A QUADTREE-BASED EMBEDDED BLOCK CODER FOR SCALABLE IMAGE COMPRESSION

Chang-Mo Yang and Yo-Sung Ho

Kwangju Institute of Science and Technology (K-JIST)
1 Oryong-Dong Puk-Gu, Kwangju, 500-712, Korea
changmo@kjist.ac.kr, hoyo@kjist.ac.kr

***Abstract**: In this paper, we propose a simple but efficient wavelet-based embedded image coder that employs a modified quadtree partitioning. The proposed scheme includes multi-level dyadic wavelet decomposition, division of resolution layers, raster scanning within each sub-band, quadtree partitioning according to the parent-children relationship, and adaptive arithmetic entropy coding. Although the proposed scheme is simple, it produces a bitstream with a rich set of features, including resolution and SNR scalability together with the embedded nature. Experimental results demonstrate that the new scheme is quite competitive to and often outperforms other good image coders in the literature.*

***Keywords:** Wavelet, Embedded Coding, Scalability.*

1 INTRODUCTION

The embedded coder has a good property that a bitstream can be truncated at any point and still decoded to reconstruct reasonably good images.

Previous works of embedded coding include the embedded zerotree wavelet coding (EZW) [1] and the set partitioning in hierarchical trees (SPIHT) [2] algorithms. While EZW exploits the correlation among wavelet coefficients to generate zerotrees and encodes them using the significance map, SPIHT provides even better performance than EZW by encoding insignificant blocks of wavelet coefficients with a small number of bits. In SPIHT, an inter-band spatial orientation tree is used to predict the significance of wavelet coefficients efficiently.

Although both algorithms have good rate-distortion performance with low computational complexity, their memory usage is relatively high. Therefore, those algorithms are not suitable for compression of large images.

In the modified SPIHT algorithm [3], wavelet coefficients are partitioned into small tree-preserving spatial blocks that are independently coded by SPIHT. In the set partitioned embedded block (SPECK) algorithm [4], intra-band magnitude correlation is exploited using rectangular sets within each subband and quad-trees.

Although those algorithms can provide SNR scalability with the embedded nature, it is insufficient to provide resolution scalability. In this paper, we propose a new wavelet-based image coding scheme that can provide both SNR and resolution scalability with the embedded property and low computational complexity.

In the proposed Quadtree-based Embedded Block Coder (QEBC), we use rectangular sets within each subband and modified quadtree partitioning. After we divide each subband into small blocks or sets according to the resolution layer, coefficient sets at each resolution layer are encoded independently to those at higher resolution layers. For the quadtree partitioning, the parent-children relationship [1,2] is exploited. We also propose an efficient context model for adaptive arithmetic coding of significance maps.

2 MODIFIED QUADTREE PARTITIONING

In scalable image coding schemes, a bitstream can be progressively transmitted and decoded to provide different version of images in terms of either spatial resolutions (resolution scalability) or quality levels (SNR scalability).

Previous works of SNR scalable coding include EZW, SPIHT and SPECK [1,2,4]. In these algorithms, after transformed coefficients are partially ordered by their magnitudes, the ordering information is transmitted by a subset-partitioning algorithm. Refinement bits are transmitted in the bit-plane ordering. However, these algorithms are not suitable for resolution scalability because coding bits in lower resolution layers are related to those in higher resolution layers.

In order to provide resolution scalability as well as SNR scalability in a single bitstream, we divide the wavelet coefficients into multiple resolution layers L_r, as shown in Figure 1. Subbands in each resolution layer are then subdivided into blocks or sets, nominally with the dimension of 64×64, 32×32, or 16×16 pixels.

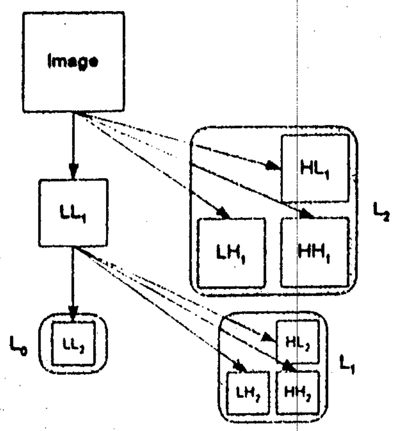

Figure 1. Multiple Resolution Layers.

After we divide each subband into small sets, the significance of a set T is identified by

$$\max_{(i,j)\in T}\{|c_{i,j}|\}\ge 2^n \quad (1)$$

where $c_{i,j}$ is a wavelet coefficient in the position (i,j) in the transformed image. We can rewrite the significance of a set T as a function of n.

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

When a set T is identified to be significant with respect to the quantization step n, it is quadtree partitioned. In SPECK, a set is partitioned into four quadrant sets of the same size by the quadtree partitioning operation. The significance test for the same quantization step n is applied to each set. If a set is identified to be significant, it is partitioned again into four quadrant sets. The significant set is recursively split until there is no significant pixel in the set. The motivation for the quadtree partitioning is to find high-energy areas quickly and code them first.

For the quadtree partitioning, we use a different method from SPECK. Instead of using the quadtree partitioning directly, we use a modified quadtree partitioning which exploits the parent-children relationship. In a hierarchical subband system, every coefficient at a given scale is related to a set of coefficients at the next finer scale of similar orientation. The coefficient at the coarse scale is called the parent, and all coefficients corresponding to the same spatial location at the next finer scale of similar orientation are called children [1,2]. Except for the lowest frequency subband, all parents have four children. For the lowest frequency subband, each parent has three children. By using this parent-children relationship, we define nine different sets, as shown in Figure 2.

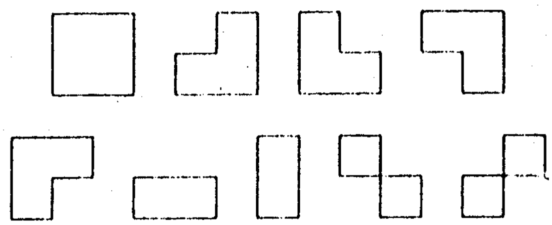

Figure 2. Nine Different Sets.

Each set is quadtree-partitioned into subsets of various shape and size, as shown in Figure 3.

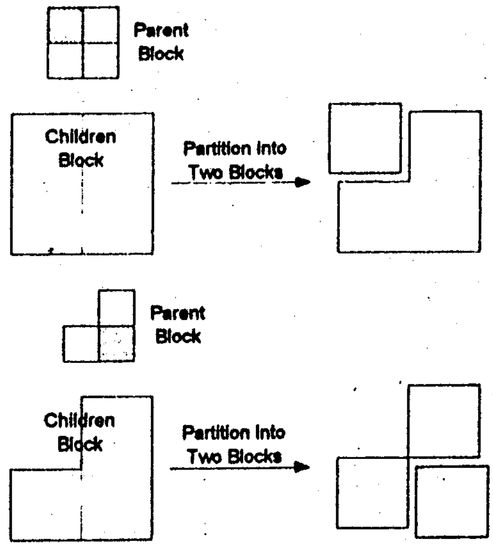

Figure 3. Modified Quadtree Partitioning.

In Figure 3, the gray region in the parent set indicates that it is significant with respect to the current or previous quantization step.

For each set of size 4×4 pixels, we encode significance identification of coefficients by an adaptive arithmetic coder. Sets in each resolution layer are encoded independently to those at higher resolution layers.

3 THE QEBC ALGORITHM

Similarly to SPIHT, the significance information is stored in three ordered lists, called list of insignificant pixels (LIP), list of insignificant blocks (LIB), and list of significant pixels (LSP). In all 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 sets in LIB. LIB includes the information of the size and the type of the set. Pixels and sets in the list are encoded according to the order of the resolution layer L_r.

QEBC consists of four coding passes: initialization, sorting pass, refinement pass, and update of quantization step.

3.1 Initialization

In the first step of initialization, we divide each subband of the transformed image into multiple resolution layers. Subbands in each resolution layer are divided again into smaller blocks or sets, as mentioned in Section 2.

In the next step, we calculate the initial quantization step n.

$$n=\left\lfloor \log_2\left(\max_{\forall(i,j)\in X} |c_{i,j}|\right)\right\rfloor \qquad (3)$$

In the third step, we determine entries of LIB[r], where r indicates the resolution layer. All partitioned sets are used as initial LIS[r] entries. LIP[r] and LSP[r] are set to null.

3.2 Sorting Pass

In the sorting pass, we identify the significance of pixels and sets in LIP[r] and LIB[r]. When a pixel in LIP[r] is identified to be significant, the sign of the pixel is also identified and the pixel is moved to LSP[r].

Similarly, we evaluate sets in LIB[r] sequentially following the LIB[r] order. When a set is identified to be significant, it is partitioned according to the partitioning rule that is described in Section 2, and removed from LIB[r]. Partitioned sets are added back to LIB[r].

When the size of a set in LIB[r] is 4×4, significance of pixels in the set is identified and the set is removed from the list. If a pixel is identified to be significant, the sign of the pixel is identified and the pixel is added to the end of LSP[r]. Otherwise, the pixel is added to the end of LIP[r]. Figure 4 explains the whole procedure of the sorting pass.

```
Sorting_Pass(r)

1. For each entry c_ij in LIP[r]
     Output S_n(c_ij)
     If S_n(c_ij)=1
        Move c_ij to LSP[r]
        Output the sign of c_ij
2. For each entry T in LIB[r]
     If the size of T is 4×4
        Output S_n(c_ij) for each c_ij in T
        Remove T
        If S_n(c_ij)=1
           Add c_ij to the end of LSP[r]
           Output the sign of c_ij
        Else
           Add c_ij to the end of LIP[r]
     Else
        Partition T according to the partitioning rule
        Remove T in the list
        Add new subbsets to the end of LIB[r]
```

Figure 4. Procedure of the Sorting Pass.

3.3 Refinement Pass

In the refinement pass, the nth most significant bit (MSB) of entries in LSP[r] is identified with respect to the quantization step n. In this pass, we do not consider entries included in the last sorting pass with the same quantization step n.

3.4 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.

3.5 Context Model

We use an adaptive arithmetic coder to encode the significance identification in each coding pass. In this paper, we define seven different contexts for encoding the significance identification of pixels and four contexts for coding of signs.

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 set T and the refinement bit with one fixed context.

All of the above contexts are not shared among different wavelet scales or different resolution layers. Coefficients in the higher resolution layer are coded independently to coefficients in the rower resolution layer. For unavailable neighboring or parent coefficients, the corresponding context bits are set to zero.

4 EXPERIMENTAL RESULTS

Performance comparisons between QEBC and other wavelet-based image coders are summarized in Table 1, Table 2 and Table 3. Experiments are performed on three popular monochrome images, LENA, GOLDHILL and BARBARA, of size 512×512 pixels. Each of these images is decomposed by 5-level dyadic 9/7 tap biorthogonal filters [5].

As a performance measure, we use the peak signal-to-noise ratio (PSNR) defined by

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \quad dB \qquad (4)$$

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

Our experimental results demonstrate that QEBC provides good results and often performs better than other wavelet-based image coding algorithms, such as EZW, SPIHT and SPECK. For LENA and GOLDHILL, SPIHT provides slightly better performance than QEBC. However, QEBC outperforms SPIHT for BARBARA that has a lot of high-frequency components. When we compare QEBC with EZW and SPECK, QEBC consistently outperforms them for all the test images.

5 CONCLUSIONS

In this paper, a new quadtree-based embedded block coder, QEBC, is introduced. Since QEBC is completely embedded, a single bitstream can be used to reconstruct images at different bit rates. 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.

QEBC has low computational complexity, because it does not contain floating-point multiplication, error calculation, and rate allocation. Although computational complexity of QEBC is quite low, the rate-distortion performance of QEBC is competitive to other good image coders in the literature. QEBC can provide resolution scalability as well as SNR scalability. In addition, the memory usage of QEBC is lower than those of SPIHT and EZW.

Acknowledgments

This work was supported in part by KOSEF through UFON, and in part by MOE through BK21.

Table 1. PSNR for LENA.

Algorithm	0.25 bpp	0.5 bpp	1.0 bpp
EZW	33.17 dB	36.28 dB	39.55 dB
SPIHT	34.11 dB	37.21 dB	40.44 dB
SPECK	34.03 dB	37.1 dB	40.25 dB
QEBC	34.07 dB	37.22 dB	40.28 dB

Table 2. PSNR for GOLDHILL.

Algorithm	0 25 bpp	0 5 bpp	1.0 bpp
EZW	30 31 dB	32 87 dB	36 20 dB
SPIHT	30 56 dB	33.13 dB	36 55 dB
SPECK	30 50 dB	33 03 dB	36.36 dB
QEBC	30.51 dB	33.07 dB	36.45 dB

Table 3. PSNR for BARBARA.

Algorithm	0.25 bpp	0.5 bpp	1.0 bpp
EZW	26.77 dB	30.53 dB	35.14 dB
SPIHT	27.58 dB	31.40 dB	36.41 dB
SPECK	27.76 dB	31.54 dB	36.49 dB
QEBC	28.06 dB	31.77 dB	36.96 dB

REFERENCES

[1] J. Shapiro, Embedded Image Coding using Z rotrees of Wavelet Coefficients, *IEEE Trans. Signal Proc.*, vol.41, no.12, pp.3445-3462, Dec'9

[2] A. Said and W. Pearlman, A New, Fast and Ef cient Image Codec based on Set Partitioning Hierarchical Trees, *IEEE Trans. on Circuits a Systems for Video Tech.*, vol. 6, no. 3, pp. 24 250, June 1996.

[3] F. Wheeler and W. Pearlman, Low-Memo Packetized SPIHT Image Compression, *Proc. Asilomar Conf. on Signals, Systems, and Compu ers*, vol. 2, pp. 1193-1197, Nov. 1999.

[4] A. Islam and W. Pearlman, An Embedded a Efficient Low-Complexity Hierarchical Ima Coder, *Visual Communication and Image Pro essing*, vol. 3653, pp. 294-305, Jan. 1999.

[5] M. Antonini, M. Barlaud, P. Mathieu, and Daubechies, Image Coding using Wavelet Tran form, *IEEE Trans. on Image Proc.*, vol. 1, no. pp. 205-220, April 1992.