Classified Zerotree Wavelet Image Coding and Adaptive Packetization for Low-Bit-Rate Transport

Taekon Kim, *Member, IEEE*, Seungkeun Choi, Robert E. Van Dyck, *Member, IEEE*, and Nirmal K. Bose, *Fellow, IEEE*

***Abstract*—In this paper, a novel robust image-coding and adaptive-packetization algorithm suitable for very low bit-rate transport is suggested. This algorithm can be applied to any zerotree-based encoder, such as the embedded zerotree wavelet coder of Shapiro and set partitioning in hierarchical trees by Said and Pearlman. A very explicit segmentation and packetization method of an image bitstream, where the lowest frequency subband is separately encoded from the higher frequency subbands for unequal protection over a noisy channel, is proposed. The trees in the higher frequency subbands are split, classified, and assembled for efficient image coding and packetization according to their initial threshold and subband. The use of these classified trees enables one to make robust packets, while giving priority to some packets. In practice, each packet has a different initial threshold and can be decoded independently. In spite of additional overhead bits required for packetization, the algorithm reported is comparable to the original zerotree-based image coders at low bit rates. Additionally, simulation results show that the new method is resilient under severe packet losses.**

***Index Terms*—Adaptive packetization, error resilient, image compression, low bit rate, robust, subband coding, wavelet transform, zerotree.**

I. INTRODUCTION

A. Error-Resilient Wavelet Image Coding

THE pyramid wavelet decomposition [1], [2] is endowed with excellent energy compaction and desirable statistical properties for image compression [3]. Choosing appropriate structures in the wavelet domain for representing and quantizing the data then becomes a primary challenge in the design of an image encoder. Lewis and Knowles [4] defined a spatial orientation tree by a set of the wavelet coefficients corresponding to the same spatial location and orientation. A zerotree is then a spatial orientation tree with no significant coefficients with respect to a given value. Shapiro [5] introduced the embedded zerotree wavelet (EZW) encoder, which uses both a bit plane coding scheme and the zerotree. An improved algorithm, called set partitioning in hierarchical trees (SPIHT), was suggested by Said and Pearlman [6]. In general, the zerotree-based encoders like EZW and SPIHT have shown excellent rate-distortion performance with low computational complexity, while generating an embedded bitstream. These properties enable one to send images in a progressive manner and to encode images at any target bit rate. Xiong *et al.* [7] utilized the zerotree in an adaptive manner to obtain the best result among the zerotree-based image coders, albeit with high computational complexity.

While the mentioned zerotree-based encoders exploit the inter-subband correlation through the tree, there are some other coders that exploit either intra-subband or inter-subband correlation through structures closely related to the tree. Taubman and Zakhor [8] proposed layered zero coding for still images and video. This coder uses adaptive arithmetic coding [9] more efficiently than other methods, but requires some amount of side information. Servetto *et al.* [10] suggested a morphological representation of the wavelet data. The clustering property of significant coefficient intra- and inter- subbands was also exploited by Chai *et al.* [11]. The last two encoders emphasized the (hierarchical) morphologically significant structures among subbands and showed results comparable to zerotree-based encoders. In this paper, the proposed algorithm, classified zerotree wavelet image coding and adaptive packetization (CZWAP), which is based on a zerotree-based encoder, exploits both the tree and the clustering features simultaneously by grouping trees.

Recently, with increasing use of wireless communications and multimedia, error-resilient image coders with good compression performance are very much required. The original wavelet image coders, however, are very sensitive to bit errors and are, therefore, not good for a noisy channel. Research to improve the error resilience of wavelet image coders has been done extensively using channel coding [12]–[15], joint source-channel decoding [16], [17], robust image compression [18], and segmentation and packetization methods [19]–[21]. These methods have been applied to zerotree-based image coders more frequently than to other kinds of wavelet encoders because zerotree-based encoders have several advantages in error-resilient applications, especially those that require improved rate-distortion performance. First, they have excellent performance with very low computational complexity. Forward-error-correction (FEC) codes, such as block codes and convolutional codes, can be used. Second, the simple algorithm can be made robust to bit errors. That is, zerotree-based coders could be easily modified and implemented for robust communications. Third, the small dependency on the entropy coding

Manuscript received July 10, 2000; revised May 21, 2001. This research was supported in part by the National Science Foundation under Grant ECS-9711590. This paper was recommended by Associate Editor Z. Xiong.

T. Kim is with the Technology & Research Labs, Intel Corporation, Chandler, AZ 85226 USA (e-mail: taekon.kim@intel.com).

S. Choi is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: skchoi@psualum.com).

R. E. Van Dyck is with the Advanced Network Technologies Division, National Institute of Standards and Technology, Gaithersburg, MD 20899 USA (e-mail: vandyck@antd.nist.gov).

N. K. Bose is with the Pennsylvania State University, University Park, PA 16802 USA (e-mail: nkb1@psu.edu).

Publisher Item Identifier S 1051-8215(01)08088-0.

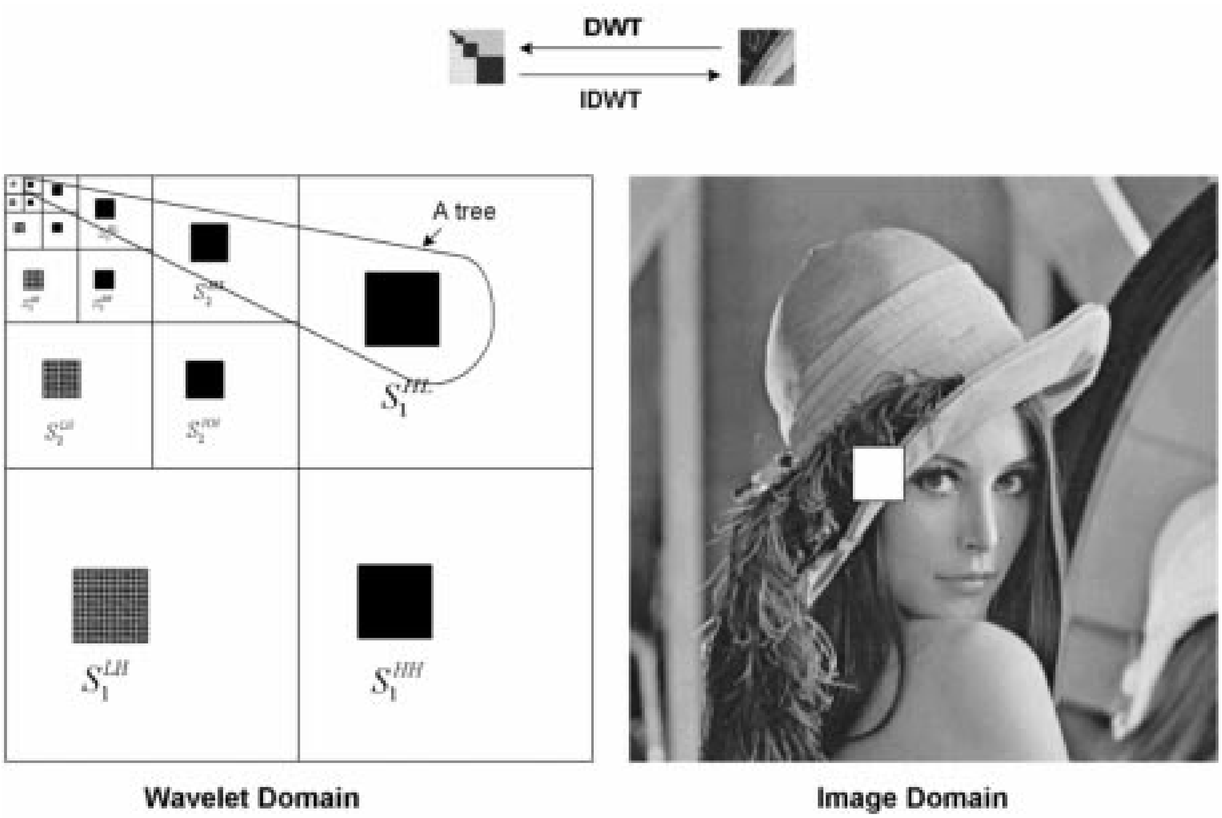


Fig. 1. Hierarchical structures in the dyadic (pyramidal) decomposed wavelet domain. A *tree* is encircled in the left figure and is the assembly of all small squares with the same font style. The union of three trees is called a *total tree*, which is composed of all small squares regardless of style. A total tree plus one coefficient in the LFS (star mark), comprises a *square tree*. A square tree corresponds to a square block in the image domain (white block in the right figure). The corresponding blocks in the wavelet and image domains are on the top of the figure. $S_l^{\mathrm{orientation}}$ represents a subband with l decomposition level and one of three orientations (LH, HL, HH). H means high-pass filtered and L means low-pass filtered.

(usually less than 0.5–1.0 dB) lets one avoid the vulnerable arithmetic coding. In image coding with segmentation and packetization, the bit-rate savings with arithmetic coding are not significant and is dependent on the packet size. In other words, image encoders, which are significantly dependent on the entropy coding, are generally not adequate for a packetization scheme.

B. Recent Research

Some of the remarkable error-resilient methods for zerotree-based encoders that are related to our work are briefly reviewed and discussed. Sherwood and Zeger [12] introduced the combination of an inner convolutional code, rate-compatible punctured convolutional (RCPC) codes [22], and an outer cyclic redundancy check (CRC) for each 200-bit packet. This work has been expanded by adding Reed–Solomon codes between packets and the method evaluated with the Gilbert–Elliott channel model [23], [24] instead of the binary symmetric channel [13]. Vass and Zhuang [14] applied Reed-Solomon codes with unequal error protection to the significance-linked connected component analysis image-coding scheme. An application of the priority encoding transmission (PET) algorithm [25] was suggested for forward error correction in packet erasure channels. Chande and Farvardin [16] proposed a progressive joint source-channel coding scheme. The bitstream from embedded source coders is segmented and packetized progressively with rate-compatible codes. All the methods mentioned above, use error-correcting codes that try to restore the information lost from either bit errors or missing packets. On the other hand, hidden Markov model-based MAP estimation, applied to the lowest and higher frequency subbands, can exploit the residual redundancy in the received data to reconstruct some of the lost data [17].

Man *et al.* [18] modified the SPIHT coder for robustness and adopted RCPC. Creusere [19] divided the wavelet coefficients into subgroups, which are encoded and transmitted independently so that bit errors only affect a single group and cannot propagate across groups. This algorithm, called robust embedded zerotree wavelet (REZW), suggested zerotree preserving wavelet coefficient partitioning, which corresponds to a square block in the original image domain (see Fig. 1), as a group structure. Rogers and Cosman [20] extended this scheme to the packetized zerotree wavelet (PZW) algorithm by using fixed-length packetization. Each packet is filled with basic structures, used in REZW, until it reaches a given size. Growing and pruning of bits at a given rate are necessary to meet the fixed packet size. Cosman *et al.* [21] introduced a hybrid of the PZW algorithm and the channel coding scheme from [13] that can use FEC to correct packet loss and bit errors. Bajic *et al.* [26] recently considered a robust image and video codec based on the concept of dispersive packetization (DP). In DP, the coefficients of a wavelet-decomposed image are packetized so that no two coefficients from a common space-wavenumber neighborhood appear in a common packet. This allows estimation of a lost coefficient from neighboring coefficients.

Most of the error-resilient algorithms reviewed above concentrated on the error correction or interpolation ability for the lost information with a noisy channel model *i.e.*, binary symmetric and Gilbert–Elliot models. However in a number of situations, such as the (wireless) Internet or an asynchronous transfer mode (ATM) packet network, a packet erasure channel model is more appropriate. So, an efficient system of error-resilient image coding and packetization is necessary. In a packet network, each packet may or may not have a different degree of priority, depending on the application. Multiple description image coders [27], [28] make the packets equally important and independent of each other. They generate multiple bitstreams rather than a single bitstream for error resilience. In this case, image quality is proportional to the number of received packets and not to the specific packets. On the other hand, priority encoding transmission [25] encodes source information into packets with priority and transmits these over a lossy packet network. The priority encoding scheme can be easily combined with a zerotree-based encoder and has abundant applications. In fact, CZWAP generates packets with unequal importance for image quality and can transmit each packet with a certain amount of priority. However, these packets can also be assumed to be equivalent. In other words, CZWAP lies between two extreme cases, namely sequential packetization for the progressive bitstream and multiple description encoding.

C. Contributions and Paper Organization

The major contribution of this work is combining the error-resilient image coding and adaptive packetization into a useful system in order to simultaneously reduce the overhead bits and the image degradation as a function of packet loss or corruption. The specific features of CZWAP are summarized as follows.

1) The zerotree-based encoders are modified to be both error resilient as well as suitable for segmentation and packetization. The lowest frequency subband (LFS) is separately encoded from the higher frequency subbands (HFS) to allow unequal protection over the noisy channel. The reason for this is that the LFS is very important for overall image quality and should be protected. The basic structure used in [19], [20] for packetization is very risky because it corresponds to a square block in the original image domain. When a packet is lost, there is no information of the corresponding square blocks.
2) Classified zerotree wavelet (CZW) image coding utilizes the side information that is necessary for packetization and does not encode trees that are not significant at a given rate or threshold.
3) CZWAP classifies each tree according to its threshold and the number of encoding bits at a given rate or threshold. So, CZWAP can split a tree into trees (more than one) that have different thresholds but specify the same tree location in the wavelet domain. This feature is very desirable in packetization.
4) The classified tree structures are assembled into groups for packetization according to their contribution to image quality. This means that the proposed coder exploits both inter- and intra- subbands correlation using both the tree structure and a clustering feature in the wavelet domain. The grouping pattern is dependent on the subband and image decomposition level.
5) In most packetization schemes, bits are segmented and packetized by a pre-determined (scan) order. Important packets should be small with a few groups and CZWAP adaptively packetizes groups of trees according to their threshold and the number of bits. CZWAP can easily control the packetization procedure.
6) In this work, CZWAP was implemented with a variable length packet method. In this case, the overhead bits are not significant and the synchronization of packets in the bitstream is almost as accurate as that of the fixed length packet method. This is made possible by using simultaneously the initial and encoding thresholds of a packet and packet length information. In fact, groups in a packet can be encoded and decoded with just the initial and encoding thresholds and even a single bit error in the bits of a packet, including the header part, can be detected without CRC bits.

There has been little attention paid to methods that perform adaptive and explicit packetization. This paper concentrates on the jointly optimal combination of the modified zerotree-based encoder and a packetization scheme in packet-based networks. Of course, there is always a tradeoff between performance and computational complexity.

This paper is organized as follows. The general zerotree-based encoders are briefly reviewed and CZW coding is described in Section II. The general packet format for the zerotree-based encoders is explained and the adaptive packetization algorithm is proposed in Section III. The various computer simulation results with CZWAP are included in Section IV. The specific characteristics and error resilience are tested. Conclusions are presented in Section V.

II. CZW Image Coding

There have been many variants of zerotree-based encoders since Shapiro introduced his algorithm in 1993 [5]. The SPIHT algorithm, developed by Said and Pearlman, shows excellent results in this class of coders. In this section, the general procedures of zerotree-based image coders and the modified zerotree-based encoder, CZW image coder, are briefly reviewed for understanding the proposed algorithm.

A. Zerotree-Based Image Coding

It is important to define and understand the hierarchical structures in the wavelet domain for the following review. The tree structure, called a tree, is a set of wavelet coefficients corresponding to the same spatial location and orientation (see Fig. 1). The assembly of three trees, which specifies the same spatial location, is called a total tree. The union of three trees (a total tree) and one coefficient in the LFS, called a square tree, corresponds to a square block in the image domain. In other words, a square tree has the complete information about the corresponding square block. It is noted that most of the zerotree-based encoders could be modified to encode each square tree, total tree or tree independently. It is very efficient to encode with square trees from a rate-distortion standpoint, because these are good for exploiting the correlation among square blocks. However, as mentioned before, it is not desirable for a noisy channel.

The zerotree-based image coders assume that if there are insignificant coefficients in the low-frequency subbands in a tree, then there are most likely insignificant coefficients in the corresponding positions in the higher frequency subbands. This is the zerotree with respect to a given threshold. Most trees can be efficiently represented by using the zerotree. However, when this assumption does not hold, considerable bits are required to specify nonzerotree structures. For example, relatively large coefficients in high-frequency subbands cost lots of bits to specify their values and locations.

Although there are some minor differences among the zerotree-based image coders, their encoding procedures can be summarized as consisting of three categories of operations: 1) the significance map pass; 2) the zerotree map pass; and 3) the refinement pass. In the significance map pass, the significance function, with respect to a given threshold, is applied to each wavelet coefficient using a predefined scanning order. The two possible results for each coefficient are significant (1 symbol) or insignificant (0 symbol). This is a form of simple binary quantization. Usually, the initial threshold T_0 is given by the following:

$$T_0 = 2^{\lfloor \log_2(\max_{i,j} |c(i.j)|) \rfloor} \quad (1)$$

where $c(i,j)$ is the wavelet coefficient at location (i,j) and $\lfloor x \rfloor$ denotes the largest integer less than or equal to x. In the next pass, the threshold is generally decreased to $T_0/2$. In the zerotree map pass, the zerotree function, which also has two possible outputs with respect to a given threshold, is applied to the trees. If there are no significant coefficients in a tree, the zerotree function outputs the insignificant symbol. Otherwise, this function outputs the significant symbol and the positions of the significant coefficients in the tree should be specified by an appropriate method at a given threshold. The choice of the specifying method determines the computational efficiency and rate-distortion performance of the particular zerotree-based encoder. In fact, the SPIHT coder improves its performance compared to most other zerotree-based encoders by applying a more sophisticated tree set in the zerotree map pass. In the refinement pass, each coefficient that turned out to be significant in the zerotree map pass, approaches its exact value. One bit is allocated for each coefficient. It is noted that the refinement pass is applied to the coefficients that are significant with respect to the former thresholds, not those that are significant with respect to the given threshold. The algorithm that creates the bitstream in an embedded and progressive manner can be terminated at any time.

The zerotree-based encoders show good rate-distortion performance with very low computational complexity. However, their performances are greatly dependent on several factors, such as the set of wavelets, the normalization unit in the wavelet transform, a scan order, and the first threshold T_0 [29]. Among these, the normalization unit, which multiplies successively the coefficients in the same decomposition level, might cause significant loss (about 3–5 dB). The first threshold may reduce the performance up to 1 dB. The number of decomposition levels can also make some differences in image compression performance. In a 512×512 image case, the six-level decomposition shows good results.

B. CZW Image Coding

In error-resilient zerotree-based image coding, it is very important to separate the LFS coding from the HFS. The coefficients in the LFS play an essential role in image quality in terms of both human visual system (HVS) and the distortion (PSNR) value. However, most of the zerotree-based image coders use the correlation between the coefficients in the LFS and trees in the HFS. So, elimination of this correlation from the encoding procedure results in from 0.5 to 1.0 dB loss at the same rate. As reviewed before, it is also crucial to split the total tree into three trees for image quality in error-resilient applications. When a packet that includes total trees is lost, one does not have any information about the corresponding square blocks in the image domain except a coefficient in the LFS. So, a new image-coding scheme that encodes the LFS and the HFS separately and gives almost the same performance as the original zerotree-based encoders is proposed.

One of the most important advantages of using the hierarchical structures (square tree, total tree and tree) in encoding images is that each tree can be encoded and decoded independently. This allows classification of trees with respect to their number of bits for a given bit rate and their initial thresholds, which come from the maximum coefficient in each tree. In most zerotree-based encoders, the coefficients in all trees are scanned in a predetermined order and some bits are assigned to specify zerotree structures. At very low encoding bit rates, considerable parts of an image are assigned zerotree symbols (see Fig. 2) and a significant number of bits could be saved by classifying trees by their initial thresholds. In fact, recent research on the patterns of wavelet coefficients in a multi-resolution representation showed that the contribution to image quality of each tree is quite different and significant trees tend to cluster [10], [11]. There is also close correlation between the initial threshold and the number of bits in a tree for a given bit rate. The more bits a tree includes, the higher the initial threshold it has. In fact, there are some trees that have low initial thresholds and few encoding bits for a given rate. So, the classification of the trees by their initial thresholds and encoding bits enables one to predict their importance for image quality. The initial threshold information of the trees is necessary for packets as header information and this classification of trees works well with packetization. In a sense, the classification scheme is a kind of joint source-channel coding. Furthermore, this CZW image coding is comparable to the original zerotree-based encoder in spite of not using the correlation between the coefficients in the LFS and tree structures in the HFS. It is also noted that the classified zerotree method leads to an adaptive packetization method. A tree can be split into more trees using the classification. This topic is discussed in detail in Section III.

In summary, the CZW coder encodes the coefficients in the LFS separately and identifies the initial threshold for each

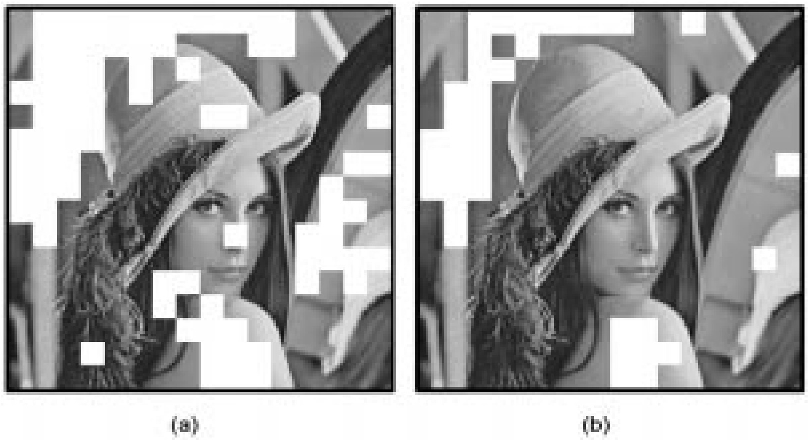


Fig. 2. Encoded part of Lena image at low bit rate at: (a) encoding rate 0.09 and (b) 0.19 bpp. As we have reviewed, a square block corresponds to three trees and one coefficient in the LFS. When more than two of the three trees are not encoded in the wavelet domain, then the corresponding square in the image domain is whitened. These squares in the image domain just have zerotree information at a given bit rate and will not be encoded in the proposed algorithm.

tree and sends this as side information. The trees whose initial thresholds are less than a given one, are not encoded. At low bit rates, a number of trees are not encoded and this saving compensates for the elimination of the LFS from the HFS.

III. Adaptive-Packetization Algorithm

In a number of applications, such as wireless communications and multimedia, segmentation and packetization of the encoded bitstream would be one of the most efficient ways to reduce the propagation of channel errors. However, little attention has been paid to methods that perform adaptive and explicit packetization in image and video coding. Subjective tests have shown that adaptive packetization with concealment of lost packets (by using similar adjacent signal segments) can reduce significantly the impact of isolated packet losses to speech [30]. In practice, an efficient packetization algorithm can greatly reduce the image degradation and encoding bit rate in a noisy channel. In this section, variable- and fixed-length packet formats are reviewed and an adaptive-packetization algorithm is discussed.

A. Packet Format for the Zerotree-Based Encoders

There are two types of packetization according to their packet size form: fixed- and variable-length. Each type has its own advantages and disadvantages, depending on the application. A fixed-length method is more convenient and robust than a variable-length one from a decoder's point of view. However, an encoder must either fill packets with zero values (null padding), providing the higher resolution information than required at a given rate, or give less information than expected. With a variable-length method, an encoder can make packets without any kind of padding, while spending more bits in the header part to specify the length of each packet. In this research, a variable-length method is adopted for the following reasons. First, it is more efficient than a fixed-length one in terms of rate-distortion at low bit rates. The null-padding or the higher resolution information will cause heavy overhead with at most a slight image quality improvement. Second, a priority on the specific tree structures can be imposed by using the packet length. For example, smaller packets of the trees that are more important for image quality can be made. Longer packets are more likely to be corrupted over a noisy channel. It is noted that there is a tradeoff between the length of packets and robustness to packet errors. In fact, segmentation and packetization with priority create several unequal bitstreams in parallel rather than a single one. Third, the number of bits for a tree in the zerotree-based encoder is quite different. So, a fixed-length packet might cause severe overhead for some trees. This usually occurs when the number of bits for a tree is much larger than a given packet size. The preceding reasons do not, however, imply that the proposed algorithm works well only with variable-length packetization. A slight modification of the algorithm enables one to use fixed-length packetization. Although variable-length packetization is adopted here, our algorithm makes the length of each packet closely converge to the predetermined length.

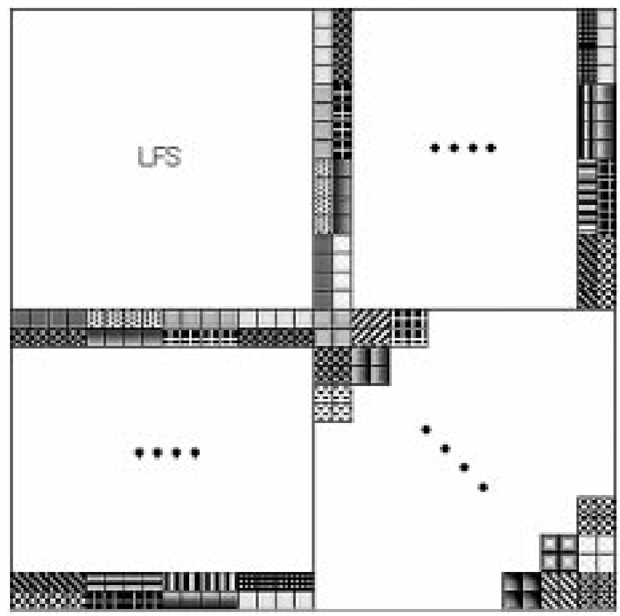


Fig. 3. Group pattern of trees. This is an example of grouping in a five-level decomposition. A group contains four trees, which have the same style blocks in the figure. There are three kinds of groupings according to orientation, exploiting the correlation vertically in the HL, horizontally in the LH, and locally in the HH.

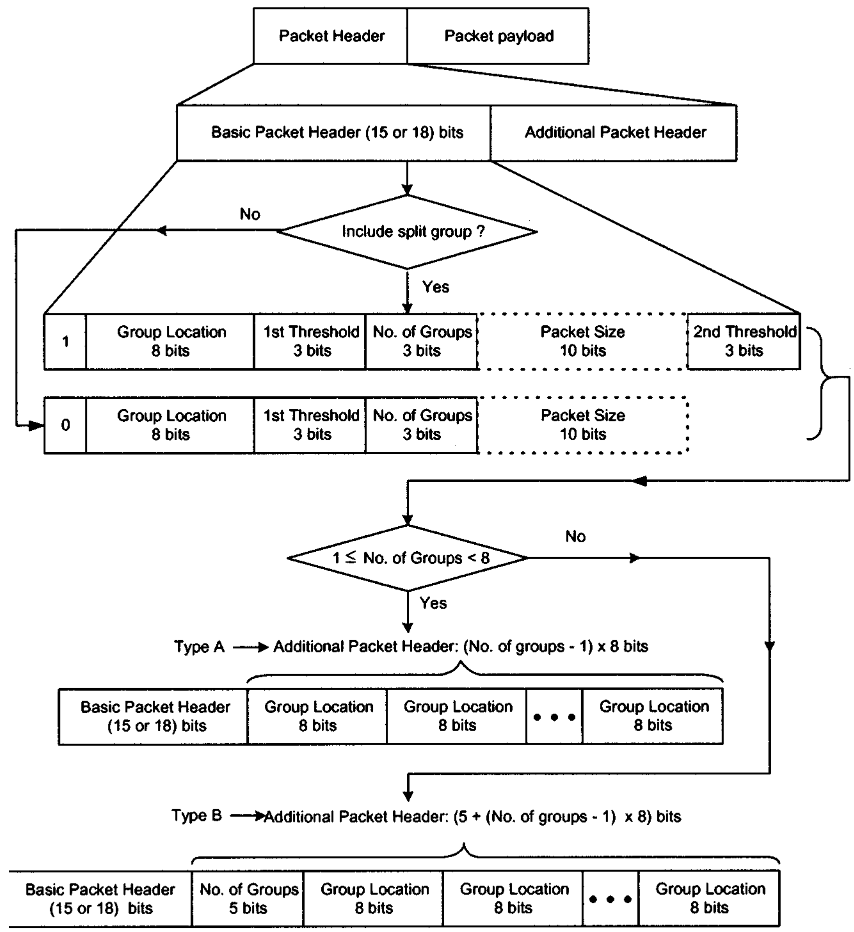


Fig. 4. Diagram of packet header generation. When a packet includes a split group, then the symbol "1" is assigned as the first bit and the last threshold (second threshold) of the group (3 bits) is necessary. So, the basic packet header is 15 or 18 bits. Packet size information (10 bits), a dotted square in the figure, is optional and will be used for synchronizing each packet in the buffer. Every group in a packet has the same initial threshold (first threshold), and we can put groups into a packet by adding group locations (8 bits/group) to the basic packet header. If a packet includes more than seven groups, then it is classified as "type B" and 5 bits are assigned for the number of groups. If not, it is classified as type A.

Most packetization schemes use a scan order such as a raster, Morton, or Peano scan and put trees in a packet until it is filled [27]. This predetermined scan order can save bits in the header part of a packet. Bits for the position of the starting tree and the number of trees in the present packet are enough to provide the packet with independent decoding. However, these kinds of packetization methods are highly image dependent and send unnecessary information at a given rate. The algorithm proposed addresses these problems and suggests an efficient solution. Rather than using a scan order, the new algorithm uses the classified trees as the units of packetization. In this case, one needs to specify the position of each tree in the present packet and this action requires a number of bits as side information. When there are many classified trees, a large number of bits are required; this can then cause serious problems. However, as reviewed before, there are some similarities between the adjacent trees and one can exploit these morphological characteristics in the wavelet domain.

For example, if a grouping of four trees is used as a basic unit for packetization, then two bits/address are saved and there is a four-fold reduction in the number of addresses. By using a grouping of these classified trees, one not only saves a considerable number of bits but also exploits the close correlation among trees. Fig. 3 shows the pattern of the grouping of trees that have the lowest coefficient in the lowest level ($S_5^{LH}, S_5^{HL}, S_5^{HH}$), except the LFS, in the five-level image decomposition case. There are three kinds of groupings according to the orientation. There is close correlation: 1) vertically in the HL; 2) horizontally in the LH; and 3) locally in the HH. As one will see in the experimental results, the grouping methods work well with the classified zerotree image coding, i.e., the initial thresholds in a group are more likely to be very similar. It is noted that CZWAP can work with any level of image decomposition and the grouping scheme should be changed according to the decomposition level. So, both the tree and the group are used as basic units of packetization from now on.

Fig. 4 shows the layout of the different packet types and the procedure for header generation in detail. In the proposed algorithm, one specifies the number of groups (3 bits), the positions (8 bits), and the initial thresholds (3 bits) of groups and the packet size (10 bits) in the packet header. There are two types of packets, based on whether or not the packet contains a split group (1 bit). Also, there is at least one group in a packet, so the basic packet header will be 25 bits. Additional groups in a packet can be represented with 8 bits (the position of a group) in the additional packet header part. The binary symbol 000 is reserved to specify that there are more than seven groups in the

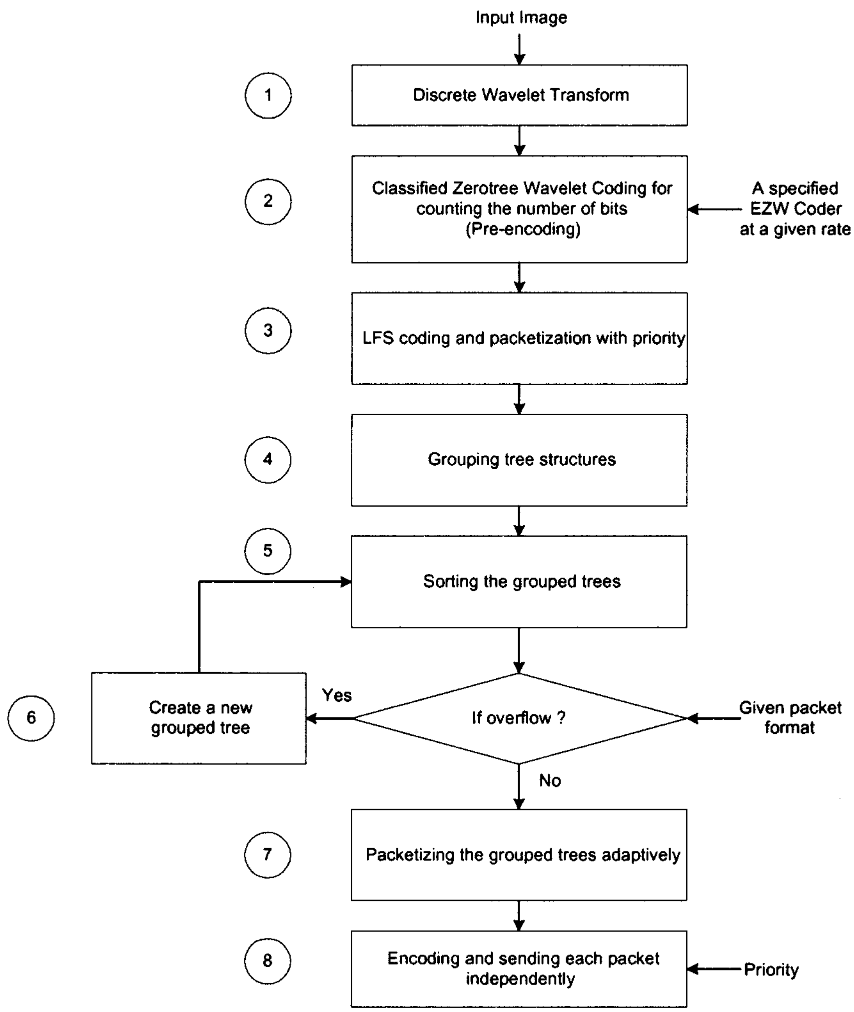


Fig. 5. Flowchart of adaptive packetization algorithm.

packet. In this case, 5 bits are given for the number of groups. All of the groups in a packet have the same initial threshold and so extra bits for these other thresholds are not necessary.

B. Adaptive Packetization with Priority

One of the most difficult problems in segmentation and packetization of the bitstream from the zerotree-based encoders, is that the distribution range of the number of bits for each tree is too wide. For example, the number of bits for a tree can be a couple of hundred bits or a few bits at 0.1 bpp for the Lena image. As the encoding rate goes up, the distribution range increases. This fact can cause some significant problems and might reduce the performance with a packetization scheme in three ways:

1) When the number of bits for a tree is larger than the packet size, a certain number of bits should be used to specify that the tree lies in more than one packet and which packet is first, second and so on. In these cases, the tree cannot be decoded from the second or third packet in the absence of the first.
2) At low bit rates, there are many trees that have just zerotree information. In fact, this kind of tree is not necessary for decoding images when we consider segmentation and packetization of the encoded bitstream.
3) The contribution to the image quality of each tree is quite different and it is approximately proportional to the number of encoding bits at a given rate.

Unequal protection for the trees over a noisy channel would be preferred in some applications. However, with other packetization schemes, it is very hard to impose a priority on the trees. All of these problems can cause significant inefficiency with the use of packetization and they can be solved with the new packetization algorithm.

Fig. 5. shows the overall algorithm from the discrete wavelet transform of an input image to encoding and sending each packet. Because CZW coding and grouping of the trees have been reviewed, steps 5, 6, 7, and 8 will be discussed in detail. There are a number of encoding, packetization and special protection schemes for the coefficients in the LFS, which are discussed in Section IV. After step 4, groups can be sorted, first by their initial thresholds and then by the number of encoding bits (see Fig. 6).

The sorting of groups enables one to fill a packet with groups that have the same initial threshold and adaptively to make packets within these groups. So, each packet has only one initial threshold and this fact gives a chance to impose appropriate priority if necessary.

(a)

Initial Group Threshold	Number of Bits	Index of Group Location
T	1086	47
T	890	48
T	462	4
T	456	8
T	286	61
T	255	64
T	179	7
T	110	191
T/2	1223	19
T/2	1168	23
T/2	675	87
T/2	645	14
T/2	407	164
⋮	⋮	⋮
T/32	87	9
T/32	62	146
T/32	31	93
T/32	23	138
T/32	22	131
T/32	20	21
T/32	20	69
T/32	16	73
T/32	15	175
T/32	13	108
T/32	12	77
T/32	7	89

(b)

Initial Group Threshold	Number of bits	Index of Group Location
T	475	48
T	479	47
T	462	4
T	456	8
T	286	61
T	255	64
T	179	7
T	110	191
T/2	675	87
T/2	645	14
T/2	516	19
T/2	456	23
T/2	407	164
⋮	⋮	⋮
T/32	713	19
T/32	699	23
T/32	613	47
T/32	415	48
⋮	⋮	⋮
T/32	87	9
T/32	62	146
T/32	31	93
T/32	23	138
T/32	22	131
T/32	20	21
T/32	20	69
T/32	16	73
T/32	15	175
T/32	13	108
T/32	12	77
T/32	7	89

(c)

Initial Group Threshold	Number of bits	Index of Group Location
T	475	48
T	473	47
T	255	64
T	462	4
T	286	61
T	456	8
T	179	7
T	110	191
T/2	675	87
⋮	⋮	⋮
T/32	713	19
T/32	698	15
⋮	⋮	⋮
T/32	415	48
T/32	87	9
T/32	62	146
T/32	31	93
T/32	23	138
T/32	22	131
T/32	20	21
T/32	20	69
T/32	16	73
T/32	15	175
T/32	13	108
T/32	12	77
T/32	7	89

Fig. 6. Example of sorting, splitting and packetizing procedures of groups. (a) Sorting groups: each group is first sorted by its initial threshold and by the number of encoding bits. (b) Creating new groups and re-sorting: if a group is larger than a given packet size, it is split into new groups. In this example, we use 768 bits (2 × 48 B). The old group and the new groups are specified with the same style in (a) and (b). These are also recognized by the same index of group location in (b). All groups are re-sorted. (c) Packetizing groups: groups with the same initial threshold are adaptively packetized.

After the sorting process with a given packet format, one can check if a group overflows the size of the packet or not. If it overflows, we generate a new group with a smaller initial threshold than the original one. In other words, a group is split into multiple groups that have the same location but different initial thresholds. This is a beautiful feature in most of the zerotree-based encoders. In practice, one can split one large group into two or more groups with a few extra bits. For example, let G be a group whose initial threshold is T_1 and encoding threshold is T_n where $T_1 > T_2 > \cdots > T_n$. The group G is represented by $G\{T_1, \ldots, T_n\}$. When a group G is split into m subgroups $G_1, G_2, \ldots, G_m$, this process can be expressed as

$$G\{T_1,\ldots,T_n\} = G_1\{T_{11},\ldots,T_{1l_1}\} \cup \cdots \cup G_i\{T_{il},\ldots,T_{il_i}\} \cup \cdots \cup G_m\{T_{m1},\ldots,T_{ml_m}\} \quad (2)$$

$$n = l_1 + l_2 + \cdots + l_m \quad (3)$$

$$T_{11} > \cdots > T_{1l_1} > T_{21} > \cdots > T_{ml_m} \quad (4)$$

where $\cup$ means independent decoding and summation. T_{il_i} is called the last threshold of the packet G_i.

Then, all of the groups are re-sorted and packetized in an adaptive manner within the groups with the same initial threshold. A larger group has the priority to be alone, but it can also share a packet with small ones. In this process, each packet size approaches the given one and each packet becomes very similar in terms of the contribution to image quality. The same procedure applies to the next threshold that is usually half of the former threshold and the process continues until a given bit rate is met. Groups with an initial threshold, which is smaller than a given threshold, are not even encoded eventually. One encodes and sends individual packets independently, with priority if necessary. It is noted that adaptive packetization is implied by two features of the algorithm: adaptive group splitting with a given packet format and adaptive packetization of groups with the same initial threshold.

IV. Experimental Results

In this set of experiments, the SPIHT algorithm [6] is chosen as the zerotree-based image coder and the CZWAP developed here is then applied without arithmetic coding. This algorithm (CZWAP) shows excellent performance without iterative computations and even without vulnerable arithmetic coding, among the class of zerotree-based encoders. In fact, arithmetic coding, in general, can improve the rate-distortion value by about 1.0 dB, but just by about 0.5 dB in SPIHT. This property has two desirable advantages when considering packetization.

1) It is possible to avoid arithmetic coding by tolerating an additional 0.5 dB degradation in PSNR. In a noisy channel, this entropy coding could cause some difficulties in decoding [17], [18].
2) Segmentation and packetization can significantly reduce the performance of adaptive arithmetic coding because

TABLE I
COMPARISON IN PSNR OF RATE–DISTORTION PERFORMANCES OF ORIGINAL SPIHT AND CZW IMAGE CODING WITH TEST IMAGES. (a) LENA. (b) BARBARA. (c) GOLDHILL. (d) BABOON

	0.0075 bpp	0.0154 bpp	0.0380 bpp	0.0879 bpp	0.1898 bpp
CZW	21.18	23.44	26.23	29.25	32.47
SPIHT	21.72	23.67	26.31	29.27	32.48

(a)

CZW	19.30	20.97	22.75	25.82	29.83
SPIHT	19.70	21.07	22.77	25.83	29.83

(b)

	0.0061 bpp	0.0123 bpp	0.0321 bpp	0.0918 bpp	0.2474 bpp
CZW	21.31	23.10	25.06	27.38	30.14
SPIHT	21.95	23.30	25.10	27.39	30.13

(c)

CZW	19.63	20.40	21.46	23.64	27.29
SPIHT	20.08	20.54	21.49	23.65	27.30

(d)

the number of bits in a packet is not enough to exploit the probability distribution of the bits. If a zerotree-based encoder is deeply dependent on the adaptive arithmetic coding, then its performance might decrease by about 0.5–1.0 dB, even with arithmetic coding.

The biorthogonal 9/7 filter bank and 512×512 gray scale images with 8 bpp are used for the experiments. The six- and five-level decompositions are constructed by a symmetric extension at the image edges. The six-level decomposition is used in the comparisons of CZW coding versus SPIHT and the five-level one in the packetization studies. The corresponding size of the LFS is 8×8 for the six-level case and 16×16 for the five-level case. As discussed in Section II, the six-level decomposition shows the best results in compression performance. However, it is less efficient than the five-level one for a noisy channel in terms of both rate-distortion and error-resilience properties. The size of the LFS in the six-level decomposition is too small for effective unequal protection and the grouping in the five-level one works well with packetization. A packet erasure channel model without FEC is implemented for error-resilient transmission.

A. CZW Coding

One of the advantages of using the CZW algorithm is the ability to separately encode and protect the LFS coefficients from the HFS. In a six-level decomposition, there are 64 coefficients (8×8) and appropriate bits are assigned according to a given rate. For example, 7 or 8 bits show good results around 0.2 bpp; 448 or 512 bits are enough to encode the LFS. Even in a five-level decomposition, 2048 bits are sufficient. These bits can be protected using packetization and channel coding such as Reed–Solomon codes in error-resilient applications.

Table I compares the performance of the proposed CZW algorithm (using SPIHT) versus the original SPIHT algorithm. While the results at extremely low bit rates are 0.4–0.6 dB lower, the PSNR difference is almost equivalent at all the other rates. CZW identifies the initial thresholds for the trees and sends these as side information. At low bit rates, 2 or 3 bits are enough to specify these initial thresholds. Overall, either 384 or 576 bits are sent. Encoding starts with the maximum initial threshold and zerotree information for trees that have initial thresholds less than the maximum need not be sent. A significant number of trees are never encoded at low bit rates and this saving compensates for the elimination of the LFS from the HFS coefficients.

TABLE II
COMPARISON IN PSNR OF RATE-DISTORTION PERFORMANCES OF ORIGINAL SPIHT AND CZWAP WITH LENA IMAGE

	0.04 bpp	0.10 bpp	0.20 bpp
CZWAP	26.24	29.45	32.47
SPIHT (w/o packetization)	26.59	30.00	32.74

B. Adaptive Packetization

In these experiments, variable-length packetization is implemented and this method requires the packet length information in the header. The maximum packet size is chosen to be 96 B, which is twice as many as that of the ATM cell format. In this case, the packet length information needs 10 bits and the overhead increases as the encoding rate rises and more packets are generated. However, if the encoding condition is given by a threshold, then the decoder needs only the initial threshold and the encoding threshold. This means that the 10 bits for packet-length information are not necessary for independent decoding.[1] Furthermore, the use of the initial and encoding thresholds enables the decoder to detect even a single bit error in the bits of a packet including the header part, without adding CRC bits. Of course, when an encoding rate—which does not match with a threshold—is given, the packet-length information is crucial. Without this information, packets in the received bitstream cannot be synchronized. When the encoding threshold is $32\ (= 2^5)$ for the Lena image, seven groups (28 trees) are not encoded and 75 packets are generated from 195 groups with an average packet length of 679 bits. The ten groups, which are larger in size than the given packet, are further split into twenty groups. It is noted that the initial and the last thresholds are needed for the split groups. In this case, 645 extra bits can be saved by eliminating the packet length information from the header part.

Table II shows the performance results of SPIHT and CZWAP for the Lena image. CZWAP generates packets with a small number of overhead bits (about 2000 bits at approximately 0.2 bpp). This produces huge savings compared to other packetization methods. Of course, the number of overhead bits are proportional to the number of packets and more packets can resist errors better. Around 0.2 bpp, CZWAP produces 75 packets and the resulting packet stream is able to withstand packet losses of 20%–30%, as will be seen below.

C. Progressiveness and Priority

CZWAP classifies trees by their initial thresholds and makes groups according to their subbands and image decomposition

[1]This simplification is dependent on the assumption of a packet erasure channel. If one were to consider a discrete channel at the bit level, a resynchronization marker would begin each packet. (For more details, please see [17]).

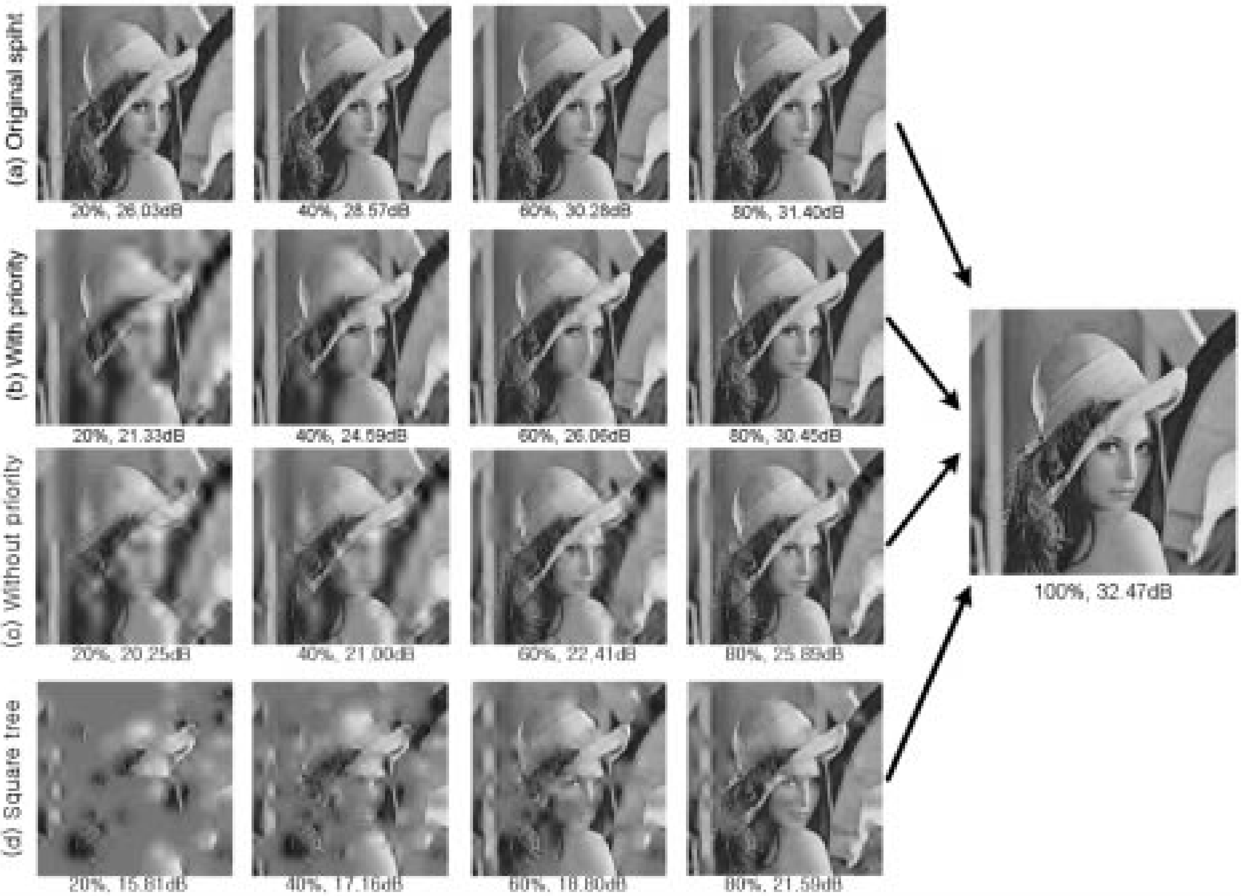


Fig. 7. Comparison of the degree of progressiveness. The encoding rates of four schemes are different but they have the same final image quality: (a) original SPIHT, 0.1926 bpp; (b) CZWAP with 100% priority for 20% of the packets, 0.2009 bpp; (c) CZWAP without priority, 0.2009 bpp; and (d) modified CZWAP (square tree) without priority, 0.2003 bpp.

TABLE III
COMPARISON OF PSNR WITH CZWAP AND PRIORITIES (100%, 50% AND 0%) OF 20% OF THE PACKETS PACKETS AND MODIFIED CZWAP WITHOUT PRIORITY AT PACKET ERASURES (0%, 10%, 20% AND 30%) OF LENA IMAGE

	Rate [bpp]	No loss	10% loss	20% loss	30% loss
100% priority	0.2009	32.47	29.02	26.88	25.43
50% priority	0.2009	32.47	28.50	26.29	24.65
No priority	0.2009	32.47	28.15	25.82	24.28
Square tree	0.2003	32.47	25.71	23.01	21.27
PZW^2 [20]	0.21	32.19	26.29	24.63	-

level. Then, groups are sorted by the initial threshold and the number of bits. It is clear that there is close correlation between the initial threshold and the number of bits for a group. A group with a high initial threshold usually requires a lot of encoding bits. If a group has a large number of bits at a given rate, then it obviously contributes to image quality a lot. So, the initial thresholds of groups and the numbers of bits can play a key role in determining the priority order of groups. In this experiment, we assume that the priority order of groups is equal to the order of sorted groups. We will demonstrate the validity of the assumption by showing the degree of the progressive transmission of groups.

The zerotree-based encoders generate the bitstream in a progressive manner. However, most of the packetization schemes get rid of some progressiveness. Until a packet is decoded entirely, other packets should be in buffer memory and cannot be decoded. If packets are filled with segments of the progressive bitstream and received in the exact order, then any progressiveness is not lost. This is an extreme case of the PET algorithm [25]. However, a packet then depends on the previous packets and perhaps even cannot be decoded without these. In other packetization methods based on the hierarchical structures and a scan order, packets are expected to share information equivalently in general. Multiple-description (MD) coding is an extreme example. It is assumed that each packet has the same amount of information. In this case, packetization will lead to entire loss of progressiveness. Of course, an efficiency compar-

(a) (b)

(c) (d)

Fig. 8. Lena image encoded CZWAP with 100% priority for 20% of the packets. (a) No packet loss, PSNR = 32.47 dB. (b) 10% packet loss, PSNR = 29.02 dB. (c) 20% packet loss, PSNR = 26.88 dB. (d) 30% packet loss, PSNR = 25.43 dB.

(a) (b)

(c) (d)

Fig. 9. Lena image encoded CZWAP with 50% priority for 20% of the packets. (a) No packet loss, PSNR = 32.47 dB. (b) 10% packet loss, PSNR = 28.51 dB. (c) 20% packet loss, PSNR = 26.29 dB. (d) 30% packet loss, PSNR = 24.65 dB.

ison of the two methods, PET and MD, is greatly dependent on the channel model and application. CZWAP generates packets with unequal importance for image quality and can transmit packets with priority. In other words, CZWAP keeps some progressiveness.

(a) (b)

(c) (d)

Fig. 10. Lena image encoded CZWAP without priority. (a) No packet loss, PSNR = 32.47 dB. (b) 10% packet loss, PSNR = 28.15 dB. (c) 20% packet loss, PSNR = 25.82 dB. (d) 30% packet loss, PSNR = 24.28 dB.

(a) (b)

(c) (d)

Fig. 11. Lena image encoded modified CZWAP (square tree) without priority. (a) No packet loss, PSNR = 32.47 dB. (b) 10% packet loss, PSNR = 25.71 dB. (c) 20% packet loss, PSNR = 23.01 dB. (d) 30% packet loss, PSNR = 21.27 dB.

Packetization schemes, based on the square tree or the total tree, are more vulnerable to errors than ones based on a tree. To demonstrate this argument, the CZWAP algorithm was slightly modified to use a square tree as a basic structure for encoding and packetization. Fig. 7. shows progressiveness of the original SPIHT, CZWAP with and without priority and square tree

CZWAP. Here, the priority order of packets is the same as the sorting one. CZWAP with priority shows remarkable progressiveness and even with only 40% reception of packets the Lena image can be identified. CZWAP without priority shows better progressiveness than square tree CZWAP due to the use of the tree. As expected, square tree CZWAP receives the information based on square blocks in the image domain. In other words, there are always undefined square blocks until all packets arrive.

D. Packet-Loss Network

The noisy channel is modeled as one with packet loss. In this model, packet erasure occurs equally for all packets except those for the LFS and packets with 100% priority (by ARQ), regardless of the packet size. 50% priority for packets means that these packets have 50% probability of loss less than other packets without priority. In a bit-error model, a larger packet is more vulnerable to errors than a smaller one. So, this assumption seems to be not fair considering the nature of transmission errors. However, as reviewed, packets generated by CZWAP compactly cluster around the average length and important packets tend to be smaller than the average. Packets for the LFS and those with perfect priority are assumed to be neither lost nor corrupted. In an ATM network, these priorities can be implemented by using the cell-loss priority [31].

Table III shows that transmission with 100% priority for 20% of the packets (selected by sorting order of packets) is the best in the 10%, 20%, and 30% packet-loss cases. As the percentage of packet loss goes up, the advantage gained by using priority increases. Yet, the simulation results show that CZWAP is still quite resilient to packet loss, even without priority, because it outperforms square tree CZWAP by 2–3 dB. One can argue that this big difference directly comes from the unequal protection for the LFS. However, the LFS coefficients in square tree CZWAP are interpolated by the adjacent coefficients and even the LFS substitution with true coefficients can improve image quality by at most 1 dB. Furthermore, the resulting images in Figs. 8–11 clearly show the difference. Table III also shows the performance of PZW [20] for 10% and 20% packet loss.

V. Conclusion

In this paper, a novel error-resilient image compression algorithm (CZWAP), in which both a zerotree-based encoder and packetization scheme are modified for performance and robustness, has been proposed. This algorithm specifies explicit and adaptive packetization procedures, which received little attention before. At low bit rates, the CZWAP algorithm is almost equivalent to the original zerotree-based encoder with respect to rate distortion performance (less than 0.2 dB), in spite of a heavy overhead for the packets.

Most of the other packetization schemes with zerotree-based encoders use a predetermined scan order and cause significant side information bits for the packets. Even so, some of them can be applied under restricted circumstances, i.e., with certain decomposition level, around any specific encoding rate, and with a few zerotree-based encoders. By jointly using an image coder and a packetization scheme, CZWAP suggests a solution for all of these problems. The experimental results are a remarkable improvement over the existing methods. The overhead bits are about 2000 bits at low bit rate (<0.25 bpp) and CZWAP can be applied to any zerotree-based encoder regardless of encoding rate and image decomposition level. The use of the tree as a basic structure, instead of the square- or total-tree, for encoding and packetization, greatly increased the robustness.

Although the variable-length packet method has been used here, the overhead bits are not significant and the synchronization of packets in the bit tream is almost as accurate as that of the fixed-length packet method. This is made possible by using simultaneously the initial and encoding thresholds of a packet and packet length information.

There are several candidate wavelet image coders for the JPEG 2000 [32] and IMT2000 [33] standards, and a few zerotree-based encoders will be adopted as standard. This is because the zerotree-based encoders have shown excellent performances with low computation, even at very low bit rates. However, these are very sensitive to noise, and without appropriate protection, they are useless in many applications. In this sense, CZWAP provides an essential algorithm for error-resilient applications and makes these zerotree-based encoders more competitive than other wavelet ones.

References

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

[2] J. W. Woods and S. D. O'Neil, "Subband coding of images," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 1278–1288, Oct. 1986.

[3] R. W. Buccigrossi and E. P. Simoncelli, "Image compression via joint statistical characterization in the wavelet domain," *IEEE Trans. Image Processing*, vol. 8, pp. 1688–1701, Dec. 1999.

[4] A. S. Lewis and G. Knowles, "Image compression using the 2-D wavelet transform," *IEEE Trans. Image Processing*, vol. 1, pp. 244–250, Feb. 1992.

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

[6] A. Said and W. A. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technology*, vol. 6, pp. 243–250, June 1996.

[7] Z. Xiong, K. Ramchandran, and M. T. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. Image Processing*, vol. 6, pp. 677–693, May 1997.

[8] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Processing*, vol. 3, pp. 572–588, Sept. 1994.

[9] I. H. Witten, M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520–540, June 1987.

[10] S. D. Servetto, K. Ramchandran, and M. T. Orchard, "Image coding based on a morhpological representation of wavelet data," *IEEE Trans. Image Processing*, vol. 8, pp. 1161–1174, Sept. 1999.

[11] B.-B. Chai, J. Vass, and X. Zhuang, "Significance-linked connected component analysis for wavelet image coding," *IEEE Trans. Image Processing*, vol. 8, pp. 774–784, June 1999.

[12] P. G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," *IEEE Signal Processing Lett.*, vol. 4, pp. 189–191, July 1997.

[13] ——, "Error protection for progressive image transmission over memoryless and fading channels," *IEEE Trans. Commun.*, vol. 46, pp. 1555–1559, Dec. 1998.

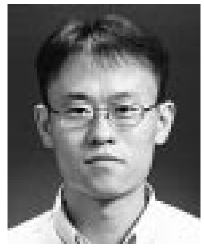
[14] J. Vass and X. Zhuang, "Robust significance-linked connected component analysis for low complexity progressive image transmission over noisy channel," in *Proc. IEEE Int. Conf. Image Processing*, Kobe, Japan, 1999.

[15] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Graceful degradation over packet erasure channels through forward error correction," in *Proc. IEEE Int. Conf. Image Processing*, Kobe, Japan, 1999.

[16] V. Chande and N. Farvardin, "Joint source-channel coding for progressive transmission of embedded source coders," in *Proc. Data Compression Conf.*, Snowbird, UT, 1998.
[17] R. E. Van Dyck, "MPEG-4 image transmission using MAP source-controlled channel decoding," *IEEE J. Select. Areas Commun.*, pp. 1087–1098, June 2000.
[18] H. Man, F. Kossentini, and M. J. T. Smith, "Robust EZW image coding for noisy channels," *IEEE Signal Processing Lett.*, vol. 4, pp. 227–229, Aug. 1997.
[19] C. D. Creusere, "A new method of robust image compression based on the embedded zerotree wavelet algorithm," *IEEE Trans. Image Processing*, vol. 6, pp. 1436–1442, Oct. 1997.
[20] J. K. Rogers and P. C. Cosman, "Robust wavelet zerotree image compression with fixed-length packetization," in *Proc. Data Compression Conf.*, Snowbird, UT, 1998, pp. 418–427.
[21] P. C. Cosman, J. K. Rogers, P. G. Sherwood, and K. Zeger, "Combined forward error control packetized zerotree wavelet encoding for transmission of images over varying channels," *IEEE Trans. Image Processing*, vol. 9, pp. 982–993, June 2000.
[22] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC Codes) and their applications," *IEEE Trans. Commun.*, vol. COM-36, pp. 389–400, Apr. 1988.
[23] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Syst. Tech. J.*, vol. 39, pp. 1253–1265, Sept. 1960.
[24] E. O. Elliot, "Estimates of error rates for codes on burst noise channels," *Bell Syst. Tech. J.*, vol. 42, pp. 1977–1997, Sept. 1963.
[25] A. Albanese, J. Blömer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1737–1744, Nov. 1996.
[26] I. V. Bajic, J. W. Woods, and A. M. Chaudry, "Robust transmission of packet video through dispersive packetization and error concealment," in *Proc. Packet Video Workshop (PV 2000)*, Sardinia, Italy, May 2000.
[27] S. D. Servetto, K. Ramchandran, V. A. Vaishampayan, and K. Nahrstedt, "Multiple description wavelet based image coding," *IEEE Trans. Image Processing*, vol. 9, pp. 813–826, May 2000.
[28] W. Jiang and A. Ortega, "Multiple description coding via polyphase transform and selective quantization," in *Proc. SPIE Visual Communication Image Processing Conf.*, San Jose, CA, 1999, pp. 998–1008.
[29] V. R. Algazi and R. R. Estes Jr, "Analysis based coding of image transform and subband coefficients," in *Proc. SPIE Visual Commun. Image Processing Conf.*, San Jose, CA, 1995.
[30] H. Sanneck, "Concealment of lost speech packets using adaptive packetization," in *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, 1998, pp. 140–149.
[31] W. Stallings, *Data and Computer Communications*, 5 ed. Englewood Cliffs, NJ: Prentice-Hall, 1997.
[32] *JPEG 2000 Working Draft 3.0*, ISO/IEC JTC1/SC29/WGI N144, 1999.
[33] N. R. Prasad, "GSM evolution toward third generation UMTS/IMT 2000," in *Proc. IEEE Int. Conf. Personal Wireless Communication*, Jaipur, India, 1999, pp. 50–54.

Taekon Kim (M'01) received the B.S. degree in oceanography from Seoul National University, Seoul, Korea, in 1991, the M.S. degree in electrical engineering from Syracuse University, Syracuse, NY, in 1996, and the Ph.D. degree in electrical engineering from the Pennsylvania State University at University Park in 2001.

Since March 2001, he has been with Technology & Research Labs, Intel Corporation, Chandler, AZ. His research interests include image/video compression and error-resilient transmission techniques over packet network communications.



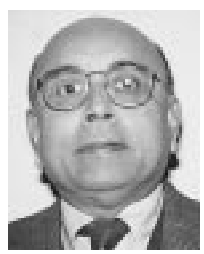
Seungkeun Choi received the B.E. degree in electrical engineering from the Soongsil University, Seoul, Korea, in 1997, and the M.S. degree in electrical engineering from the Pennsylvania State University, University Park, PA, in 2000. He is currently working toward the Ph.D degree in electrical and computer engineering at the Georgia Institute of Technology, Atlanta.

He was a Semiconductor Research Engineer in the Manufacturing Technology Planning Group, LG Semicon Co., Ltd., Cheongju-Si, Korea, from 1997 to 1998, and a Research Assistant at the Center for Spatial and Temporal Signal Processing, Pennsylvania State University. His research interests include multimedia signal processing, error-resilient source/channel coding for wireless communications and data hiding.

Robert E. Van Dyck (S'92–M'92) received the B.E. and M.E.E. degrees from Stevens Institute of Technology, Hoboken, NJ, in 1985 and 1986, respectively, and the Ph.D. degree in electrical engineering from North Carolina State University at Raleigh in 1992.

Since June 2000, he has been a member of the Advanced Network Technologies Division of the National Institute of Standards and Technology, Gaithersburg, MD. Prior to that, he was an Assistant Professor in the Department of Electrical Engineering, the Pennsylvania State University, University Park, PA. During 1999, he was a Summer Faculty Research Fellow at Rome Laboratory. His other previous affiliations include GEC-Marconi Electronic Systems, Wayne, NJ (1995–1996), the Center for Computer Aids for Industrial Productivity, Rutgers University, Piscataway, NJ (1992–1995), the Computer Science Company, Research Triangle Park NC (1989), and the Communications Laboratory, Raytheon Co., Marlborough, MA (1985–1988). His research interests are in multimedia communication and networking, video signal processing, and source and channel coding for wireless communications.

Dr. Van Dyck is a member of Eta Kappa Nu, Pi Mu Epsilon, and Sigma Xi.



Nirmal K. Bose (F'81) received the B.Tech (Hons.) degree from the Indian Institute of Technology (I.I.T.), Kharagpur, India, the M.S. degree from Cornell University, Ithaca, NY, and the Ph.D. degree from Syracuse University, Syracuse, NY, all in electrical engineering.

From 1967 to 1986, he was with the University of Pittsburgh, Pittsburgh, PA, where he became a Professor of Electrical Engineering and Professor of Mathematics. Currently, he is the HRB-Systems Professor of Electrical Engineering at The Pennsylvania State University, University Park, which he joined in 1986 as the Singer Professor and Founding Director of the Center for Spatial and Temporal Signal Processing. He was a visiting faculty member at several institutions, including the American University of Beirut in Lebanon, the University of Maryland, CNRS-NRIA in Paris, France, LAAS in Toulouse, France, Ruhr Universitaet in Germany, the University of California at Berkeley, and Princeton University, Princeton, NJ. During 1994–1995, he was invited by the United Nations to serve as a Technical Advisor to the Government of India. He is the author of *Applied Multidimensional Systems Theory* (New York: Van Nostrand Reinhold, 1982), *Digital Filters* (Amsterdam, The Netherlands: Elsevier, 1985; Malabar, FL: Krieger, 1993), author and editor of *Multidimensional Systems: Progress, Directions and Open Problems* (Amsterdam, the Netherlands: Reidel, 1985), and co-author of *Neural Network Fundamentals with Graphs, Algorithms and Applications* (New York: McGraw-Hill, 1996). He is the founding Editor-In-Chief of the *International Journal on Multidimensional Systems and Signal Processing* and serves on the editorial boards of several journals.

Dr. Bose was invited to serve as the plenary Keynote Speaker on several occasions, including the First and Second International Workshops on Multidimensional Systems in Poland in 1998 and 2000. He received the Alexander Von Humboldt Research Award of Germany in 2000 and the Invitational Fellowship from the Japan Society for the Promotion of Science in 1999.