An Embedded Image Coding Algorithm using Set Partitioning in Block Trees of Wavelet Coefficients

Chang-Mo Yang* and Yo-Sung Ho

Dept. of Information and Communications
Kwangju Institute of Science and Technology (K-JIST)

ABSTRACT

In this paper, we propose a simple but efficient wavelet-based embedded image coder that employs a new inter-band magnitude relationship in the wavelet coefficients and block trees. The proposed scheme includes multi-level dyadic wavelet decomposition, raster scanning within each subband, formation of block trees, partitioning of block trees and adaptive arithmetic entropy coding. Although the proposed scheme is simple, it produces a bitstream with a rich set of features, including SNR scalability and the embedded nature. Experimental results demonstrate that the new scheme is quite competitive to and outperforms other good image coders in the literature.

Keywords: Wavelet Transform, Image Coding, Block Tree, Block Partitioning

1. INTRODUCTION

Image compression schemes often consist of three stages: reversible transform, quantization, and error-free encoding. Block-based transforms and wavelet decompositions are commonly used to convert an image into a representation with good energy compaction. The transformed coefficients are then quantized to reduce the information and achieve the desired bit rate. The quantized coefficient is then entropy encoded in a lossless fashion.

In general, natural images display a high degree of correlation among neighboring pixels, i.e., they are of low-frequency characteristics in nature. Usually, the first stage in the image compression system involves linear transform to decorrelate the data and concentrate the energy into a few coefficients. Common image coding algorithms include block-based transforms, such as the discrete cosine transform (DCT) used in JPEG, and joint spatial-frequency based decompositions as used in subband and wavelet coding. From some viewpoint, block-based DCT codes can be interpreted as subband coding techniques. What we need is to reorder the data so that coefficients which share the same frequency band are grouped together. However, a major problem is that in block-based codes there is no interaction between pixels in different blocks which, when coupled with coarse quantization, results in blocking artifacts. Subband and wavelet techniques decompose the image into several frequency subbands. Since they are filtering-based approaches, they do not suffer from blocking artifacts and typically generate higher-quality images at low bit rates. In wavelet transforms, we use a hierarchical decomposition which recursively decomposes the low-frequency subband into $LL$, $LH$, $HL$ and $HH$ subbands. These subbands are critically subsampled so that the number of samples remains the same after the decomposition.

Early discrete wavelet transform (DWT)-based image coding algorithms have shown only marginally better performance than standard image coding algorithms based on block transforms, until the introduction of the embedded zerotree wavelet (EZW) algorithm proposed by Shapiro. The performance improvement of EZW is due to two important properties of the zerotree data structure. First, by taking advantage of the localization property of the wavelet transform and the fact that natural images tend to have decaying spectrums, zerotrees organize small wavelet coefficients into a quadtree hierarchy for compact encoding. A zerotree of insignificant wavelet coefficients can be viewed as a two-dimensional zero run which has the same effect as the run-length encoding of DCT coefficient in JPEG. Second, the successive quantization scheme used by the EZW algorithm produces very small alphabet to facilitate adaptive entropy coding. There are only three or four distinct symbols in particular segments of the zerotree code stream. From the point view of universal source coding, a source drawn from a smaller alphabet can be better modeled as a Markov chain. Since the number of states of the Markov model increases

* changmo@kiist.ac.kr; phone +82-62-970-2263; fax +82-62-970-2204; http://visual.kiist.ac.kr/; Visual Communications Lab., Dept. of Information and Communications, Kwangju Institute of Science and Technology (K-JIST), 1 Oryong-Dong Puk-Gu, Kwangju, 500-712, Korea.
exponentially along with the alphabet size, adaptive entropy coding based on Markov modeling quickly becomes impractical and inefficient as the number of source symbols increases. The zerotree techniques cleverly reduce the alphabet size needed to describe wavelet coefficients, and hence achieve attractive performance.

The work by Said and Pearlman provides even better performance than the EZW algorithm by encoding insignificant blocks of wavelet transformed coefficients with a smaller number of bits. This algorithm, called the set partitioning in hierarchical trees (SPIHT)[6], also exploits the correlation among the amplitudes of related coefficients at different scales. While the EZW algorithm scans the zerotree from subband to subband and labels the nodes as zerotree root (ZTR), isolated zero (IZ), positive significant (POS) or negative significant (NEG), the SPIHT algorithm uses the concept of set partitioning to encode the location of nonzero coefficients. The descendants of a given coefficient at a lower subband are considered as a set in the list of insignificant sets (LIS). Once a significant coefficient emerges from this set, the set is further decomposed.

Although the EZW algorithm and the SPIHT algorithm have good rate-distortion performance with low computational complexity, their memory usage is relatively high. Therefore, those algorithms are not suitable for coding of large images. A set partitioned embedded block (SPECK)[7] coder has been introduced to alleviate memory requirements of the coder. In SPECK, intra-band magnitude correlation is exploited using rectangular sets within each subband and quadrees. They use block-based quadtree partitioning where a set is partitioned into four quadrants of the same size. Although SPECK provides slightly better performance with reduced memory usage than SPIHT for some images, it does not exploit inter-band magnitude correlation existing in wavelet coefficients.

In this paper, we propose a new embedded block coder using the set partitioning in block trees (SPIBT) that can provide SNR scalability with low computational complexity and low memory usage. After the input image is decomposed into several frequency subbands, we divide each subband into small blocks and construct block trees using new inter-band magnitude correlation among wavelet coefficients. Significant coefficients in block trees are encoded by a new set partitioning method and an adaptive arithmetic coder.

The organization of this paper is as follows. In Section 2, we introduce the discrete wavelet transform of images. In Section 3, the motivation behind the proposed algorithm is described. We also suggest a new block tree structure using inter-band magnitude relationship and a set partitioning method. In Section 4, we explain the proposed coding algorithm in detail. In Section 5, performance of the proposed algorithm is analyzed and compared with that of SPIHT. Finally, Section 6 summarizes all the experimental results and advantages of the proposed scheme are also outlined.

2. DISCRETE WAVELET TRANSFORM

The wavelet transform of a signal captures the localized time-frequency information of the signal. The property of time-frequency localization greatly enhances the ability to study the behaviors of the signal as well as to change these features locally without significantly affecting the state of the signal characteristics in other regions of frequency or time. In addition, the continuous wavelet transform is closely linked to the discrete wavelet transform. This relationship allows us to speak of a “wavelet series” of any finite energy signal, obtained by “discretizing” the continuous wavelet transform. The coefficients of such a wavelet series of a signal completely capture the time-frequency characteristics of the signal, with each coefficient corresponding to a discrete time-and-frequency window. This feature provides the discrete wavelet transform with a multi-resolution property, which makes possible the study of a signal at varying resolutions.\(^3^{10}\).

![Figure 1. One Stage of Subband Decomposition](image-url)
The discrete wavelet transform used in this paper is identical to a hierarchical subband system, where the subbands are logarithmically spaced in frequency and represent the octave-band decomposition. At first, the image is divided into four subbands and critically subsampled. Each coefficient represents a spatial area corresponding to approximately a $2 \times 2$ area of the original image. The four subbands arise from separable applications of vertical and horizontal filters. The subbands labeled $LL_1$, $LH_1$, $HL_1$ and $HH_1$ represent the finest scale wavelet coefficients. To obtain the next coarser scale of wavelet coefficients, the subband $LL_1$ is further decomposed and critically subsampled. The process continues until the final scale is reached. Note that for each coarser scale, the coefficients represent a larger spatial area of the image but a narrower band of frequencies. Fig. 1 represents one stage of a subband decomposition of an image. The filters $L$ and $H$ are one-dimensional low and high pass filters, respectively.

3. MOTIVATION OF THE WORK

When we decompose the image into wavelet coefficients as shown in Fig. 2, most coefficients in high frequency bands have very small magnitudes and can be quantized to zero without any noticeable distortion. Thus, the portion of quantized zero coefficients is high at low bitrates. Therefore, we only need to send the positions of nonzero coefficients in high frequency subbands. The most straightforward way is to send a map of nonzero coefficients for each subband to the decoder. However, inter-band correlation is not exploited in this case and some redundancy still exists in the data.

![Figure 2. Two-Level Wavelet Decomposition](image)

Generally, a zerotree structure of wavelet coefficients is employed to improve the compression efficiency for the significance map. One could easily understand the inter-band correlation which used in zerotree coding schemes, while observing the tree structured decomposition of a natural image, where high frequency subbands look very much alike. Fig. 2 illustrates that even though an image is discrete wavelet transformed, occurrence of the coefficients with large values is not an independent event. As shown in Fig. 3, every coefficient at a given scale is related to a set of coefficients at the next finer scale of the same orientation with the exception of highest frequency subbands. A coefficient at the coarse scale is called as a parent, and all coefficients corresponding to the same spatial location at the next finer scale of the same orientation are called as children. A set of all coefficients at all finer scales of the same orientation and location are called as descendants. Similarly, for a given child, coefficients at coarser scales of the same orientation are called as ancestors. The lowest frequency band of the decomposition is represented by root nodes. Each parent node, except for the root of the tree, has four children in the immediately high band, which is the result of down sampling by two in each direction. Therefore, zerotree coding reduces the cost of encoding the significance map using self-similarity.

However, the memory usage of zerotree coding schemes is relatively high and they are not suitable for coding of large images. In order to reduce the memory usage of the zerotree coding scheme, we can exploit a new inter-band magnitude relationship existing in the wavelet coefficients. After the input image is decomposed into subbands, one wavelet coefficient
at a given scale is related to other coefficients corresponding to the same spatial location at the same scale of different orientation. We call the other coefficients as cousins of the given coefficient.

![Figure 3. Tree Structure of Wavelet Coefficients](image)

Therefore, except for the lowest frequency subband, every coefficient has two cousins. For the lowest frequency subband, no coefficient has cousins. We define this relationship as the coefficient-cousin relationship. Fig. 4 illustrates this inter-band magnitude relationship.

![Figure 4. Relationship among Cousins](image)

In order to construct block trees using the coefficient-cousin relationship, we divided each subband into small blocks, nominally with the dimension of 64x64 pixels. After each subband is divided into small blocks, we can construct block trees using the relationship among cousins, as shown in Fig. 5.
Once we construct block trees, we identify the significance of a tree $T$ using $c_{ij}$, the coefficient in the position $(i,j)$, and quantization step $n$ by

$$S_n(T) = \begin{cases} 1, & \text{if } 2^n \leq \max_{(i,j)\in T} |c_{ij}| < 2^{n+1} \\ 0, & \text{else} \end{cases} \quad (1)$$

where "1" means significance and "0" means insignificance with respect to $n$. Similarly, we identify the significance of a coefficient $c_{ij}$ by

$$S_n(c_{ij}) = \begin{cases} 1, & \text{if } 2^n \leq |c_{ij}| < 2^{n+1} \\ 0, & \text{else} \end{cases} \quad (2)$$

When a tree $T$ is identified to be significant with respect to $n$, it is partitioned into four small trees of the same size by the partitioning operation, as shown in Fig. 6. The significant tree is recursively split until the size of blocks $B(T)$, $i \in \{1,2,3\}$, is 4x4. When the size of blocks $B(T)$, $i \in \{1,2,3\}$, is 4x4, all coefficients in $T$ are encoded. The motivation for the proposed partitioning is to find high-energy areas quickly and encode them first.

**Figure 6. Set Partitioning in Block Trees**

4. **THE SPIBT ALGORITHM**

In the proposed algorithm, the significance information is stored in three ordered lists: the list of insignificant pixels (LIP), the list of insignificant blocks (LIB), and the list of significant pixels (LSP). For all the lists, each entry is identified by a coordinate $(i,j)$ that represents an individual pixel in LIP and LSP, and represents the starting position of the tree in LIB. The proposed algorithm consists of four coding passes: initialization, sorting pass, refinement pass, and update of the quantization step. Significance identifications in all coding passes are encoded by an adaptive arithmetic coder with a context model.
4.1. Initialization

In the first step of initialization, we divide each subband into smaller blocks and construct block trees as mentioned in Section 3. In the next step, we calculate the initial quantization step \( n \) in the transform image \( X \).

\[
  n = \left\lfloor \log_2 \left( \max_{i,j} \left| c_{i,j} \right| \right) \right\rfloor
\]  

(3)

In the third step, we determine entries of LIP, LIB, and LSP. All coefficients in the lowest subband are used as initial LIP entries and all block trees are used as initial LIB entries. LSP are set to null in this initialization pass.

4.2. Sorting Pass

In the sorting pass, we identify the significance of pixels and trees in LIP and LIB. When a pixel in LIP is identified to be significant, the sign of the pixel is also identified and the pixel is moved to LSP. Fig. 7 shows the coding procedure of LIP.

![Diagram of Sorting Pass](image)

Figure 7. Coding Procedure of LIP

Similarly, we sequentially evaluate the significance of trees in LIB. When a tree \( T \) in LIB is identified to be significant, it is partitioned according to the partitioning rule, which is described in Section 3. and removed from LIB. All partitioned small trees \( T_i, i \in \{1,2,3,4\} \), are added back to LIB. When the size of blocks \( B_T(T), i \in \{1,2,3\} \), is 4x4, significance of all pixels in the tree \( T \) is identified and the tree is removed from LIB. If a pixel is identified to be significant, the sign of the pixel is identified and the pixel is added to LSP. Otherwise, the pixel is added to LIP. Fig. 8 shows the coding procedure of LIB.

4.3. Refinement Pass

In the refinement pass, the \( n \)th most significant bit (MSB) of entries in LSP is identified with respect to \( n \). In this pass, we do not consider entries included in the last sorting pass with the same \( n \).

4.4. Update of Quantization Step

In this coding pass, we decrease the quantization step \( n \) by one, and repeat the coding step from the sorting pass.

4.5. Context Model

We use an adaptive arithmetic coder\(^1\) to encode significance identifications in each coding pass. It is well known that the adaptive arithmetic coder is computationally efficient. The adaptive arithmetic coder estimates the probability of significance of coefficients with a state machine and then arithmetically encodes it. The minimum code-length \( l \) of the sequence in bits is given by

\[ l \]
where \( p(x_i | x_{i-1}, x_{i-2}, ..., x_1) \) is a conditional probability of \( x_i \) given \( x_{i-1}, x_{i-2}, ..., x_1 \). However, \( p(x_i | x_{i-1}, x_{i-2}, ..., x_1) \) is generally unknown in practice. Therefore, we have to estimate \( p(x_i | x_{i-1}, x_{i-2}, ..., x_1) \) based on the past observations in the coding process. A set of past observations on which the probability of the current symbol is conditioned is called as the modeling context.

\[
I = -\log_2 \prod_{i=1}^{n} p(x_i | x_{i-1}, x_{i-2}, ..., x_1)
\]

(4)

![Figure 8. Coding Procedure of LIB](image)

In order to encode the significance identification of pixels, we use one parent coefficient and neighborhoods located to the north, west, south, east, northwest, and northeast of the current coefficient. For coding of signs, we use neighborhoods located to the north, west, south, and east of the current coefficient. We encode the significance identification of each tree \( T \) and the refinement bit with one fixed context.

All of the above contexts are not shared among different wavelet scales. For unavailable neighboring or parent coefficients, the corresponding context bits are set to zero.

5. EXPERIMENTAL RESULTS

Experiments are performed on two popular monochrome images, BOATS and BARBARA, of size 576×720 pixels. Each of these images is decomposed by 4-level dyadic 9/7 tap biorthogonal filters. As a performance measure, we use the peak signal-to-noise ratio (PSNR) defined by

\[
PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \ dB
\]

(5)

where MSE denotes the mean squared error between the original and reconstructed images.

Performances of the proposed and the SPIHT algorithms are compared in Fig. 8. Figs. 9 and Fig. 10 show reconstructed images of BOAT and BARBARA coded at rates 0.125, 0.25, 0.5 and 1.0 bpp using the proposed algorithm, respectively. Our experimental results demonstrate that the proposed algorithm provides good results and often outperforms SPIHT. For
BOAT, the proposed algorithm provides slightly better performance than SPIHT. However, the proposed algorithm significantly outperforms SPIHT for BARBARA that has a lot of high-frequency components.

![Graph showing PSNR (dB) vs. Rate (bpp) for BOAT and SPIHT on BARBARA](image)

**Figure 9. Performance Comparisons**

6. CONCLUSIONS

In this paper, we have proposed a new embedded image coding algorithm using a set partitioning in block trees of wavelet coefficients. Since the proposed algorithm is completely embedded, a single bitstream can be used to reconstruct images at different resolutions. The ability to adjust the compression ratio simply by truncating the bitstream makes embedded coding very attractive for various multimedia applications, such as progressive image transmission, internet browsing, scalable imaging, video database, and digital camera.

The proposed algorithm has low computational complexity because it does not contain floating-point multiplication, error calculation, and rate allocation. Although computational complexity of the proposed algorithm is quite low, the rate-distortion performance of the proposed algorithm is competitive to other good image coders in the literature. The proposed algorithm can provide SNR scalability with an embedded nature. In addition, memory usage of the proposed algorithm is lower than that of SPIHT.
(a) 0.125 bpp (PSNR = 29.47 dB) 

(b) 0.25 bpp (PSNR = 32.67 dB) 

(c) 0.5 bpp (PSNR = 36.57 dB) 

(d) 1.0 bpp (PSNR = 41.02 dB) 

Figure 10. Coding Results of BOAT
Figure 11. Coding Results of BARBARA
ACKNOWLEDGMENTS

This work was supported in part by the Korea Science and Engineering Foundation (KOSEF) through the Ultra-Fast Fiber-Optic Networks (UFON) Research Center at Kwangju Institute of Science and Technology (K-JIST), and in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK21) project.

REFERENCES