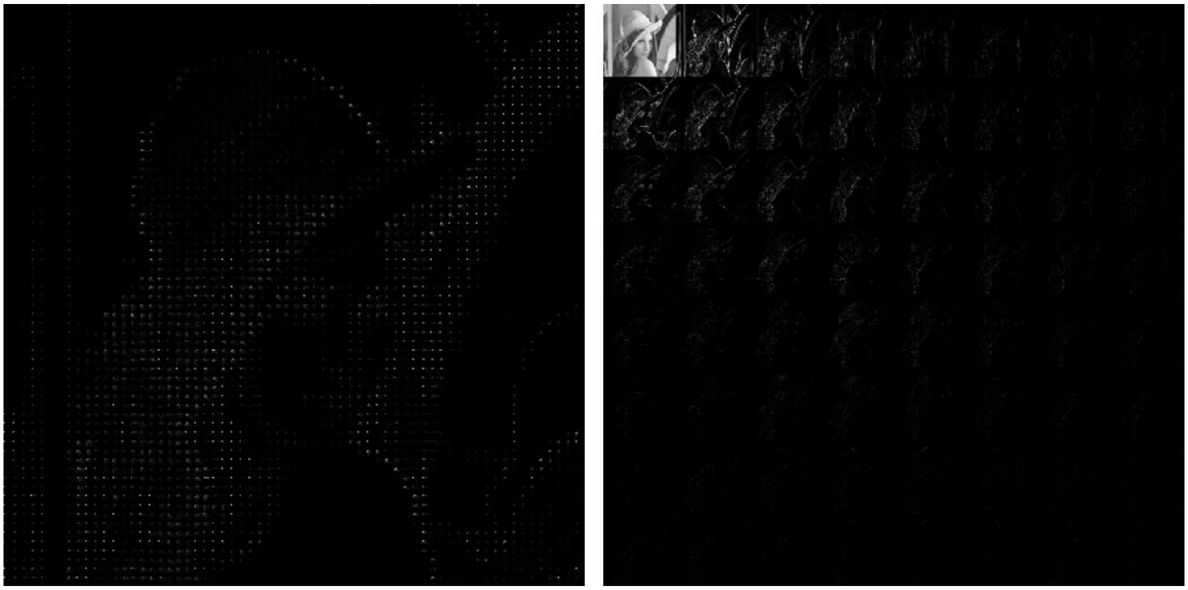


Fig. 5. Representations of block transform coefficients for the Lena image. Left: block representation. Right: subband representation.

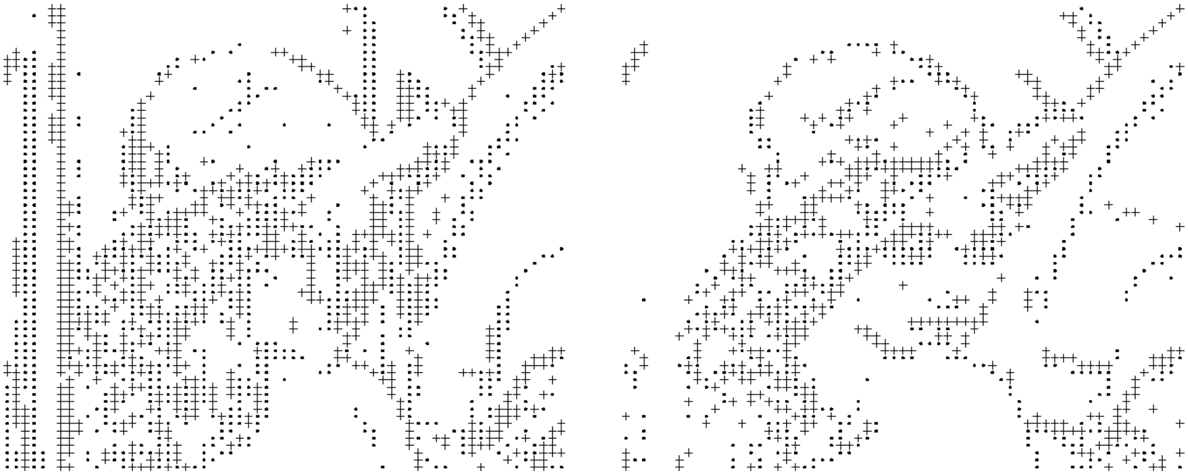


Fig. 6. DCT subband sign maps of Lena, "+" positive and "." negative. Left: (1, 0) subband. Right: (0, 1) subband.

the benefits of sign modeling [5], [6], [8], [9]. Besides offering better coding efficiency, sign modeling can also improve the final estimate of coefficients which are quantized to zero. Deever and Hemami provided a detailed analysis of sign behaviors of wavelet coefficients and demonstrate how much gain can be obtained by sign coding and sign extrapolation [16].

In the block-transform case, although sign correlation among block neighbors is weak, sign correlation among subband neighbors is strong especially in low-frequency subbands. Fig. 6 plots the sign maps of two 8 × 8 DCT subbands of the Lena image. Sign patterns do exist, especially along and across strong edges. Clearly, sign modeling for block transform coefficients is worthwhile.

## IV. Context Modeling of Block Transform Coefficients

Before encoding the quantized coefficients, they are first represented as a sequence of symbols $\{x_1, x_2, \ldots, x_n\}$ which is then compressed into a bitstream. The way to generate this sequence and the way to compress it should be jointly optimized. This paper is mainly concerned with the latter.

The minimum code-length of a sequence in bits is given by

$$-\log_2 \prod_{i=1}^{n} P\left(x_i | x^{i-1}\right) \qquad (3)$$

where $P(x_i|x^{i-1})$ is the conditional probability of $x_i$ given $x^{i-1}$, which denotes the sequence $\{x_{i-1}, x_{i-2}, \ldots, x_1\}$. Here, $x^{i-1}$ is the known information about $x_i$. Hence, $x^{i-1}$ is called the context of $x_i$.

The probability $P(x_i|x^{i-1})$ is highly image dependent and not known before coding $x_i$. CEB relies on adaptive arithmetic coding to estimate $P(x_i|x^{i-1})$ on the fly and allocates the bit budget accordingly. To do so, different contexts need different adaptive models and a tremendous number of models are necessary since $x^{i-1}$ may contain a large set of symbols. In practice, a large number of models not only increases the complexity, but also decreases the performance of the arithmetic coder because of context dilution, i.e., there are not enough samples to reach a good estimate of $P(x_i|x^{i-1})$.

The task of context modeling is to estimate $P(x_i|x^{i-1})$ using an affordable number of models. We formulate this problem as

$$x_i\colon f(x^{i-1}) \tag{4}$$

which means that the symbol $x_i$ is coded based on the adaptive model indexed $f(x^{i-1})$. Here, $f$ is a function of $x^{i-1}$ that maps the context $x^{i-1}$ to an estimated model of $P(x_i|x^{i-1})$. The function $f$ is a many-to-one mapping and, thus, maps a large number of contexts to a small number of context models. Generally, $f$ only depends on a small subset of $x^{i-1}$ containing the most relevant symbols regarding $x_i$.

Wu demonstrates the power of high-order context modeling in wavelet coding [5], [6] where context modeling is fully optimized for encoding wavelet coefficients. A similar gain can be expected from context modeling optimized for encoding block transform coefficients. Based on the discussion in Section III, the key point is that a block transform coefficient should be coded conditioned on known information of both its block neighbors and subband neighbors. More gain can be expected if different subbands are treated differently according to their characteristics. For example, models for the V subbands should be tuned toward capturing vertical edges. Usually, low-frequency subbands and high-frequency subbands should also be treated differently since correlation starts to disappear when the principal frequency decreases. The main innovation in CEB is its context modeling optimized in this manner.

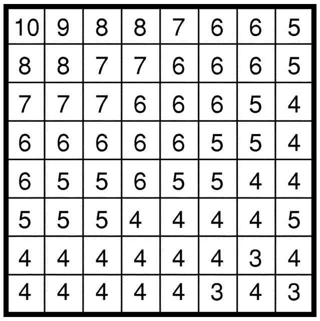
## V. Generic CEB Algorithm

CEB is a block-based algorithm. We treat transform coefficients in a subband context but never reorder them into subbands. Only a minimal amount of information from a few adjacent blocks needed by context modeling is buffered.

### A. L-CEB Algorithm

Local CEB (L-CEB), the local and low-complexity version of CEB, processes image blocks one by one from top to bottom and from left to right. Prefiltering, DCT, and postfiltering are performed block-wise. There is virtually no buffering except for the current block to be encoded. For each block, the coefficients are processed in four steps.

1) *Quantization:* Generally scalar quantization with a double dead zone is used. Other quantization methods may produce better objective and/or visual quality.
2) *Significance Testing:* Each quantized coefficient in the block should be tested to determine if it is significant (nonzero) or insignificant (zero).
3) *Magnitude Coding:* If a coefficient is significant, its magnitude is coded.
4) *Sign Coding:* Obviously, the sign of each significant coefficient should also be coded.



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 10 | 9 | 8 | 8 | 7 | 6 | 6 | 5 |
| 8 | 8 | 7 | 7 | 6 | 6 | 6 | 5 |
| 7 | 7 | 7 | 6 | 6 | 6 | 5 | 4 |
| 6 | 6 | 6 | 6 | 6 | 5 | 5 | 4 |
| 6 | 5 | 5 | 6 | 5 | 5 | 4 | 4 |
| 5 | 5 | 5 | 4 | 4 | 4 | 4 | 5 |
| 4 | 4 | 4 | 4 | 4 | 4 | 3 | 4 |
| 4 | 4 | 4 | 4 | 4 | 3 | 4 | 3 |

Fig. 7. Most significant bits of 8 × 8 DCT subbands of Lena.

### B. E-CEB Algorithm

Embedded-CEB (E-CEB), the progressive version of CEB, buffers all transform coefficients. When E-CEB is initialized, the highest bitplane is found and all coefficients are marked as insignificant. E-CEB then proceeds from the highest bitplane down to the lowest bitplane until a given bit rate or a given distortion level is achieved. In each bitplane, blocks are still encoded one by one from top to bottom and left to right. For the $b$th bitplane, each block is encoded as follows.

1) *Refinement Coding:* The $b$th bitplane of each previously significant coefficient is coded.
2) *Significance Testing:* Check the magnitude of every insignificant coefficient. Label it as significant if its magnitude is equal to or greater than $2^b$.
3) *Sign Coding:* The signs of coefficients which just became significant are coded.

For natural images, the DCT compacts most signal energy into low-frequency subbands. Consequently, low-frequency subband coefficients have much higher magnitudes than high-frequency subband coefficients. The distribution of the most significant bits of 8 × 8 DCT subbands of Lena in Fig. 7 illustrates this fact. The information of the most significant bits is not available to L-CEB, but is available to E-CEB since E-CEB buffers all coefficients. Knowing the most significant bits, E-CEB can avoid coding some significance information. For example, E-CEB never needs to code the significance information of (7, 7) subband coefficients before coding the third bitplane since they must be insignificant. In this manner, E-CEB saves a good number of bits on coding significance information. To do so, the encoder has to send the most significant bits to the decoder. The cost for coding the most significant bits is trivial comparing to the bandwidth savings.

### C. Significance Testing Strategy

Besides the quantization method, two aspects that differentiate different CEB implementations are the significance testing strategy and the coding scheme for signs, significance, and magnitude/refinement information. Significance testing is generally based on a certain group testing method: a group of coefficients is tested together and divided into subgroups if it is found significant. The standard group testing method for wavelet coders is the zerotree technology in EZW [2]. However, group testing becomes less important if the following context-based entropy coding is well designed. For instance, in the wavelet coders

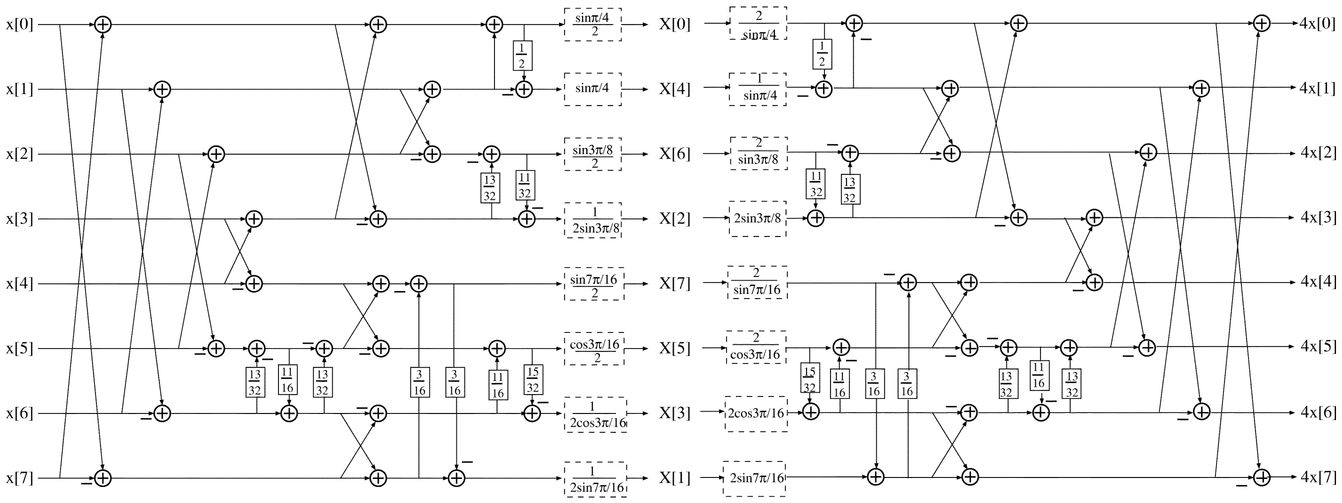


Fig. 8. Fast DCT implementation. Left: forward transform. Right: inverse transform.

ECECOW and FD [5], [6] which yield the best R-D performances so far in the literature, wavelet coefficients are tested one by one. In this paper, we concentrate on optimizing the coding of signs, significance, and magnitude/refinement information.

### D. Entropy Coding

Once a coefficient becomes significant, it is generally believed that entropy coding does not help much in coding later bitplanes. So refinement information is encoded in raw binary, i.e., one bit is used to indicate whether the bitplane is "0" or "1." Since the magnitude and sign of a coefficient are highly correlated with those of its block and subband neighbors, context-based entropy coding still helps substantially in coding all other information.

CEB employs binary adaptive arithmetic coding. This is the simplest and the fastest version of adaptive arithmetic coding, which can be easily implemented in both software and hardware [17]. Furthermore, it approaches the underlying context model very quickly. There is no problem in coding significance and sign information since they are binary. However, the magnitude of a significant coefficient is not binary. CEB binarizes them into sequences of binary symbols before coding. Specifically, CEB binarizes a nonbinary symbol $n$ into $n-1$ binary "0"s followed by a binary "1." For example, 4 is binarized as 0001. Different adaptive models can be used for different bins.

## VI. Specific CEB Implementations

This section describes in detail the implementation of a simple L-CEB coder and a simple E-CEB coder. Here we only deal with 8 × 8 blocks and the DCT, but it is easy to generalize to other block transforms of other sizes. The DCT implementation used in CEB is a "binary" DCT approximation [18] as illustrated in Fig. 8. Only integer additions and shifts are needed except for the final floating-point scaling which is combined with quantization.

Coefficients in a block are labeled in the zigzag order of JPEG as shown in Fig. 9. Coefficient $C_{x,y}^{i,j}$ is also denoted as $C_{x,y}^{l} (0 \leq l < 64)$ where $l$ is its zigzag label. For example, $C_{x,y}^{7} = C_{x,y}^{2,1}$.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 5 | 6 | 14 | 15 | 27 | 28 |
| 2 | 4 | 7 | 13 | 16 | 26 | 29 | 42 |
| 3 | 8 | 12 | 17 | 25 | 30 | 41 | 43 |
| 9 | 11 | 18 | 24 | 31 | 40 | 44 | 53 |
| 10 | 19 | 23 | 32 | 39 | 45 | 52 | 54 |
| 20 | 22 | 33 | 38 | 46 | 51 | 55 | 60 |
| 21 | 34 | 37 | 47 | 50 | 56 | 59 | 61 |
| 35 | 36 | 48 | 49 | 57 | 58 | 62 | 63 |

Fig. 9. Zigzag scanning.

Define $S_{x,y}^{l}$ as the set of coefficients in block $(x, y)$ whose zigzag label $k$ satisfies $k \geq l$

$$S_{x,y}^{l} = \{C_{x,y}^{l}, C_{x,y}^{l+1}, \ldots, C_{x,y}^{63}\}. \qquad (5)$$

We say $S_{x,y}^{l}$ is insignificant if all of its members are insignificant. Otherwise, it is said to be significant. Note that if $S_{x,y}^{l}$ is insignificant, then $S_{x,y}^{k}$ with $k > l$ is also insignificant. Secondly, if $S_{x,y}^{l}$ is significant, $C_{x,y}^{l}$ and $S_{x,y}^{l+1}$ can not be both insignificant because

$$S_{x,y}^{l} = \{C_{x,y}^{l}, S_{x,y}^{l+1}\}. \qquad (6)$$

Since optimizing significance testing algorithms is not our main concern in this paper, a simple sequential group testing algorithm is used: begin with $S_{x,y}^{0}$ and increment $l$; if $S_{x,y}^{l}$ is significant, it is subdivided into $C_{x,y}^{l}$ and $S_{x,y}^{l+1}$; this subdivision process is repeated until all significant coefficients are identified.

As mentioned in Section IV, in the context of modeling, different subbands should be treated differently. In our CEB implementations, we differentiate seven types of subbands as depicted in Fig. 10: one DC subband, one principal V subband (PV), one principal H subband (PH), three low-frequency V subbands (LV), three low-frequency H subbands (LH), eight

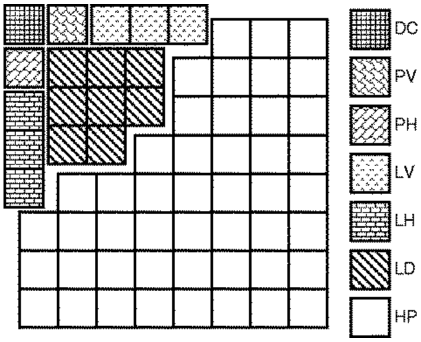


Fig. 10. Subband classification.

low-frequency D subbands (LD) and the remaining 47 high-pass subbands (HP). Subbands of the same type share the same context modeling. Usually, different adaptive models are used for subbands of different types since their characteristics differ greatly. The DC subband is nearly independent of the AC subbands. CEB does not use one as the context of the other.

### *A. L-CEB Implementation*

L-CEB is specifically designed for low-cost, low-power, high-speed real-time applications. We assume that onboard memory is very limited. Only a few adaptive models are allowed. The DC of block $(x, y)$, $C^{0,0}_{x,y}$, is predicted as the mean of the reconstructed DC coefficients from the left and top neighboring blocks, i.e., $(C^{0,0}_{x-1,y} + C^{0,0}_{x,y-1})/2$. The prediction error is quantized and coded. Simple scalar quantization with a double dead zone is used. We use $[C^l_{x,y}]$ and $[S^l_{x,y}]$ to denote the significance information of $C^l_{x,y}$ and $S^l_{x,y}$, respectively

$$[C^l_{x,y}] = \begin{cases} 1, & \text{if } C^l_{x,y} \text{ is significant} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

$$[S^l_{x,y}] = \begin{cases} 1, & \text{if } S^l_{x,y} \text{ is significant} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

The block-coding process proceeds as follows.

1) Find $l$, the zigzag label of the last significant coefficient.
2) Code $[C^0_{x,y}]$.
3) Code the magnitude and the sign of $C^0_{x,y}$ if $[C^0_{x,y}] = 1$.
4) Code $[S^1_{x,y}]$.
5) If $[S^1_{x,y}] = 1$ then for $i = 1$ to $l$
   - code $[C^i_{x,y}]$;
   - if $[C^i_{x,y}] = 1$ then
     — code the magnitude and the sign of $C^i_{x,y}$;
     — if $i < 63$ then code $[S^{i+1}_{x,y}]$, which is 1 if $i < l$ and 0 otherwise.

Since the DC coefficient, $C^0_{x,y}$, is not grouped with AC coefficients, we never code $[S^0_{x,y}]$. Note that coding $[S^{i+1}_{x,y}]$ if $[S^i_{x,y}] = 1$ and $[C^i_{x,y}] = 0$ should be avoided since $[S^{i+1}_{x,y}] = 1$ is already known. Also, since $C^{63}_{x,y}$ is the same as $S^{63}_{x,y}$, we should also avoid coding $[C^{63}_{x,y}]$ twice.

To keep L-CEB simple, signs are not entropy coded and one bit is spent for the sign information per significant coefficient. Magnitudes of significant coefficients are not binary but most of them are small. L-CEB codes magnitudes without using any context. Magnitudes are simply binarized as mentioned in Section V and coded based on binary adaptive models. A single model is used for all bins of the magnitude value. Four models are used for different subbands: one for DC; one for PV and PH; one for LV, LH, and LD; and one for HP.

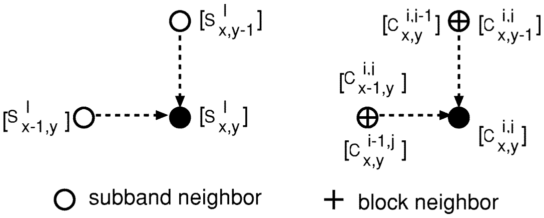


Fig. 11. Context modeling for L-CEB.

As an example of how L-CEB works, consider the encoding process of a quantized 8 × 8 block of transform coefficients shown in the following (actually taken from [19])

$$C_{x,y} = \begin{bmatrix} 3 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

L-CEB will map this coefficient block into the following string of symbols and binary bitstream

$$\begin{array}{lllllll} [C^0_{x,y}] & 3 & + & [S^1_{x,y}] & [C^1_{x,y}] & [C^2_{x,y}] & 2 \\ \mathbf{1} & \mathbf{001} & \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{01} \\ \\ - & [S^3_{x,y}] & [C^3_{x,y}] & 1 & - & [S^4_{x,y}] & [C^4_{x,y}] \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \\ 1 & - & [S^5_{x,y}] & [C^5_{x,y}] & 1 & - & [S^6_{x,y}] \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \\ [C^6_{x,y}] & [C^7_{x,y}] & [C^8_{x,y}] & 1 & - & [S^9_{x,y}] & \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0}. & \end{array}$$

A total of 30 bits is used to represent the data block. However, this is only raw binary and the reader can easily observe that there is still a lot of redundancy in the bitstream. Except for the six binary symbols representing signs, all 24 remaining binary symbols will be further compressed by arithmetic coding. The 15 significance symbols, $[S^i_{x,y}]$ and $[C^i_{x,y}]$, will be coded conditioned on the known information of neighboring blocks. In the same example, JPEG spends 31 bits in its runlength coding scheme

$$\begin{array}{llll} (2)(3) & (1,2)(-2) & (0,1)(-1) & (0,1)(-1) \\ \mathbf{001\ 11} & \mathbf{11\ 011\ 01} & \mathbf{00\ 0} & \mathbf{00\ 0} \\ \\ (0,1)(-1) & (2,1)(-1) & (0,0) & \\ \mathbf{00\ 0} & \mathbf{11\ 100\ 0} & \mathbf{1010}. & \end{array}$$

However, JPEG has already utilized entropy coding (Huffman) and the above is the final bitstream.

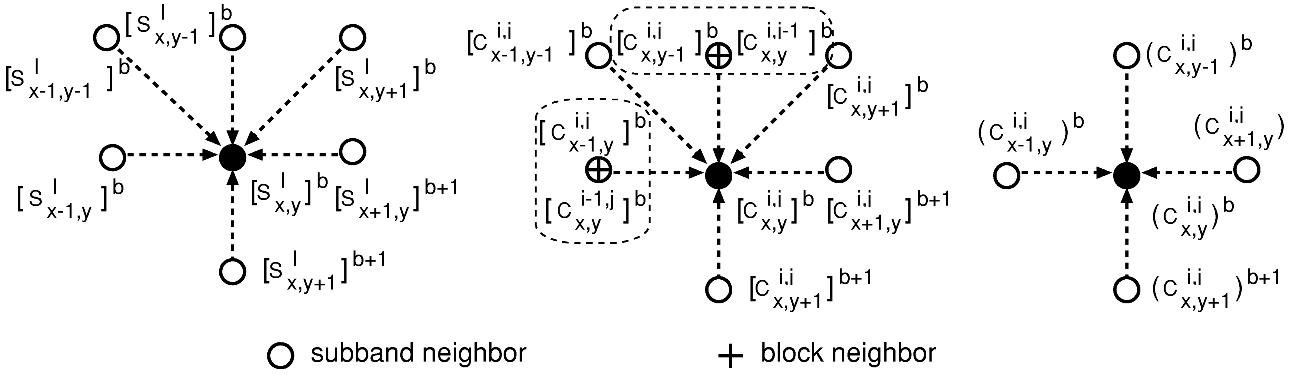


Fig. 12. Context modeling for E-CEB.

As previously mentioned, the only information coded by L-CEB with context-based arithmetic coding is the significance information. L-CEB uses a simple context modeling method illustrated in Fig. 11: the significance information of $S^l_{x,\,y}$ is coded conditioned on the number of significant left subband neighbor ($S^l_{x-1,\,y}$) and top subband neighbor ($S^l_{x,\,y-1}$); the significant information of $C^{i,\,j}_{x,\,y}$ is coded conditioned on how many of its left subband neighbor ($C^{i,\,j}_{x-1,\,y}$), top subband neighbor ($C^{i,\,j}_{x,\,y-1}$), left block neighbor ($C^{i-1,\,j}_{x,\,y}$) and top block neighbor ($C^{i,\,j-1}_{x,\,y}$) are found significant. If any of the block (subband) neighbors is out of block (image) boundaries, it is assumed to be insignificant.

Following the notation of Section IV, the detailed context modeling is given by

$$[S^l_{x,\,y}]\text{: } [S^l_{x-1,\,y}] + [S^l_{x,\,y-1}] + \begin{cases} 0, & \text{if } l = 1 \\ 3, & \text{if } l = 2 \\ 6, & \text{if } 2 < l < 15 \\ 9, & \text{if } l > 14. \end{cases} \tag{9}$$

$$\begin{aligned}&[C^{i,\,j}_{x,\,y}]\text{: } [C^{i,\,j}_{x-1,\,y}] + [C^{i,\,j}_{x,\,y-1}] \\ &\quad + \begin{cases} 12, & \text{if } C^{i,\,j}_{x,\,y} \in DC \\ 15, & \text{if } C^{i,\,j}_{x,\,y} \in PV \\ 18, & \text{if } C^{i,\,j}_{x,\,y} \in PH \\ 21 + [C^{i-1,\,j}_{x,\,y}], & \text{if } C^{i,\,j}_{x,\,y} \in LV \\ 25 + [C^{i,\,j-1}_{x,\,y}], & \text{if } C^{i,\,j}_{x,\,y} \in LH \\ 29 + [C^{i-1,\,j}_{x,\,y}] + [C^{i,\,j-1}_{x,\,y}], & \text{if } C^{i,\,j}_{x,\,y} \in LD \\ 34 + [C^{i-1,\,j}_{x,\,y}] + [C^{i,\,j-1}_{x,\,y}], & \text{if } C^{i,\,j}_{x,\,y} \in HP. \end{cases}\end{aligned} \tag{10}$$

Different models are used for different subband classes. Adaptive models are $x$, $y$-independent. L-CEB needs a total of 38 adaptive models to code significance information and four adaptive models to code magnitudes. Altogether only 42 binary models are necessary.

Significance information of one horizontal slice of blocks should be buffered for context modeling. Each block needs 64 bits. We also need to buffer DC coefficients of the slice to perform DC prediction. Besides the trivial memory needed for the 64 transform coefficients and the 42 binary models, L-CEB only requires to buffer $W(64 + 32)/8 = 12\ W$ bits, or 1.5 $W$ bytes, to code an image of width $W$. L-CEB can easily fit an 8 K memory footprint. The combination of simple context modeling, simple group testing, binary arithmetic coding, fast prefiltering/postfiltering and fast DCT implementation makes L-CEB ideal for handheld devices such as PDAs and cellular phones.

### *B. E-CEB Implementation*

In E-CEB, an index, $l_{x,\,y}$, is associated with block $(x,\,y)$ to indicate the zigzag label of the last significant coefficient. When E-CEB is initialized, $l_{x,\,y}$ is set to zero and all coefficients are labeled as insignificant. Let $m^l$ be the most significant bit of the subband with zigzag label $l$. Let $|C^l_{x,\,y}|^b$ be the $b$th bitplane of $C^l_{x,\,y}$. Denote $[C^l_{x,\,y}]^b$, $(C^l_{x,\,y})^b$, and $[S^l_{x,\,y}]^b$ the significance of $C^l_{x,\,y}$, the sign of $C^l_{x,\,y}$, and the significance of $S^l_{x,\,y}$ at the $b$th bitplane

$$[C^l_{x,\,y}]^b = \begin{cases} 1, & \text{if } C^l_{x,\,y} \text{ is significant at } b\text{th bitplane} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

$$(C^l_{x,\,y})^b = \begin{cases} 1, & \text{if } [C^l_{x,\,y}]^b = 1 \text{ and } C^l_{x,\,y} > 0 \\ -1, & \text{if } [C^l_{x,\,y}]^b = 1 \text{ and } C^l_{x,\,y} < 0 \\ 0, & \text{if } [C^l_{x,\,y}]^b = 0 \end{cases} \tag{12}$$

$$[S^l_{x,\,y}]^b = \begin{cases} 1, & \text{if } S^l_{x,\,y} \text{ is significant at } b\text{th bitplane} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

The process of encoding the $b$th bitplane of a certain block $(x,\,y)$ is as follows:

```
1) For $i = 0$ to $l_{x,\,y}$
    • code $|C^i_{x,\,y}|^b$, if $C^i_{x,\,y}$ is significant;
    • if $C^i_{x,\,y}$ is insignificant then
        — code $[C^i_{x,\,y}]^b$, if $b \le m^i$;
        — if $[C^i_{x,\,y}]^b = 1$ then
            * label $C^i_{x,\,y}$ as significant;
            * code $(C^i_{x,\,y})^b$.
2) If $l_{x,\,y} < 63$, then
    • code $[S^{l_{x,\,y}+1}_{x,\,y}]^b$, if $b \le m^j (j \le l_{x,\,y} + 1)$;
    • if $[S^{l_{x,\,y}+1}_{x,\,y}]^b = 1$ then for $i = l_{x,\,y} + 1$ to 63,
        — set $l_{x,\,y} = i$;
        — code $[C^i_{x,\,y}]^b$, if $b \le m^i$;
        — if $[C^i_{x,\,y}]^b = 1$ then
            * label $C^i_{x,\,y}$ as significant;
            * code $(C^i_{x,\,y})^b$;
            * code $[S^{i+1}_{x,\,y}]^b$, if $i < 63$ and $b \le m^j (j \le i + 1)$.
        — stop, if $i = 63$ or $[S^{i+1}_{x,\,y}]^b = 0$.
```

TABLE I
CODING COMPARISON. BIT RATE (bpp)/PSNR (dB)

| | 9/7 wavelet | | 8 × 8 DCT | | | 8 × 8 DCT with pre/post | | |
|---|---|---|---|---|---|---|---|---|
| bpp | SPIHT | JPEG2000-SL | EZ | L-CEB | E-CEB | EZ | L-CEB | E-CEB |
| Lena (512 × 512) | | | | | | | | |
| 0.0625 | **28.38** | 28.30 | 26.90 | 26.67 | 26.82 | 28.14 | 27.91 | 28.07 |
| 0.125 | 31.10 | **31.22** | 29.59 | 29.61 | 29.83 | 30.98 | 30.99 | **31.22** |
| 0.25 | 34.11 | 34.28 | 32.82 | 32.94 | 33.16 | 34.07 | 34.22 | **34.43** |
| 0.5 | 37.21 | 37.43 | 36.24 | 36.34 | 36.63 | 37.12 | 37.19 | **37.46** |
| 1.0 | 40.41 | **40.61** | 39.61 | 39.54 | 40.08 | 39.89 | 40.03 | 40.43 |
| Goldhill (512 × 512) | | | | | | | | |
| 0.0625 | 26.73 | 26.74 | 26.00 | 25.90 | 25.96 | 26.77 | 26.79 | **26.97** |
| 0.125 | 28.48 | 28.58 | 27.85 | 27.90 | 27.99 | 28.57 | 28.64 | **28.73** |
| 0.25 | 30.55 | 30.71 | 30.00 | 30.14 | 30.23 | 30.71 | 30.84 | **30.94** |
| 0.5 | 33.12 | 33.35 | 32.72 | 32.80 | 32.97 | 33.31 | 33.46 | **33.60** |
| 1.0 | 36.54 | 36.72 | 36.24 | 36.22 | 36.51 | 36.60 | 36.80 | **37.04** |
| Barbara (512 × 512) | | | | | | | | |
| 0.0625 | 23.35 | 23.45 | 22.71 | 22.59 | 22.73 | 23.74 | 23.86 | **24.10** |
| 0.125 | 24.86 | 25.55 | 24.60 | 24.82 | 24.99 | 25.93 | 26.42 | **26.52** |
| 0.25 | 27.58 | 28.55 | 27.23 | 27.68 | 27.83 | 28.95 | 29.64 | **29.76** |
| 0.5 | 31.39 | 32.48 | 31.14 | 31.62 | 31.91 | 32.91 | 33.56 | **33.75** |
| 1.0 | 36.41 | 37.37 | 36.24 | 36.54 | 36.98 | 37.57 | 37.98 | **38.38** |
| Bike (2048 × 2560) | | | | | | | | |
| 0.0625 | 23.44 | **23.89** | 22.72 | 22.77 | 22.89 | 23.47 | 23.39 | 23.65 |
| 0.125 | 25.89 | 26.49 | 25.44 | 25.70 | 25.83 | 26.13 | 26.43 | **26.63** |
| 0.25 | 29.12 | 29.76 | 28.74 | 29.02 | 29.19 | 29.31 | 29.63 | **29.83** |
| 0.5 | 33.01 | **33.68** | 32.65 | 32.82 | 33.12 | 33.09 | 33.27 | 33.66 |
| 1.0 | 37.70 | **38.29** | 37.35 | 37.29 | 37.77 | 37.40 | 37.58 | 38.14 |
| Cafe (2048 × 2560) | | | | | | | | |
| 0.0625 | 18.95 | **19.10** | 18.50 | 18.40 | 18.50 | 18.97 | 18.71 | 18.83 |
| 0.125 | 20.67 | 20.88 | 20.24 | 20.22 | 20.40 | 20.73 | 20.74 | **20.95** |
| 0.25 | 23.03 | 23.29 | 22.71 | 22.74 | 22.97 | 23.22 | 23.43 | **23.54** |
| 0.5 | 26.49 | 27.00 | 26.71 | 26.40 | 26.59 | 26.78 | 26.97 | **27.17** |
| 1.0 | 31.74 | **32.27** | 31.43 | 31.45 | 31.73 | 31.75 | 31.86 | 32.23 |
| Woman (2048 × 2560) | | | | | | | | |
| 0.0625 | 25.43 | **25.67** | 25.06 | 24.80 | 24.98 | 25.32 | 25.07 | 25.28 |
| 0.125 | 27.33 | 27.46 | 26.92 | 26.90 | 27.10 | 27.22 | 27.31 | **27.53** |
| 0.25 | 29.95 | 30.15 | 29.53 | 29.60 | 29.84 | 30.07 | 30.07 | **30.26** |
| 0.5 | 33.59 | 33.81 | 33.21 | 33.28 | 33.62 | 33.63 | 33.68 | **34.06** |
| 1.0 | 38.28 | **38.67** | 37.93 | 38.00 | 38.38 | 38.03 | 38.11 | 38.60 |
| First luminance frame of News (176 × 144) | | | | | | | | |
| 0.0625 | 20.63 | 21.08 | 20.94 | 21.39 | 21.17 | 21.45 | **21.90** | 21.81 |
| 0.125 | 23.03 | 23.31 | 22.99 | 23.49 | 23.47 | 23.52 | **24.16** | 24.00 |
| 0.25 | 26.12 | 26.48 | 25.83 | 26.60 | 26.46 | 26.46 | **27.05** | **27.05** |
| 0.5 | 30.24 | 30.38 | 30.11 | 30.75 | 30.62 | 30.59 | **31.21** | 31.18 |
| 1.0 | 35.94 | 36.59 | 35.94 | 36.16 | 36.40 | 36.27 | 36.71 | **36.83** |
| First luminance frame of Glasgow (176 × 144) | | | | | | | | |
| 0.0625 | 21.32 | 21.53 | 21.29 | 21.72 | 21.65 | 21.70 | **22.19** | 22.13 |
| 0.125 | 22.81 | 22.98 | 22.81 | 23.24 | 23.21 | 23.36 | **23.82** | 23.72 |
| 0.25 | 24.82 | 24.96 | 24.94 | 25.40 | 25.30 | 25.53 | **25.89** | 25.75 |
| 0.5 | 27.53 | 27.68 | 27.85 | 28.23 | 28.11 | 28.24 | **28.57** | 28.48 |
| 1.0 | 32.33 | 32.47 | 32.31 | 32.68 | 32.62 | 32.58 | 32.80 | **32.82** |

Note that $|C^i_{x,y}|^b$ is not entropy coded. All other information, including signs, is coded by context-based binary adaptive entropy coding. The context modeling for coding significance information is similar to the L-CEB context modeling for coding significance information. The only difference is that more subband neighbors are involved as shown in Fig. 12

$$[S^l_{x,y}]^b\colon [S^l_{x-1,y-1}]^b + [S^l_{x,y-1}]^b + [S^l_{x+1,y-1}]^b$$

$$+ [S^l_{x-1,\,y}]^b + [S^l_{x+1,\,y}]^{b+1} + [S^l_{x,\,y+1}]^{b+1} + \begin{cases} 0, & \text{if } l = 1 \\ 7, & \text{if } l = 2 \\ 14, & \text{if } 2 < l < 15 \\ 21, & \text{if } l > 14 \end{cases} \tag{14}$$

$$\begin{aligned}[C^{i,\,j}_{x,\,y}]^b\colon & [C^{i,\,j}_{x-1,\,y-1}]^b + [C^{i,\,j}_{x,\,y-1}]^b + [C^{i,\,j}_{x+1,\,y-1}]^b \\ & + [C^{i,\,j}_{x-1,\,y}]^b + [C^{i,\,j}_{x+1,\,y}]^{b+1} + [C^{i,\,j}_{x,\,y+1}]^{b+1} \\ & + \begin{cases} 28, & \text{if } C^{i,\,j}_{x,\,y} \in DC \\ 35, & \text{if } C^{i,\,j}_{x,\,y} \in PV \\ 42, & \text{if } C^{i,\,j}_{x,\,y} \in PH \\ 49 + [C^{i-1,\,j}_{x,\,y}]^b, & \text{if } C^{i,\,j}_{x,\,y} \in LV \\ 56 + [C^{i,\,j-1}_{x,\,y}]^b, & \text{if } C^{i,\,j}_{x,\,y} \in LH \\ 64 + [C^{i-1,\,j}_{x,\,y}]^b + [C^{i,\,j-1}_{x,\,y}]^b, & \text{if } C^{i,\,j}_{x,\,y} \in LD \\ 73 + [C^{i-1,\,j}_{x,\,y}]^b + [C^{i,\,j-1}_{x,\,y}]^b, & \text{if } C^{i,\,j}_{x,\,y} \in HP. \end{cases}\end{aligned} \tag{15}$$

The sign of $C^l_{x,\,y}$ is coded conditioned on the signs of its left, right, top, and bottom subband neighbors (Fig. 12)

$$\begin{aligned}(C^l_{x,\,y})^b\colon & 3 \times p((C^l_{x,\,y-1})^b + (C^l_{x,\,y+1})^{b+1}) + p((C^l_{x-1,\,y})^b \\ & + (C^l_{x+1,\,y})^{b+1}) + \begin{cases} 82, & \text{if } C^l_{x,\,y} \in DC \\ 91, & \text{if } C^l_{x,\,y} \in PV \cup LV \\ 100, & \text{if } C^l_{x,\,y} \in PH \cup LH \\ 109, & \text{if } C^l_{x,\,y} \in LD \cup HP \end{cases}\end{aligned} \tag{16}$$

where the function $p$ is defined as

$$p(x) = \begin{cases} 0, & \text{if } x = 0 \\ 1, & \text{if } x < 0 \\ 2, & \text{if } x > 0. \end{cases} \tag{17}$$

Altogether we have a total of 118 binary models, 36 of which are for sign coding. Notice that when coding the $b$th bitplane, all information about the $k$th $(k > b)$ bitplane is known. So some information from the $(b+1)$th bitplane is used as context.

The only nontrivial memory other than the buffering of all DCT coefficients that E-CEB needs is one byte per block to store the $l_{x,\,y}$ associated with the block. Unlike most zerotree-based progressive algorithms such as SPIHT, E-CEB does not utilize any expensive list operation. Therefore, E-CEB is competitive with any embedded algorithm in term of complexity as well as speed.

## VII. Coding Results

Table I provides a detailed comparison of our CEB coding results with those of current state-of-the-art block transform coding algorithms as well as of wavelet coding algorithms. Three groups of 8-bit grayscale images are used: the 512 × 512 images Lena, Goldhill, and Barbara; the 2048 × 2560 images Bike, Cafe, and Woman; the first luminance frames of QCIF (176 × 144) video sequences News and Glasgow.

The wavelet coders in comparison are SPIHT [3] with arithmetic coding and JPEG2000 [9] in the single layer (SL) mode. JPEG2000-SL is optimized for R-D performance and is not scalable. The Daubechies 9/7 filters [20] are used in both SPIHT and JPEG2000. Four-, five-, and six-level dyadic wavelet decompositions are employed for images of size 176× 144, 512 × 512, and 2048 × 2560, respectively. The two progressive block transform coders in comparison are labeled EZ: one is based only on the 8-point DCT whereas the other has prefiltering/postfiltering turned on [11]. In both EZ coders, additional 9/7 wavelet decomposition is performed on the DC subband to ensure a fair comparison, i.e., to enforce the same tree depth.

CEB with prefiltering and postfiltering outperforms all other methods most of the time, especially at medium bit rates. Sometimes CEB's performance can be slightly below EZ's. This is due solely to the fact that EZ is more efficient in handling DC coefficients (EZ uses wavelet decomposition to further decorrelate the DC subband). Although L-CEB is much simpler than E-CEB, E-CEB can only outperform L-CEB by a small margin. For low-resolution images, the wavelet transform starts to lose its global advantage and CEB outperforms SPIHT as well as JPEG2000 by a large margin. To be fair, the 114-byte header of JPEG2000 has been compensated.

Portions of reconstructed Barbara images at 0.125 bpp are shown in Fig. 13. With prefiltering/postfiltering turned on, blocking artifacts are eliminated. Lots of fine features, which are not visible in the portions reconstructed by SPIHT and JPEG2000, are clear in the portions reconstructed by CEB with preprocessing/postprocessing. In these coding examples, E-CEB gives the best overall visual quality.

In short, compared to CEB, especially L-CEB, other algorithms are a lot more complex and have a much larger memory requirement. Our results demonstrate that block transform coders, and if designed appropriately, can yield as high R-D performances as current state-of-the-art wavelet coders while requiring a much lower level of complexity.

## VIII. Conclusion

In this paper, we point out an important feature of block transform coefficients that can be exploited in compression applications: a coefficient is highly correlated with its block neighbors as well as its subband neighbors. This space–frequency relationship leads to a simple, yet efficient, context-based entropy coding algorithm of block transform coefficients (CEB). The generic approach and two simple versions of CEB are presented: the local CEB (L-CEB) and the embedded CEB (E-CEB). Outstanding coding performance of CEB demonstrates the power of high-order context modeling of block transform coefficients. Finally, there is still room for CEB improvements: better context-modeling, better coding methods for DC coefficients, adaptive prefiltering/postfiltering, variable transform block sizes, etc. CEB is also perfectly suited for the block-based coding framework popular in video coding.

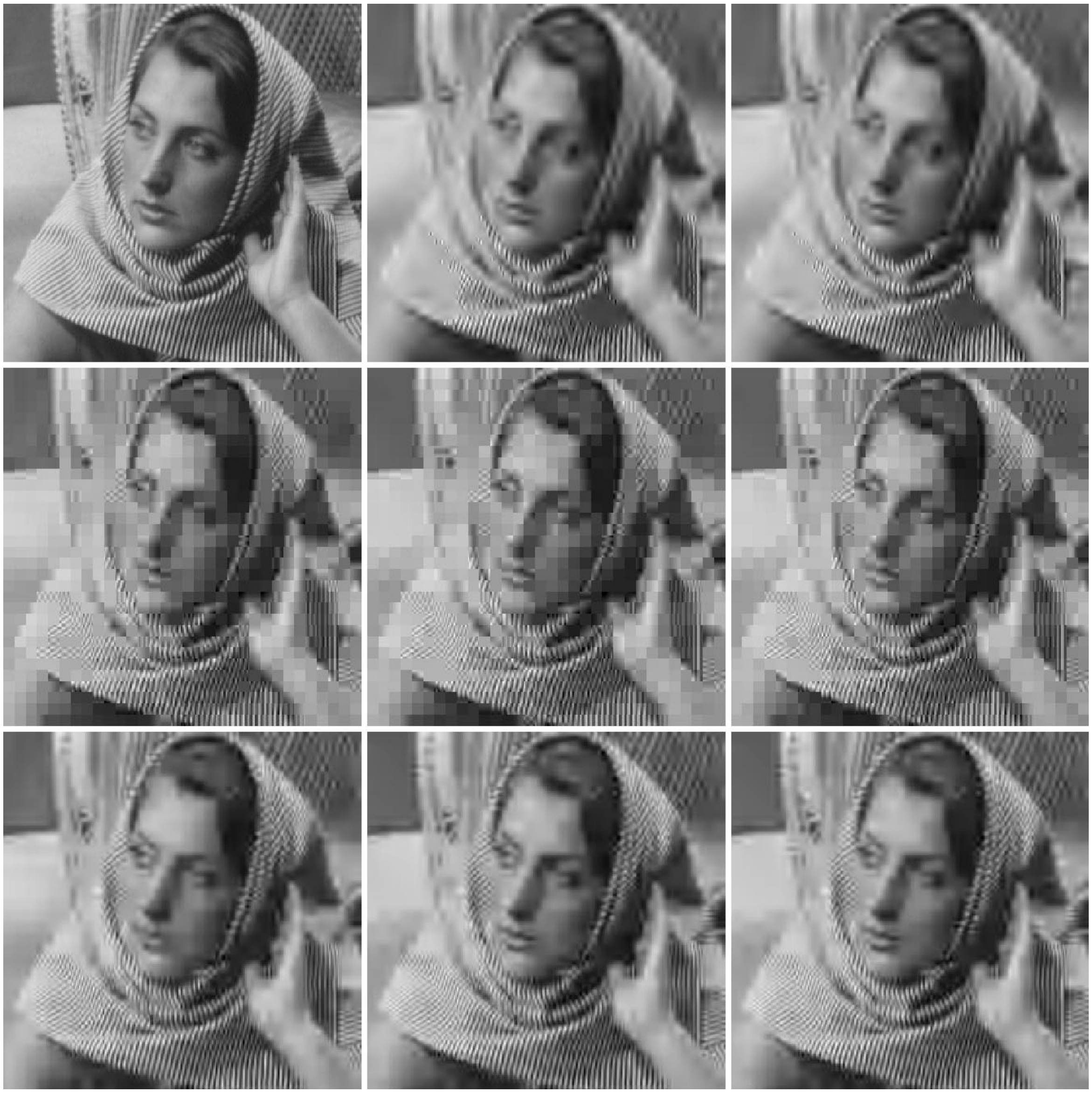


Fig. 13. Enlarged 256 × 256 portions of the Barbara image coded at 0.125 bpp. From left to right, top to bottom: original image; coded by SPIHT: 24.86 dB; coded by JPEG2000-LS: 25.55 dB; coded by EZ: 24.60 dB; coded by L-CEB: 24.82 dB; coded by E-CEB: 24.99 dB; coded by EZ with pre/post: 25.93 dB; coded by L-CEB with pre/post: 26.42 dB; coded by E-CEB with pre/post: 26.52 dB.

## Acknowledgment

The authors would like to thank the anonymous reviewers for their thorough reviews and numerous constructive suggestions which significantly enhance the presentation of the paper.

## References

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90–93, Jan. 1974.

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

[3] A. Said and W. A. Pearlman, "New, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits, Syst., Video Technol.*, vol. 6, pp. 243–249, June 1996.

[4] C. Chrysafis and A. Ortega, "Efficient context-based entropy coding for lossy wavelet image compression," in *Proc. 1997 Data Compression Conf.*, Mar. 1997, pp. 241–250.

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

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

[7] E. S. Hong and R. E. Ladner, "Group testing for image compression," in *Proc. 2000 Data Compression Conf.*, 2000, pp. 3–12.

[8] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, pp. 1158–1170, July 2000.

[9] "JPEG-2000 VM3,1A Software," ISO/IEC JTC1/SC29/WG1 N1142, Jan. 1999.

[10] Z. Xiong, O. Guleryuz, and M. T. Orchard, "A DCT-based embedded image coder," *IEEE Signal Processing Lett.*, vol. 3, pp. 289–290, Nov. 1996.

[11] T. D. Tran and T. Q. Nguyen, "A progressive transmission image coder using linear phase uniform filterbanks as block transforms," *IEEE Trans. Image Processing*, vol. 8, pp. 1493–1507, Nov. 1999.

[12] H. S. Malvar, "Fast progressive image coding without wavelets," in *Proc. 2000 Data Compression Conf.*, Mar. 2000, pp. 243–252.

[13] T. D. Tran, "Lapped transform via time-domain pre- and post-processing," in *Proc. 35th Conf. Information Sciences Systems (CISS01)*, Baltimore, MD, Mar. 2001, pp. 890–895.

[14] J. Liang, C. Tu, and T. D. Tran, "Fast lapped transforms via time-domain pre- and post-filtering," in *Proc. ICICS*, Singapore, Oct. 2001.

[15] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression*. New York: Van Nostrand Reinhold, 1992.

[16] A. Deever and S. S. Hemami, "What's your sign?: Efficient sign coding for embedded wavelet image coding," in *Proc. 2000 Data Compression Conf.*, Mar. 2000, pp. 273–282.

[17] W. B. Pennebaker, J. L. Mitchell, G. 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. 6, no. 3, pp. 243–249, Nov. 1988.

[18] J. Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Trans. Signal Processing*, vol. 49, pp. 3032–3044, Dec. 2001.

[19] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consumer Electronics*, vol. 38, pp. 18–34, Feb. 1992.

[20] M. Atonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. Image Processing*, vol. 1, pp. 205–220, Apr. 1992.

[21] H. S. Malvar, *Signal Processing With Lapped Transforms*. Norwood, MA: Artech House, 1992.

**Chengjie Tu** (S'02) received the B.E. and M.E. degrees from the University of Science and Technology of China (USTC) in 1994 and 1997, respectively, and the M.S.E degree from The Johns Hopkins University, Baltimore, MD, in 2002. He has been pursuing the Ph.D. degree at the Department of Electrical and Computer Engineering, The Johns Hopkins University, since 1999.

His current research interests include multirate signal processing, image/video compression, and error control and concealment for image/video communication.

**Trac D. Tran** (S'94–M'98) received the B.S. and M.S. degrees from the Massachusetts Institute of Technology, Cambridge, in 1993 and 1994, respectively, and the Ph.D. degree from the University of Wisconsin, Madison, in 1998, all in electrical engineering.

He joined the Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD, in July 1998 as an Assistant Professor. His research interests are in the field of digital signal processing, particularly in multirate systems, filter banks, transforms, wavelets, and their applications in signal analysis, compression, processing, and communications. He was the co-director (with J. L. Prince) of the 33rd Annual Conference on Information Sciences and Systems (CISS'99), Baltimore, in March 1999. In the summer of 2002, he was an ASEE/ONR Summer Faculty Research Fellow at the Naval Air Warfare Center Weapons Division (NAWCWD), China Lake, CA.

Dr. Tran received the NSF CAREER award in 2001. He currently serves as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING.