On the Improvements of Embedded Zerotree Wavelet (EZW) Coding

Jin Li and Po-Yuen Cheng and C.-C. Jay Kuo

Signal and Image Processing Institute and Department of Electrical Engineering-Systems
University of Southern California, Los Angeles, California 90089-2564

ABSTRACT

In this research, we investigate several improvements of embedded zerotree wavelet (EZW) coding. Several topics addressed include: the choice of wavelet transforms and boundary conditions, the use of arithmetic coder and arithmetic context and the design of encoding order for effective embedding. The superior performance of our improvements is demonstrated with extensive experimental results.

Keywords: wavelet transform, boundary extension, arithmetic context, embedded coding

1 INTRODUCTION

A novel still image compression scheme based on the embedded zerotree wavelet (EZW) coding idea was recently proposed by J. Shapiro [10]. In addition to giving an excellent rate-distortion performance, the method has the property that bits in the bit stream are generated in the order of significance so that it is ideal for progressive transmission. Later, to generalize the zerotree concept, T. Taubman and A. Zakhor[12] derived a improved version of EZW by introducing a generic prediction structure, known as the layer zero coding (LZC). In this research, we investigate each component of EZW, and propose ways to improve its performance. Several topics are addressed in the research, including the choice of wavelet transforms and boundary conditions, the use of arithmetic coder and arithmetic context and the design of encoding order for effective embedding.

Generally, data compression algorithms can be classified into three separate stages: transform, quantization and entropy coding. The transform projects the original image onto a set of basis functions. Because of the nature of the image signal and mechanisms of human visions, the transform used must accept nonstationarity and be well localized in both the space and frequency domains. The 2-D wavelet transform provides an excellent candidate for the task. Some criteria in developing the wavelet basis include the length of the filter for efficient computation, the energy compaction ratio, and the linear phase characteristic. Several wavelet basis functions are examined and their performance is compared in this work. Discrete wavelet transform is designed to operate on infinite length data. To operate it on a finite image, we have to extend the data. Besides, the input image size may not be a multiple of 2^n so that byte stuffing is required in performing the transform. Issues related to data extension and byte stuffing are also discussed in this work.

The major difference between the embedded coding and the traditional coding lies in the quantization. In this work, embedded coding is explained in a simple unified framework. We point out that the main feature of embedded coding is its successive quantization, where coding residue is reduced step by step. The embedded coding idea can also be applied to other image coding methods such as DCT. Embedded coding uses a simplified quantizer with a sophisticated arithmetic coder to achieve high compression and the performance of the arithmetic coder is directly related to coding gain. Arithmetic prediction context, which serves as the essential part of the arithmetic

coder, is investigated thoroughly in this research. For optimum coding, embedded coder should organize bits in a decreasing importance order, we characterize this optimum condition as a decreasing unitary MSE criterion and show a procedure to determine the proper order of embedding.

This paper is organized as follows. The wavelet transform and boundary conditions are discussed in Section 2. We describe a general framework for embedded coding and examine the arithmetic coder and its context in Section 3. Experimental results are provided in Section 4. Some concluding remarks and possible extensions are given in Section 5.

2 ENERGY COMPACTION BY WAVELET TRANSFORM

The wavelet transform provides a multiresolution tool for signal analysis. The two-scale discrete wavelet transform of a sequence $f[n]$ can be viewed as passing the signal through a quadrature mirror filter (QMF) consisting of a low- and high-pass filter pair denoted, respectively, by $h[k]$ and $g[k]$ with $g[k] = (-1)^k h[1-k]$. The forward transform can be written as

$$\begin{aligned} c_k &= \sqrt{2}\textstyle\sum_n h[n-2k]f[n], \\ d_k &= \sqrt{2}\textstyle\sum_n g[n-2k]f[n], \end{aligned}$$

while the inverse transform takes the form

$$f[n] = \sqrt{2}\left(\sum_k h[n-2k]c_k + \sum_k g[n-2k]b_k\right).$$

Additional constraints can be imposed on $h[k]$ and $g[k]$ so that the resulting wavelet representation has some nice properties [11]. Many wavelet transforms with various filter responses $h[k]$ have been proposed. They include the Haar basis, the family of Daubechies bases [5], and the spline wavelet basis [7]. There exist many possible ways to generalize the above two-scale wavelet transform to the multiple scale case. The conventional approach is to apply the two-scale decomposition recursively only to the lowest frequency channels. Since it can be efficiently implemented via a pyramidal computational algorithm [7], we call it the pyramidal wavelet transform. For 2-D signals, one can generalize the above idea by forming the tenser product of 1-D wavelet transforms along horizontal and vertical directions.

When we transform a signal from the spatial domain to the frequency domain, some phase shift may occur. Each phase shift term corresponds to the data shifted in the spatial domain. If the linear phase holds in all directions, the data in the 2-D spatial domain may keep the same shift amount, so that the object shape in the original image is held. However, if the filter is not linear phase, the phase shift in all bands and all directions may be fractured and the object shape of the original image will be distorted accordingly. Thus, we prefer to use a linear phase filter to implement the wavelet transform. In additional to the orthogonal wavelet basis, we also examine biorthogonal wavelet transform to preserve both the FIR and linear phase properties. Various wavelet bases and techniques have been tested in our experiments to achieve a better performance. They include spline filter-III [1], variant spline filter [5], spline filter-I [1], spline filter-II [1] and 9-tap orthogonal filter [8].

We performed the embedded wavelet compression using the above filters on a set of tested image. The compression performance improvement caused by the filters is shown in Table 1, where the bit rate is 0.25 bpp.

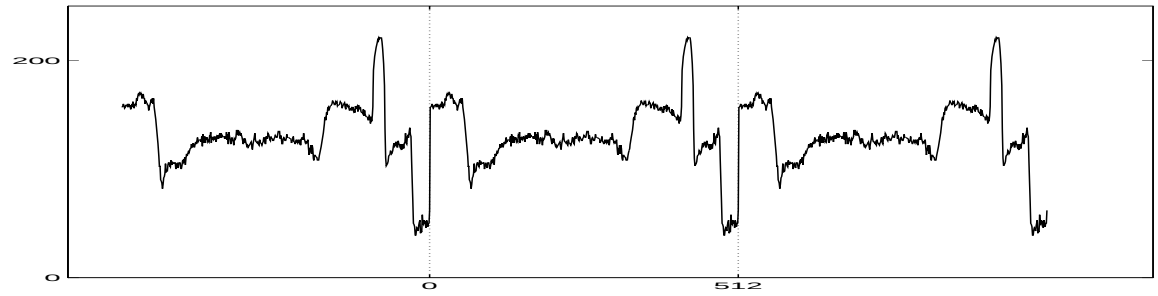
2.1 Symmetric Boundary Extension

Usually, a circular extension is used in the filtering process of the wavelet transform. The approach can be illustrated in Fig 1(a). Since the content of the image in the two ends is not necessarily equal, this convolution method causes the pseudo high frequency along the image boundary in the transform domain, which leads to famous ringing effect. The break along the image boundary also reduce the energy compaction in image boundary regions. Thus, we consider the use of symmetry boundary extension as illustrated in Fig 1(b). In the case, if the length of original data f_i for filtering is n, we symmetrily extend the data to a new sequence of length $2n$:

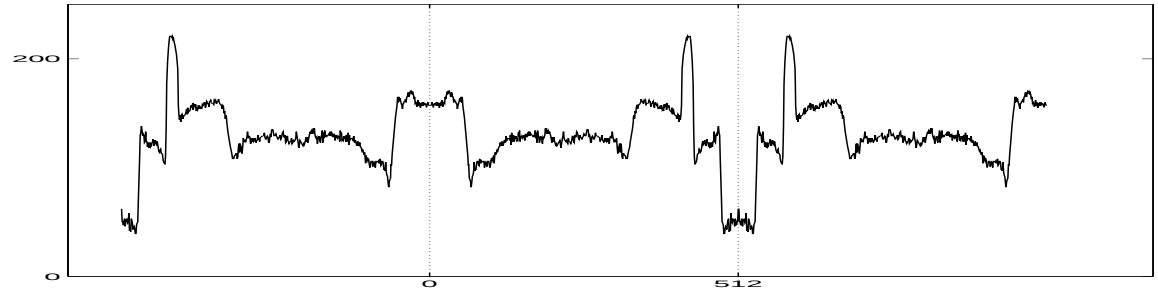
$$\tilde{f}_i = f_i, \qquad 0 \le i < n,$$

	LENA		BABOON		BOAT		CREEK	
Filter name	MSE	PSNR	MSE	PSNR	MSE	PSNR	MSE	PSNR
Spline-filter-III	32.60	33.00	421.5	21.88	75.83	29.33	221.4	24.68
Variant spline filter	33.87	32.83	446.1	21.63	80.48	29.07	231.3	24.49
Spline filter-I	29.05	33.49	440.6	21.69	74.53	29.41	212.3	24.85
Spline filter-II	24.55	34.23	392.7	22.19	66.01	29.94	206.5	24.99
9-tap orthogonal filter	26.89	33.83	402.4	22.07	69.19	29.73	210.8	24.89

Table 1: Performance comparison for various wavelet bases



(a) circular boundary extension



(b) symmetry boundary extension

Figure 1: An example of boundary extension where the data are obtained from the 30th horizontal line of the Lena image.

$$\tilde{f}_i = f_{2n-1-i}, \qquad n \le i < 2n.$$

Note that since the extended data are symmetric w.r.t the position n, the wavelet filtered low-pass coefficients $\mathbf{c}$ and high-pass coefficients $\mathbf{d}$ will also be symmetric w.r.t. the position n. Therefore, only half of $\mathbf{c}$ and $\mathbf{d}$, i.e. $n/2$ for each of them, have to be stored. The amount of coefficients to be compressed is still the same as the circular one. The two different boundary extension schemes are shown in Figs. 1(a) and (b), where the dotted line denotes to image boundary, for comparison.

2.2 Layered Byte Stuffing

For n-scale 2-D wavelet transform, the horizontal and vertical sizes of the source image are required to be a multiple of 2^n. However, many practical images do not satisfy this condition and, as a consequence, the size of the original image has to be extended by byte stuffing. We propose a layered byte stuffing technique, which is applied hierarchically to reduce the number of stuffing bytes, in this subsection.

We use a 1-D sample to illustrate this technique in Fig 2. Suppose the length n of the original 1-D data $\mathbf{f}$ is 9. In the 1st scale wavelet transform, since 9 is not a multiple of 2, one sample S is stuffed at the end of data $\mathbf{f}$. Since

Figure 2: An illustration of byte stuffing in association with wavelet transform.

S is discarded at the decoder, the content of S does not influence decoding. We can choose the stuffing byte S to be a filtered version of the original data $\mathbf{f}$ which reduce the additional energy introduced into the wavelet coefficients. After 1st scale wavelet transform, the high- and low-pass filtered coefficients (denoted, respectively, by $\mathbf{h^1}$ and $\mathbf{f^1}$) are obtained. Each of them has a length of 5. If the pyramidal wavelet transform is used, the high-pass coefficients $\mathbf{h^1}$ remain intact, the low-pass coefficients $\mathbf{f^1}$ go through the 2nd scale wavelet transform. Since length 5 is still not a multiple of 2, one additional sample is stuffed to this layer. The above byte stuffing procedure can be carried out recursively. For a 4-scale wavelet transform of data of length 9, three additional samples have to be stuffed and the total number of wavelet coefficients after the transform is 12. Compared with the scheme stuffing 7 bytes at once to construct a data of length 16 for 4-scale wavelet transform, we save 4 bytes by using the proposed layered stuffing technique.

More generally, if we apply the m-scale wavelet transform to 1-D data length n, the number of additional samples to be stuffed at scale-i is:

$$S_i(n) = \left\lceil \frac{n}{2^{i-1}} \right\rceil \mod 2$$

where $\lceil x \rceil$ means the smallest integer which is greater than or equal to x and x mod 2 denotes the remainder of the division of x by 2. The total number of additional samples stuffed in the m-scale wavelet transform is

$$S = \sum_{i=1}^{m} S_i(n). \tag{1}$$

Equation (1) can be easily generalized to the 2-D case. If we apply m-scale wavelet transform to a 2-D image of size $N_x \times N_y$, the number of stuffing samples at scale i is:

$$S_i(N_x, N_y) = S_i(N_x) \left\lceil \frac{N_y}{2^{i-1}} \right\rceil + S_i(N_y) \left\lceil \frac{N_x}{2^{i-1}} \right\rceil - S_i(N_x) \cdot S_i(N_y),$$

and the total number of stuffing samples in the m-scale pyramidal wavelet transform is

$$S = \sum_{i=1}^{m} S_i(N_x, N_y). \tag{2}$$

We use a simple example to illustrate the saving of stuffing bytes in in Table 2. Consider two images of size 513×513 and 514×514, respectively. The numbers of cumulative stuffing bytes at various layers for an m-scale wavelet transform are given in the table. For comparison, the number of total stuffing bytes required by the 8×8 DCT is also listed. We can clearly see that the total number of stuffing bytes in the wavelet transform is far less than that of DCT. Besides, most stuffing bytes occur at lower scale layers. We conclude that the wavelet transform is more flexible for irregular image coding size, which offers an additional advantage over DCT.

Image Size	Number of stuffing bytes for m-scale wavelet transform									for 8×8 DCT
	m=1	2	3	4	5	6	7	8	9	
513×513	1025	1538	1795	1924	1989	2022	2039	2048	2053	7231
514×514	0	513	770	899	964	997	1014	1023	1028	6204

Table 2: A comparison of byte stuffing between wavelet transform and DCT

3 EMBEDDED ARITHMETIC CODING OF WAVELET COEFFICIENTS

3.1 Framework of embedded coding

One major difference between embedded and traditional codings is that the quantization procedure of the embedded approach is carried out successively. The traditional image coding procedure is given in Fig 3 (a), which was established in the very beginning of image coding research. It consists of 3 separate stages: transform, quantization and entropy coding. In the 1st stage, the original image $\mathbf{X}$ is decorrelated by an invertible transform to a more compact representation $\mathbf{W}$, i.e.

$$\mathbf{W} = W(\mathbf{X}).$$

The commonly used transform includes the block DCT and multi-scale wavelet transform. Then, the transform representation $\mathbf{W}$ is quantized to reduce the representative level denoted by

$$\mathbf{Q} = Q(\mathbf{W}).$$

Finally, the quantized coefficient $\mathbf{Q}$ is transmitted through entropy coding. Since $\mathbf{Q}$ is a multi-level symbol, a relatively simple entropy coder such as the huffman coder is used. Both coding rate and distortion are determined by quantization. Even though we can adjust parameters in the quantizer for different rate-distortion tradeoff, it is in general difficult to predict the efficiency of entropy coding so that rate control is a challenging problem.

Quantization and entropy coding in traditional coding is further illustrated in Fig. 4. Consider the coding of a set of transform coefficients $W_0, W_1, \cdots, W_n$. We assume that the coefficients has been pre-normalized by T_{max} so that each coefficient is in the range of $[-1, 1]$:

$$T_{max} = \max_i \mid W_i \mid, \quad \text{and} \quad W_i^{'} = \frac{W_i}{T_{max}}.$$

In traditional image coding, we use quantization to determine the coding precision for each transform coefficients $W_0^{'}, W_1^{'}, \cdots, W_n^{'}$ as shown in Fig 4. This is equivalent to the use of quantization to determine the number of bits to be transmitted for each normalized coefficients $W_i^{'}$. The multi-level quantized value $Q(W_i)$ is further compressed by entropy coding. There is no definite relation between the quantization level and the coding rate since the efficiency of entropy coding is highly dependent on the content of the transform coefficients W_i.

An alternative approach is adopted in embedded coding. We adopt a successive quantizer and a sophisticated entropy coder (arithmetic coder) to compress the transform coefficients W_i. The successive quantization procedure can be stated as follows. For the first layer L_0, we obtain a very coarse quantization $\mathbf{Q^1}$ of $\mathbf{W}$. The quantization only identifies whether the transform coefficient W_i is over half the maximum absolute value T_{max}, i.e.

$$Q_i^1 = \begin{cases} 0, & \mid W_i \mid < T_{max}/2, \\ +1, & W_i \geq T_{max}/2, \\ -1, & W_i \leq -T_{max}/2. \end{cases}$$

As shown in Fig 4, the first quantization step basically records the most significant bit L_0 for the normalized transform coefficients $W_i^{'}$. It is called "significant identification". Note that if the magnitude of a transform coefficient W_i is greater than the threshold $T_{max}/2$, it is called significant. The corresponding reconstruction rule is:

$$\hat{Q}_i^1 = \begin{cases} 0, & Q_i^1 = 0, \\ 1.5 \cdot T_{max}/2, & Q_i^1 = 1, \\ -1.5 \cdot T_{max}/2, & Q_i^1 = -1. \end{cases}$$

Figure 4: An illustration of quantization and entropy coding.

The quantized value Q_i^1 is entropy encoded by an adaptive arithmetic coder, and the quantization residue after layer L_0 is denoted by

$$E_i^1 = W_i - \hat{Q}_i^1.$$

Then, we move to the 2nd layer of the embedded coding. We refine the quantization step size by half to quantize the residual $\mathbf{E^1}$. The quantization is denoted as $\mathbf{Q^2}$. As shown in Fig 4, the 2nd layer quantization is equivalent to identifying the 2nd most significant bit L_1 for the normalized transform coefficients W_i'. Depending on whether coefficient W_i is significant in the previous layer, the quantization and reconstruction rules can be classified into two cases: (i) significant identification quantization

$$Q_i^2 = \begin{cases} 0, & |\, E_i^1 \,| < T_{max}/4, \\ +1, & E_i^1 \geq T_{max}/4, \\ -1, & E_i^1 \leq -T_{max}/4, \end{cases} \quad \text{and} \quad \hat{Q}_i^2 = \begin{cases} 0, & Q_i^2 = 0 \\ 1.5 \cdot T_{max}/4, & Q_i^2 = 1, \\ -1.5 \cdot T_{max}/4, & Q_i^2 = -1. \end{cases} \tag{3}$$

and (ii) refinement quantization

$$Q_i^2 = \begin{cases} +1, & E_i^1 \geq 0, \\ -1, & E_i^1 < 0, \end{cases} \quad \text{and} \quad \hat{Q}_i^2 = \begin{cases} 0.5 \cdot T_{max}/4, & Q_i^2 = +1, \\ -0.5 \cdot T_{max}/4, & Q_i^2 = -1. \end{cases} \tag{4}$$

The Q_i^2 is also entropy (arithmetic) encoded. The quantization residual is now reduced to

$$E_i^2 = E_i^1 - \hat{Q}_i^2.$$

The procedure is applied recursively as shown in Fig 3(b), and layers of quantization $\mathbf{Q^1}$, $\mathbf{Q^2}$, $\cdots$, $\mathbf{Q^m}$ are stored or transmitted. The decoded coefficients $\hat{\mathbf{W}}$ is

$$\hat{W}_i = \hat{Q}_i^1 + \hat{Q}_i^2 + \cdots + \hat{Q}_i^m$$

with the quantization residue

$$E_i^m = E_i^{m-1} - \hat{Q}_i^m = W_i - (\hat{Q}_i^1 + \hat{Q}_i^2 + \cdots + \hat{Q}_i^m).$$

It is easy to see from Fig. 4 that transform coefficients W_0, W_1, $\cdots$, W_n are represented by the significant bit order in embedded coding. We record the most significant bit L_0 for each coefficient at the 1st layer, and the second most significant bit L_1 at the second layer and so on. The sign for the coefficients W_i is transmitted right after W_i is identified as significant. The rate control of embedded coding is simple, since we can terminate coding at any moments. The same is true for the decoder which can terminate decoding at any point and produce reconstructions corresponding to all lower-rate encodings. Suppose the coding is terminated at layer L_j at coefficients W_i. Then coefficients W_0, W_1, $\cdots$, W_i are refined to the jth most significant bit, while coefficients W_{i+1}, W_{i+2}, $\cdots$, W_n are refined to the (j-1)th most significant bit.

In contrast with traditional coding, the quantizer is simple in embedded coding so that a sophisticated arithmetic entropy coder can be applied to achieve high compression performance. To be more precise, since the quantizer is greatly simplified to have only two values as outputs (i.e. "0" for insignificant and "1" for significant), it is feasible for the arithmetic coder to exploit a wide variety of dependencies without making the coder too complex. Compression efficiency highly depends on whether an arithmetic coder can efficiently exploit the correlation between layers of quantized coefficients $\mathbf{Q^i}$. This will be detailed in the next section.

In addition to being applied to wavelet coding, the embedded coding approach is also applicable to DCT coefficient coding. It is in fact easy to change the structure of the arithmetic coder in JPEG to become an embedded coder. However, since that the wavelet transform can achieve higher energy compaction than the block DCT and that the pyramid-structure nature of wavelet coefficients makes arithmetic prediction far more efficient, embedded wavelet coding attracts more attention these days.

3.2 Arithmetic Coding Context

After the wavelet transform, wavelet coefficients are encoded by embedded arithmetic coding. The adaptive arithmetic coding is chosen because it provides a very efficient way to encode binary symbols and because it can effectively exploit the strong correlation of quantized wavelet coefficients in different subbands and layers. For detailed description of the arithmetic coder, we refer to [2]. We adopt two QM arithmetic coders, which are recommended by the JBIG and the JPEG standards, respectively, in this work. Both coders are based on the Bayesian estimation and derive an estimation table by taking into account the balance between rapid tracking and estimation accuracy. Generally speaking, the JBIG arithmetic coder emphasizes more on tracking the probability estimation while the JPEG arithmetic coder emphasizes more on the probability estimation quality. Experiments show that the probability characteristics of embedded quantized symbols vary greatly in the whole coding procedure, and the JBIG arithmetic coder outperforms the JPEG coder by about 0.1dB. This seems to suggest that rapid tracking of the probability estimation is more important than estimation accuracy.

To improve the efficiency of the adaptive arithmetic coder, we have to choose the arithmetic coding context properly. In the following, we denote the wavelet coefficient at position (x, y) of band j, direction s as $W_{j,s}(x, y)$, where $s \in \{L, H, V, D\}$ and the set consists of the low-low (L), low-high with horizontal high pass filtering (H), high-low with vertical high pass filtering (V), and high-high diagonal (D) filtering directions. We denote layer i quantization of $W_{j,s}(x, y)$ by $Q^i(W_{j,s}(x, y))$. The average coding rate in embedded coding is the conditional entropy of $Q^i(W_{j,s}(x, y))$

$$R(Q^i(W_{j,s}(x, y)) = H[Q^i(W_{j,s}(x, y)) \mid CTX(Q^i(W_{j,s}(x, y))],$$

where

$$H(x|Q) = \sum_Q P(Q)(-p \log_2 p - (1-p) \log_2(1-p)), \quad \text{and} \quad p = P(x = 0|Q),$$

denotes the conditional Shannon entropy. It is well known that $H(x|Q)$ decreases as p approaches 0 or 1. This implies that when x is predictable under the context Q, the coding rate will decrease. The problem of context design is therefore to determine a context under which x is more deterministic.

We can classify the context rules $CTX(Q^i(W_{j,s}(x, y)))$ into 4 categories:

$$CTX(Q^i(W_{j,s}(x, y))) = \{\mathbf{C_1}, \mathbf{C_2}, \mathbf{C_3}, \mathbf{C_4}\}.$$

The first category

$$\mathbf{C_1} = Q^{i-1}(W_{j,s}(x, y)), Q^{i-2}(W_{j,s}(x, y)), Q^{i-3}(W_{j,s}(x, y)), \cdots$$

characterizes the status of current quantized symbol $Q^i(W_{j,s}(x, y))$. As mentioned in Section 2, if $W_{j,s}(x, y)$ is identified as significant in previous layers, $Q^i(W_{j,s}(x, y))$ is simply refinement. Experiments show that refinement quantization (Equ (4)) can be closely approximated by a white uniformly distributed random variable so that it cannot be efficiently compressed by arithmetic coder. Therefore, we use a fixed probability (p=0.5) coder for refinement coding. The rate to transmit a refinement quantization is one bit. For significant identification quantization (3), context rules $\mathbf{C_2}$, $\mathbf{C_3}$ and $\mathbf{C_4}$ are used. Category

$$\mathbf{C_2} = Q^i(W_{j-1,s}(x, y)), Q^i(W_{j-2,s}(x, y)),$$

describes the hierarchical correlation context of wavelet coefficients across bands depicted as shaded circles in Fig 5. Category

$$\mathbf{C_3} = Q^i(W_{j,s}(x-1, y)), Q^i(W_{j,s}(x, y-1)), Q^{i-1}(W_{j,s}(x+1, y)), Q^{i-1}(W_{j,s}(x, y+1)),$$

characterizes the spatial correlation context depicted as empty circles in Fig 5. Finally, we use the category

$$\mathbf{C_4} = Q^i(W_{j,s-1}(x, y)),$$

to take into account the cross-directional correlation shown as the barred circle in Fig 5. We combine the three rules $\mathbf{C_2}$, $\mathbf{C_3}$, $\mathbf{C_4}$ to form one arithmetic coding context,

$$\begin{aligned} index \;=\; & Q^i(W_{j-1,s}(x, y)) \times 2^7 + Q^i(W_{j-2,s}(x, y)) \times 2^6 + Q^i(W_{j,s}(x-1, y)) \times 2^5 + Q^i(W_{j,s}(x, y-1)) \times 2^4 \\ & + Q^{i-1}(W_{j,s}(x+1, y)) \times 2^3 + Q^{i-1}(W_{j,s}(x, y+1)) \times 2^2 + Q^i(W_{j,s-1}(x, y)) \end{aligned} \tag{5}$$

Figure 5: An illustration of prediction context for embedded wavelet coefficient coding.

We have experimented many different prediction contexts of higher complexity. However, little performance improvement is observed.

3.3 Embedded Encoding Order

In this section, we propose an approach to improve the compression by reorganizing the embedded coding order. Let us consider the coding a sequence of symbols $\mathbf{S} = \{s_0, s_1, \cdots, s_n\}$ in the context of embedded wavelet coding. For optimum coding, the sequence of symbols $\mathbf{S}$ should be ordered according to the following two criteria:

1. The arithmetic prediction context of the symbol s_i should be transmitted prior to s_i.
2. The unitary MSE of the symbol sequence should be ordered in a decreasing sequence.

According to the 1st criterion, quantization layer $\mathbf{Q^i}$ should always be transmitted prior to $\mathbf{Q^{i+1}}$ and the quantized wavelet coefficient in band j should always be transmitted prior to band $j+1$, since $Q^i(W_{j,s}(x,y))$ serves as a prediction context for $Q^{i+1}(W_{j,s}(x,y))$ and $Q^i(W_{j,s}(x,y))$ serves as a prediction context for $Q^i(W_{j+1,s}(x,y))$. However, the 1st criterion does not suggest the coding order for significant identification and refinement coding.

The 2nd criterion can be elaborated as follows. Suppose b_i bits are used to transmit the symbol s_i and the MSE gain is MSE_i, we denote the unitary MSE of symbol s_i as MSE_i/b_i. We can model b_i and s_i by random variables with certain probability distribution. Then, the unitary MSE MSE_i/b_i is also a random variable. We would like to order the expectancy of unitary MSE in a decreasing sequence, i.e.

$$E[\frac{MSE_0}{b_0}] \geq E[\frac{MSE_1}{b_1}] \geq \cdots \geq E[\frac{MSE_n}{b_n}].$$

Otherwise, suppose $E[\frac{MSE_i}{b_i}] < E[\frac{MSE_j}{b_j}]$, and the prediction context of s_j does not consist of symbol s_i, we could exchange the transmission order for symbol s_i and s_j. When the bit stream is truncated between symbol s_i and s_j, a better MSE decoding can be achieved.

In the following, we explain how to improve the coding performance by separating the significant identification procedure and refinement procedure, and determine their coding order according to the 2nd criterion. Suppose we process layer i quantization with threshold T. For the previous insignificant wavelet coefficients, the original symbol lies in the range $[-2T, 2T]$, the reconstruction is 0. If the symbol remains insignificant w.r.t. threshold T, the new range after quantization will be $[-T, T]$ identified as significant w.r.t. threshold T, the new symbol range after quantization is $[T, 2T]$ or $[-2T, -T]$ depending on the sign of the symbol. The reconstruction is $1.5T$ or $-1.5T$,

(b) Refinement coding

Figure 6: A sample illustration of range change in embedded coding

respectively. The procedure of significant identification can be depicted in Fig. 6(a). Without loss of generality, let us only consider the positive significant symbol lying in the range $[T, 2T]$. The expected gain in MSE is

$$MSE_{sig} = E\|X - 0\|^2 - E\|X - 1.5T\|^2.$$

Suppose that symbol X is uniformly distribution over the range $[T, 2T]$, we can calculate the expected MSE gain as

$$MSE_{sig} = \frac{1}{T}\int_{T}^{2T} X^2 - (X - 1.5T)^2 dX = 2.25T^2, \tag{6}$$

We denote the average bits spend to identify a significant position as R_{sig}. Then, the unitary MSE for significant identification is $2.25T^2/(R_{sig} + 1)$, where the "1" in the denominator denotes the one bit required in sign coding.

For significant wavelet coefficients in previous layers, a refinement quantization is applied. The original symbol is converted from range $[2kT, 2(k+1)T]$ (with reconstruction at $(2k+1)T$) to either $[2kT, (2k+1)T]$ (with reconstruction at $(2k+0.5)T$) or $[(2k+1)T, 2(k+1)T]$ (with reconstruction at $(2k+1.5)T$). This procedure is depicted in Fig. 6(b). The expected MSE gain becomes

$$\begin{aligned} MSE_{ref} &= P\{2kT < X \le (2k+1)T\} E\left\{\|X - (2k+1)T\|^2 - \|X - (2k+0.5)T\|^2\right\} \\ &\quad + P\{(2k+1)T < X \le 2(k+1)T\} E\left\{\|X - (2k+1)T\|^2 - \|X - (2k+1.5)T\|^2\right\}. \end{aligned}$$

Suppose X is uniformly distributed over the range $[2kT, (2k+1)T]$. Then, the expected gain in MSE can be calculated as

$$MSE_{ref} = 0.25T^2. \tag{7}$$

The rate to encode a refinement bit is one bit.

Experiments show that R_{sig} is usually around 3-5 bits and $MSE_{sig}/(R_{sig} + 1)$ is greater than MSE_{ref}. Therefore, we should encode first the significant identification then the refinement quantization for each layer. The modified embedded coding order achieves a PSNR gain of about 0.1-0.2 dB.

We list a typical unitary MSE change for $R_{sig} = 4$ across layers in Table 3. We also include in table 3 a real coding sample from LENA. It is interesting to see from the table that each advancement in embedded coding from significant identification to refinement, or from refinement to the next layer of significant identification, the unitary MSE is nearly halved. Therefore, the rate-distortion curve for embedded wavelet coding can be highly approximated as a straight line in the log-log domain.

	Embedded coding layer			
	$m=0$	1	2	$3\cdots$
Threshold	$T_{max}/2$	$T_{max}/4$	$T_{max}/8$	$T_{max}/16\cdots$
Unitary MSE for significant identification	$0.1125T_{max}^2$	$0.0281T_{max}^2$	$0.0070T_{max}^2$	$0.0018T_{max}^2\cdots$
Unitary MSE for refinement coding		$0.156T_{max}^2$	$0.0039T_{max}^2$	$0.0010T_{max}^2\cdots$
	Real Sample from LENA image coding			
Threshold for LENA	697.3	348.7	174.3	$87.2\cdots$
Unitary MSE for significant identification(LENA)	3.3391	0.7132	0.1542	$0.0368\cdots$
Unitary MSE for refinement coding(LENA)		0.3750	0.1050	$0.0239\cdots$

Table 3: A sample table of unitary MSE ($R_{sig}=4$)

Rate(bpp)	PSNR(dB) of our scheme for				PSNR(dB) of LENA for		
	LENA	BABOON	BOAT	CREEK	Shapiro[10]	Said[9]	Taubman[12]
0.0625	28.34	19.75	25.35	22.12	-	-	-
0.125	31.17	20.65	27.43	23.33	30.23	-	30.96
0.25	34.23	22.19	29.94	24.99	33.17	33.67	34.12
0.5	37.28	24.14	33.08	27.09	36.28	36.84	37.25

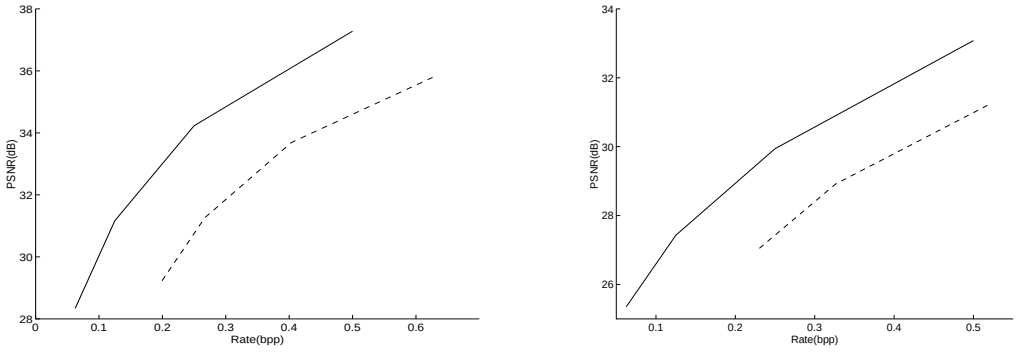
Table 4: Rate-distortion performance comparison

4 EXPERIMENTAL RESULTS

Extensive experiments were performed with LENA, BABOON, BOAT and CREEK of size 512×512 as test images. The rate-distortion performance of our proposed scheme is given in Table 4. For comparison, we also provide the embedded coding result of Shapiro[10], Said [9] and Taubman [12]. Our scheme is the best among all. We also compared our scheme with the JPEG standard and the resulting rate-distortion curves for LENA and BOAT are depicted in Fig. 7. It is clear from the figure that our scheme outperforms the JPEG standard substantially.

5 CONCLUSION AND EXTENSION

Insights into embedded zerotree wavelet (EZW) coding were provided and several improvements were examined in the paper. The contributions of our research include the choice of proper wavelet basis, symmetry boundary



(a) LENA (b) Boat

Figure 7: Comparison of the rate-distortion performance between embedded wavelet coding(solid line) and JPEG coding(dashed line)

extension, hierarchical byte stuffing, and embedded coding order adjustment. The performance of the proposed algorithm was demonstrated on test images LENA, BABOON, BOAT and CREEK. The pre- and post- processing technique for embedded wavelet coding at low bit rate and extension of the embedded coding technique to wavelet packet transformed images are under our current investigation.

6 REFERENCES

[1] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. on Image Processing*, pp. 205–230, Apr. 1992.

[2] W. B.Pennebaker and J. L. Mitchell, *JPEG still image compression standard*, Van Nostrand Reinhold, 1992.

[3] T. Chang and C.-C. J. Kuo, "Texture analysis and classification with tree structured wavelet transform," *IEEE Trans. on Image Processing*, No. 4, pp. 429–441, Oct. 1993.

[4] P.-Y. Cheng and C.-C. J. Kuo, "Feature-preserving wavelet scheme for low bit rate coding," in *1995 IS&T/SPIE Symposium on Electronic Imaging: Science and Technology*, (San Jose, California), SPIE, February 5-10 1995.

[5] I. Daubechies, *Ten lectures on wavelets*, Philadelphia: SIAM, 1992.

[6] ISO/IEC-JTC1/SC2, "Progressive bi-level image compression," ISO Standard CD 11544, ISO, Sep. 1991.

[7] S. G. Mallat, "Multifrequency channel decompositions of images and wavelet models," *IEEE Trans. on Acoustic, Speech, and Signal Processing*, pp. 2091–2110, Dec. 1989.

[8] S. G. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, No. 7, pp. 710–732, Jul. 1992.

[9] A. Said and W. Pearlman, "Image compression using the spatial-orientation tree," in *Prof. Int. Symp. Circuits, Syst.*, (Chicago, IL), pp. 279–282, May, 1993.

[10] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, No. 12, pp. 3445–3462, Dec. 1993.

[11] G. Strang, "Wavelets and dilation equations: A brief introduction," *SIAM Review*, pp. 614–627, Dec. 1989.

[12] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. on Image Processing*, No. 5, pp. 572–588, Sep. 1994.