EMBEDDED IMAGE CODING USING ZEROBLOCKS OF SUBBAND/WAVELET COEFFICIENTS AND CONTEXT MODELING

Shih-Ta Hsiang and John W. Woods

Center for Image Processing Research and
Electrical, Computer, and Systems Engineering Department
Rensselaer Polytechnic Institute, Troy, NY 12180-3590

1. ABSTRACT

With fast computation and excellent compression efficiency, two embedded coding techniques, zero-tree/-block coding and context modeling of the subband/wavelet coefficients, have been widely utilized for image coding applications. In this research, we present a new embedded wavelet image coding algorithm with an attempt to combine advantages of these two successful coding schemes. The experimental results show that the proposed algorithm outperforms the respected zero-tree/-block coders, SPIHT and SPECK, in compression efficiency. It is also comparable to the state-of-art JPEG 2000 test coder in PSNR performance while retaining the attractive low-complexity feature of the zeroblock coders.

2. INTRODUCTION

This work was motivated by success of two popular embedded coding techniques: zero-tree/-block coding [1, 2, 3, 4] and context modeling of the subband/wavelet coefficients [5, 6, 7]. Zero-tree/-block coding takes advantage of the nature of energy clustering of subband/wavelet coefficients in frequency and in space. This class of coders apply a hierarchical set partitioning process to split off significant coefficients (with respect to the threshold in the current bitplane coding pass), while maintaining areas of insignificant coefficients. In this way, a large region of zero pixels can be coded into one symbol. It provides an efficient method to compactly represent a group of leading zeros of subband/wavelet coefficients. The distinguished compression performances were demonstrated in [2, 3, 4]. Moreover, instead of all pixels, only a small number of elements in *lists* [2] needs to be processed in individual bitplane coding passes. Thus, processing speed for this class of coders is very fast.

High compression efficiency achieved with context modeling was presented in [5, 6, 7]. In this class of coders, *individual pixel* of the DWT bitplanes are coded using context-based arithmetic coding. With help of the context models, strong correlation of subband/wavelet coefficients within and across subbands can be effectively exploited. Although simple context modeling was also employed in [2, 3, 4], the limited context information in those algorithms were insufficient to accurately predict the status of the current node. With carefully designed context models, some algorithms [6, 7] have been able to outperform the best zero-tree/-block coders in PSNR performances. Nevertheless, unlike zero-tree/-block coders, these algorithms needed to scan all subband/wavelet coefficients at least once to finish coding of a full bitplane, with an implied higher computation cost.

To combine advantages of these two coding techniques, i.e., low computation complexity and effective exploitation of correlation of subband coefficients, we propose an Embedded image coding algorithm using ZeroBlocks of subband/wavelet coefficients and context modeling, or EZBC for ease of reference. This zeroblock coding algorithm is also based on the set partitioning technique. We adopted the adaptive quadtree splitting method introduced in SWEET [3] to separate the significant coefficients and code every block of zero pixels into one symbol. In this scheme, quadtree representations of DWT coefficients is first established for individual subbands. The bottom level of the quadtree consists of the subband/wavelet coefficients. The single node at the top tree level, or the root node, just corresponds to the maximum amplitude of the all DWT coefficients. To start with, the root is the only insignificant node to process. Each quadtree node splits into four insignificant descendent nodes of the next lower level once it tests as significant against the threshold of the current bitplane coding pass. The same splitting process is recursively applied to the individual descendent nodes until the bottom level of the quadtree is reached. In this way, we can quickly zoom in to high energy areas and regions of all zero pixels can be compactly represented.

A similar quadtree splitting scheme also appears in EBCOT [8], and SPECK. The EBCOT algorithm coded the decision of quadtree splitting as a simple binary bitstream (uncoded). The splitting process proceeded upto a minimum block size, with 16 x 16 suggested. In SPECK, the splitting

---

This work was partially supported by the National Science Foundation under grant MIP-9528312.

decision was coded by arithmetic coding with simple context models. However, our experiments indicate that strong dependence exits not only among DWT *coefficients* but also among *nodes* at individual quadtree levels within and across subbands. To the authors' knowledge, this strong dependence has not been fully exploited in any embedded coding algorithms based on set partitioning.

In EZBC, the context models were carefully designed for coding quadtree nodes at different tree levels and subbands. Therefore, it retains the properties of compactness and low complexity of the zeroblock coders, and adds context information in a effective way. Unlike the zerotree structure, each zeroblock only represents pixels from one subband. Hence, EZBC is inherently applicable to resolution scalable applications. With the interband context, dependence of subband coefficients across scales can still be effectively utilized without having zerotrees spanning several subbands.

3. DESCRIPTION OF THE ALGORITHM

The proposed algorithm is outlined as the follows:

1. **Definition**
    - $c(i,j)$: integral part of subband/wavelet coefficients at position (i,j) after proper scaling.
    - $QT_k[l](i,j)$: quadtree representation of DWT coefficients at position (i,j), band k, level l.
        - $QT_k[0](i,j) \equiv |c_k(i,j)|$
        - $QT_k[l](i,j) \equiv max\{QT_k[l-1](2i,2j), QT_k[l-1](2i,2j+1), QT_k[l-1](2i+1,2j), QT_k[l-1](2i+1,2j+1)\}$
    - $m(i,j)$: MSB of quadtree node (i,j).
    - D_k: depth of the quadtree of subband k.
    - $S_n(i,j)$: significance of node (i,j)

$$S_n(i,j) \equiv \begin{cases} 1 & \text{, if } n \leq m(i,j). \\ 0 & \text{, otherwise.} \end{cases}$$

    - $LIN_k[l]$: list of insignificant nodes from quadtree level l of subband k
    - LSP_k: list of significant pixels from subband k

2. **Initialization**

$$\begin{aligned} LIN_k[l] &= \begin{cases} \{(0,0)\} & , l = D_k. \\ \phi & \text{, otherwise.} \end{cases} \\ LSP_k &= \phi \quad , \forall k \end{aligned}$$

3. **CodeLIN**
    - for each (i,j) in $LIN_k[l]$
        - code $S_n(i,j)$
        - if $(S_n(i,j)=0)$, (i,j) remains in $LIN_k[l]$
        - else
            - if($l=0$), then code sign the bit of $c(i,j)$ and add (i,j) to LSP_k
            - else CodeJustSignifNode(i,j,l)
4. **CodeLSP**
    - for each pixel (i,j) in LSP_k, code the n-th bit of $|c(i,j)|$
5. **CodeJustSignifNode**
    - for each node (x,y) in $\{(2i,2j), (2i,2j+1), (2i+1,2j), (2i+1,2j+1)\}$ of quadtree level $l-1$
        - code $S_n(x,y)$
        - if($S_n(x,y)=0$), add (x,y) to $LIN_k[l-1]$
        - else
            - if($l=1$) code the sign bit of $c(x,y)$
            - else CodeJustSignifNode($x,y,l-1$)

EZBC begins with establishment of the quadtree representations of individual subbands. The value of each quadtree node is just equal to the maximum magnitude of the subband coefficients in its corresponding block region. In contrast with the conventional pixel-wise bitplane coding algorithm, EZBC also needs to deal with bitplanes of nodes at individual quadtree levels. Nevertheless, rather than all quadtree nodes, only elements in lists LIN and LSP need to be processed in each bitplane coding pass.

The coding procedure in EZBC is similar to SPECK which also adopted the same quadtree splitting method. The main contribution of our work is the utilization of a sophisticated context modeling scheme to efficiently encode this quadtree structure. We will introduce this scheme in the next section. Another difference from SPECK is that the lists in EZBC are all separately established for different quadtree levels and subbands. Therefore, all nodes in a list come from the same quadtree level and subband. With context models also separately built for the individual lists, different statistical characteristics of the individual quadtree levels and subbands can be more accurately modeled. The other added benefit of this modification is the algorithm can be applied for resolution-scalable coding applications once a proper bitstream parsing method is performed.

4. CONTEXT MODELING

To code the significance of the quadtree nodes, we include 8 neighbor nodes of the *same* quadtree level in the spatial context, as illustrated in Fig 1. This spatial context has been widely adopted for coding of the significance of *DWT coefficients*, e.g. [8]. However, the information across scale

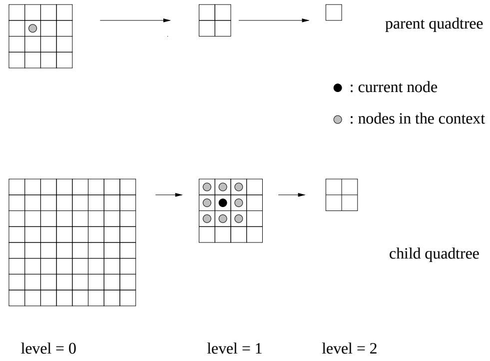

Figure 1: Context models of a quadtree node.

is given by the node of the parent subband at the *next lower* quadtree level, as shown in Fig 1. This choice is based on the fact that at the same quadtree level the dimension of the region a node corresponds to in the input image is halved in the parent subband as a result of subsampling at the transform stage. The model selection of the arithmetic coding is just based on a 9-bit string with each bit indicating the significant status of nodes in this context. To lower model cost, instead of including all context states (2^9 totally), we adopted a similar method to EBCOT for context reduction. The sign coding scheme of EBCOT is also employed in our algorithm.

5. INTER-BAND CONTEXT MODELS

The inter-band context models may increase the complexity of the algorithm. Due to the strong spatial correlation of subband coefficients, some intra-band based algorithms, e.g. [8], still demonstrate excellent compression performance. However, for coding the *quadtree nodes* of the subbbands, our experiments indicate the parent node is very helpful in predicting the current node, especially so at higher levels of a quadtree. This phenomenon can be explained as follows:

The value of a quadtree node is decided by the maximum amplitude of all DWT coefficients in the corresponding block region. This peak value can be considered as a kind of *anomaly* in statistics. Due to the nature of the energy clustering of subband coefficients in space, strong spatial correlation still exits at the lower levels of a quadtree. Nevertheless, a clustering of high energy is restricted to the local region around a peak pixel. As the block size (or quadtree level) increases, this high energy clustering in the current block will not be reflected in the neighbor nodes if the peak pixel is not close to the block boundary. However, this anomaly in space can still be observed in the parent node due to the fact that the parent/child nodes correspond to the same block region in the input image. Since the significance of the parent node is always coded earlier during every bitplane pass, we may say that the parent node provides a *look-ahead* function into the region covered by the current node.

In fact, our simulations indicate that the *interband-only* context models, including *one* node from the parent band, outperform *intraband-only* context models, including *eight* neighbor nodes, at the 3nd level of the quadtree and higher.

6. EXPERIMENTAL RESULTS

The proposed algorithm was compared with the efficient zerotree/-block coders, SPIHT and SPECK [4], and with the JPEG 2000 test coder, VM3.1A [9], in the single layer (SL) mode and in the generic scalable (GS) mode. The dyadic wavelet decomposition with Daubechies 9/7 filters was employed for subband transform in all these algorithms. The JPEG 2000 test coder is based on EBCOT and can be considered as a hybrid of the pixel- and block- wise zero coding schemes. In the SL mode, coding is optimized with respect to one target rate. The resulting bitstream is not SNR scalable.

The PSNR performances for popular test images *Lena*, *Barbara*, and *Goldhill* are shown in Tables 1 - 3. Also Tables 4 - 7 show the coding results for the JPEG 2000 test images *Cafe*, *Bike*, *Woman*, and *Aerial2* . All of those test images are in grey-scale.

Table 8 shows the average coded binary symbols per input image sample at various coding rates. Those results are actually measured by coding the four JPEG 2000 test images at various rates. It clearly indicates the computational advantage of zeroblock coders over the conventional pixel-wise embedded coders, which need to code at least one symbol (significant or not) for every pixel at each bitplane coding pass. It also shows EZBC can represent the bitplanes more compactly than the hybrid scheme EBCOT.

7. CONCLUSIONS

An new embedded image coding algorithm EZBC using zeroblock coding of the subband/wavelet coefficients and context modeling was presented. With effective exploitation of context information at the individual levels of the quadtree representation of subband coefficients, EZBC outperformed the well known zerotree coder SPIHT and a more recent zeroblock coder SPECK in compression efficiency. Our experimental results also indicate that the PSNR performance of the proposed algorithm is comparable to that of the state-of-art JPEG 2000 test coder, a hybrid of pixel- and block-wise zero coding algorithms. Nevertheless, EZBC adopts unified zeroblock coding frame work and thus possesses the desirable low-complexity feature of this class of coders.

8. REFERENCES

[1] J. M. Shapiro, “Embedded image coding using zerotrees of wavelet coefficients,” *IEEE Trans. on Signal*

bpp	0.25	0.50	1.00
J2000-SL	34.28	37.43	40.61
J2000-GS	34.16	37.29	40.48
SPIHT	34.11	37.21	40.44
SPECK	34.03	37.10	40.25
EZBC	34.35	37.47	40.62

Table 1: PSNR evaluation for Lena (512x512), in dB

bpp	0.25	0.50	1.00
J2000-SL	28.55	32.48	37.37
J2000-GS	28.40	32.29	37.11
SPIHT	27.58	31.40	36.41
SPECK	27.76	31.54	36.49
EZBC	28.25	32.15	37.28

Table 2: PSNR evaluation for Barbara (512x512), in dB

Processing, vol. 41, no. 12, pp. 3445-3462, Dec. 1993.

[2] A. Said and W. A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchal trees,” *IEEE Trans. CSVT*, no. 3, June 1996.

[3] J. Andrew, “A simple and efficient hierarchical image coder,” *ICIP’97*, vol. 3, pp 658-661, 1997.

[4] A. Islam and W. A. Pearlman, “An embedded and efficient low-complexity hierarchical image coder,” *VCIP’99*, Vol. 3653, pp 294-305, San Jose, CA, 1999.

[5] D. Taubman and A. Zakhor, “Multirate 3-D subband coding of video,” *IEEE Trans. on Image Processing*, vol. 3, pp 572-588, Sept. 1994.

[6] J. Li, S. Lei, “An embedded still image coder with rate-distortion optimization,” *IEEE Trans. on Image Processing*, vol. 8, no. 7, July 1999.

[7] X. Wu, “High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression,” *Proc. ACSSC*, Nov. 1997.

[8] D. Taubman, “EBCOT (Embedded Block Coding with Optimized Truncation): A complete reference”, ISO/IEC JTC1/SC29/WG1 N988, Sept. 1998.

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

bpp	0.25	0.50	1.00
J2000-SL	30.71	33.35	36.72
J2000-GS	30.59	33.25	36.59
SPIHT	30.56	33.13	36.55
SPECK	30.50	33.03	36.36
EZBC	30.74	33.47	36.90

Table 3: PSNR evaluation for Goldhill (512x512), in dB

bpp.	.0625	0.125	0.25	0.5	1.0	2.0
J2000-SL	19.10	20.88	23.29	27.00	32.27	39.44
J2000-GS	19.06	20.82	23.20	26.87	32.03	39.28
SPIHT	18.92	20.64	22.99	26.45	31.70	38.88
SPECK	18.93	20.61	22.87	26.31	31.47	38.75
EZBC	19.11	20.87	23.32	27.00	32.43	39.62

Table 4: PSNR evaluation for Cafe (2048x2560), in dB

bpp.	.0625	0.125	0.25	0.5	1.0	2.0
J2000-SL	23.88	26.49	29.76	33.68	38.29	44.28
J2000-GS	23.78	26.37	29.60	33.46	38.09	44.12
SPIHT	23.36	25.79	29.04	32.94	37.66	43.78
SPECK	23.31	25.59	28.84	32.69	37.33	43.10
EZBC	23.75	26.11	29.58	33.53	38.24	44.33

Table 5: PSNR evaluation for Bike (2048x2560), in dB

bpp.	.0625	0.125	0.25	0.5	1.0	2.0
J2000-SL	25.67	27.46	30.15	33.81	38.67	44.29
J2000-GS	25.63	27.39	30.04	33.70	38.49	44.17
SPIHT	25.39	27.27	29.89	33.53	38.22	43.94
SPECK	25.50	27.34	29.88	33.46	38.07	43.73
EZBC	25.71	27.54	30.31	34.00	38.82	44.48

Table 6: PSNR evaluation for Woman (2048x2560), in dB

bpp.	.0625	0.125	0.25	0.5	1.0	2.0
J2000-SL	24.71	26.59	28.66	30.71	33.39	38.31
J2000-GS	24.66	26.52	28.61	30.66	33.34	38.23
SPIHT	24.60	26.48	28.46	30.56	33.29	38.19
SPECK	24.60	26.49	28.45	30.59	33.27	38.26
EZBC	24.76	26.65	28.70	30.79	33.49	38.51

Table 7: PSNR evaluation for Aerial2 (2048x2048), in dB

bpp.	.0625	0.125	0.25	0.5	1.0	2.0
EZBC	0.07	0.15	0.29	0.58	1.15	2.24
EBCOT	0.17	0.35	0.65	1.17	2.10	3.60

Table 8: Average coded binary symbols per image sample