Adaptive Embedded Zero Tree for Scalable Video Coding

Roya Choupani, Stephan Wong, and Mehmet R. Tolun

Abstract—**Video streaming over the Internet has gained popularity during recent years mainly due to the revival of video-conferencing and video-telephony applications and the proliferation of (video) content providers. However, the heterogeneous, dynamic, and best-effort nature of the Internet cannot always guarantee a certain bandwidth for an application utilizing the Internet. Scalability has been introduced to deal with such issues (up to a certain point) by adapting the video quality with the available bandwidth. In addition, wavelet-based scalability combined with representation methods such as embedded zero trees (EZWs) provides the possibility of reconstructing the video even when only the initial part of the streams have been received. EZW prioritizes the wavelet coefficients based on their energy content. Our experiments however, indicate that giving more priority to low frequency content improves the video quality at a specific bit rate. In this paper, we propose a method to improve on the compression rate of the EZW by prioritizing the coefficients by combining each frequency sub-band with its energy content. Initial experimental show that the first two layers of the generated EZW are about 22.6% more concise.**

Index Terms—**Video coding, Scalability, Wavelet, Embedded Zero Tree**

I. Introduction

In the past several years, the steady growth in the bandwidth on the Internet has allowed increasingly more applications to incorporate streaming audio and video content [1],[2]. As the best-effort Internet is not capable of providing fixed bandwidth, the objective of video coding has been to provide the best possible video quality at different bit rates. The impracticality of storing multiple copies of a video (sequence) at different bit-rates, resolution, etc. created the need for scalability that allows video to be easily recoded to meet different requirements depending on a particular situation. Video scalability entails the logical subdivision of a single video stream into a base layer stream with multiple enhancement layer streams. The base layer stream encompasses the most rudimentary quality and/or resolution of the video. However, combined with the enhancement layer streams the resolution (spatial scalability), number of frames (temporal scalability), or video quality (SNR scalability) can be improved. Consequently, the transmission of the base layer stream receives the highest priority while the other streams are less important. On top of this, the less than perfect nature of the Internet can lead to packet losses or lengthy packet retransmissions (also effectively losing packets when displaying video in real-time). Therefore, coding schemes have been introduced in the recent past to deal with such conditions. One such scheme incorporates wavelet transforms that are effectively capable of generating different sub-bands with - in simple terms - important coefficients and less important coefficients. The ensuing embedded zero tree (EZW) method allows an ordered representation of these coefficient from important to less important by ignoring the coefficients less than a threshold, while reducing the threshold at each iteration. This means that the classification of the coefficients as significant and insignificant in EZW is based on their magnitude only. However, the coefficiens corresponding to low frequency content of the image contain more important and basic information which should be transmitted at the first stage. In this paper, we propose an alternate representation method for the EZW coefficients that organizes them according to metric which combines the frequency sub-band with the magnitude of energy at a given frequency. This feature has the benefit of providing better video quality at a given bit rate than the standard EZW method. The experiments with some video sequences show that on average 22.6% reduction in data size is achieved on the first two layers.

The remainder of this paper is organized as follows. In Section II, we discuss the related work and highlight how our proposed method differs from them. In Section IV, we introduce our proposed method and describe its encoding and decoding algorithms providing an example. In Section V, we explain our experimental setup and discuss the results. In Section VI, we draw our conclusions.

II. Related Work

Rapid advances in multimedia technologies together with the growth of the Internet have enabled many new applications and services. Although network bandwidth and digital devices storage capacity are increasing rapidly, video compression continues to play an important role due to the exponential growth in the size of multimedia content. Furthermore, many applications require not only high compression efficiency, but also enhanced flexibility for supporting real time usage. For example, in order to effectively deliver video over heterogeneous networks such as the Internet and wireless channels, error resilience and bit rate scalability are important; and in order to make a coded video bitstream usable by different types of digital devices regardless of their computational, display, and memory capabilities, resolution/temporal scalability is needed. Standard coders such as H.26x and MPEG-x are no longer able to satisfy the above requirements with their incorporated scalability. The recent MPEG-4 standard [3] adopts object-based video coding in order to support more applications. However, scalability in MPEG-4 is limited. Experiments with MPEG-2 and H.263 in

Roya Choupani is a PhD candidate at the Computer Engineering Department of Delft University of Technology, Delft, the Netherlands. Email: roya@cankaya.edu.tr

Associate Professor Dr. Stephan Wong is with the Computer Engineering Department of Delft University of Technology, Delft, the Netherlands. Email: J.S.S.M.Wong@tudelft.nl

Professor Mehmet Tolun is with the Computer Engineering Department of Çankaya University, Ankara, Turkey. Email: tolun@cankaya.edu.tr

scalable mode show that, compared with non-layered coding [5][7][10], the average peak signal-to-noise ratio (PSNR) decreases about one dB with each layer. Furthermore, it is difficult for these coding schemes to achieve scalability because there is always a potential drifting problem [9][12] associated with predictive coding. H.264 tries to control the drift error by introducing a new concept called key pictures [4]. Key frames are not necessarily intra-coded frames. For each key picture a flag is transmitted, that signals whether only the base layer information or together with the enhancement layer information was utilized in the reconstruction of reference frames in the motion estimation. By introducing a hierarchical reference frame organization, H.264 allows all enhancement layer frames to utilize the references with the highest available quality for motion estimation, which enables a high coding efficiency for these key pictures [8]. However, drift error is not eliminated completely although its effect is minimized and limited to the frames between two consecutive key pictures [6]. The tradeoff between enhancement layer coding efficiency and drift error can be adjusted by the choice of the number of frames between two consecutive key pictures or the number of hierarchy stages. Meanwhile there are proposals for MPEG-4 streaming video profile on fine-granularity scalable video coding. However, these proposals are limited to providing flexible rate scalability only and the coding performance is still about 1-1.5 dB lower than that of a non-layered coding scheme [9]. As an alternative to the predictive approaches in various video coding standards, wavelet video coding has been investigated recently by several researchers [11][2] and shown to be competitive with standard motion compensated (MC) predictive coding. Although wavelet video coders usually require a larger buffer and incur a longer delay than standard coders, an important feature of the wavelet approach is the support of scalability in the compressed video. With embedded coding techniques such as embedded zero tree and set partitioning in hierarchical trees, wavelet video coding achieves continuous bit rate scalability. Furthermore, because of the multi-resolutional nature of wavelet analysis, both spatial (resolution) and temporal (frame rate) scalabilities can be easily supported. Mallat [14] discusses the wavelet representation as a suitable tool for multi-resolution signal decomposition. Such a decomposition of video signals may permit temporal and spatial scalability. This important feature is investigated by many researchers. One of the main reasons for the success use of DWT in scalable video coding is the introduction of data structures to represent wavelet coefficients while minimizing the required memory space.

Embedded Zero Tree (EZW) is one of these data structures which is widely used for organizing and transmission of the wavelet coefficients. A drawback of EZW is that the large coefficients in high frequency sub-bands appear at the first passes affecting the compression rate in low bit-rate. This drawback stems from the fact that EZW is not basically designed for scalable video coding where the base layer is expected to contain the common and the most important data. Hence the present scheme puts priority on the magnitude of the coefficient rather than the associated frequency of each coefficient. The proposed method described below is an improvement over EZW which enhances the priority encoding of the wavelet coefficients in EZW. In the following section we briefly explain EZW first. Then the proposed improvements are introduced.

III. Embedded Zero Tree (EZW)

The Embedded Zero Tree Wavelet encoding is based on creating a quad-tree having four children at each node (except the root node that only has 3 children) and storing the obtained coefficients from the wavelet transform. The main reason to utilize this storage method is that the coefficients close to the root have a larger absolute value. Therefore, coefficients closer to the leaves of the tree are less significant and can be sometimes represented by a single symbol to greatly reduce the data size. The algorithm follows the procedure described below. The magnitude of each wavelet coefficient in the quad-tree, starting with the root of the tree is compared to a threshold T. If the magnitudes of all the wavelet coefficients in the (sub)tree - including the root of the (sub)tree - are smaller than T, the entire (sub)tree is substituted by a single symbol, the zero tree symbol ***t***. However, if the magnitude of a coefficient in the subtree excluding the root is larger than T, it is substituted by a symbol representing whether the subtree is either a significant (if the root is larger than T) or insignificant (if the root is smaller than T). In the latter two cases, descendant nodes (and their respective subtrees) are further examined to determine whether they contain zero trees or not. This process is carried out until all the nodes in all the trees are examined for possible sub zero tree structures. The significant wavelet coefficients in a tree are represented by one of two symbols, ***P*** or ***N***, depending on whether their values are positive or negative, respectively. An insignificant coefficient which is not in a zero tree is represented by a ***Z*** symbol. The process of classifying the coefficients as ***t***, ***Z***, ***P***, or ***N*** is referred to as the dominant pass. This is subsequently followed by a subordinate pass in which the significant wavelet coefficients in the image are refined by determining whether their magnitudes lie within the intervals [T,3T/2) and [3T/2,2T). Those wavelet coefficients whose magnitudes lie in the interval [T,3T/2) are represented by the symbol 0 (LOW), whereas those with magnitudes lying in the interval [3T/2,2T) are represented by the symbol 1 (HIGH). To refine the coefficients which were marked either as ***P*** or ***N*** in the subordinate pass, we push them in a FIFO list in the dominant pass. These coefficients are reduced by current threshold value after refining step. Subsequent to the completion of both the dominant and subordinate passes, the threshold value T is reduced by a factor of 2, and the process is repeated. An EZW decoder reconstructs the image by progressively updating the values of each wavelet coefficient in a tree as it receives the data. The following example illustrates two steps of EZW encoder. Given the coefficient values depicted in Figure 1 and assuming the initial threshold value to be 32 we have:

Dominant Pass 1: PNZtPttttZttttttPtt

Coefficients in FIFO : 61, -40, 51, 52

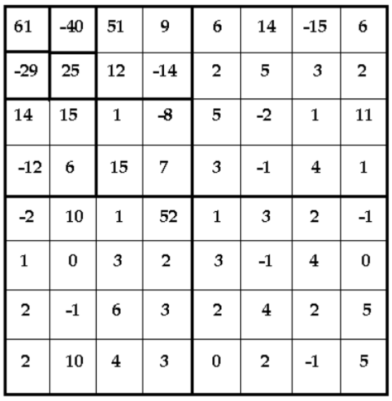
Subordinate Pass 1: 1011

The 2nd pass with a threshold of 16 is presented in the following:

Dominant Pass 2: ZtNPtttttttt

Coefficients in FIFO : 29, -8, 19, 20, -29, 25

Subordinate Pass 2: 100011



61	-40	51	9	6	14	-15	6
-29	25	12	-14	2	5	3	2
14	15	1	-8	5	-2	1	11
-12	6	15	7	3	-1	4	1
-2	10	1	52	1	3	2	-1
1	0	3	2	3	-1	4	0
2	-1	6	3	2	4	2	5
2	10	4	3	0	2	-1	5

Fig. 1. Sample data representing wavelet coefficients

IV. Proposed Method

It is obvious from the previous description that the initial number of large coefficients before any pass determines the amount of encoded data needed in the ensuing pass. Considering the scalability issues in low bandwidth networks, we propose a method to reduce the amount of encoded data in the first pass and defer transmission of the additional data to following passes. The priority mechanism of EZW is improved by considering a level based weight. The coefficients stored in the quad tree are multiplied by a weight value before thresholding. Adjusting weight values in a decreasing order defers transmission of the significant coefficients coming from the high frequencies. The following subsections describe encoding and decoding procedures of the proposed method.

- **Encoding**

In the example given above, the dominant pass in the first iteration includes four **t**s followed by a **Z**, 7 **t**s, a **P** and 2 **t**s. If the element in the 5^{th} row and 4^{th} column which is equal to 52 (greater than the threshold) is replaced by a small number, the dominant pass string will be PNZtPttttttttttt which is four symbols shorter than the original string. Replacing this entry however, generates some difference with the original image. This deviation from the original values becomes more important when we increase the number of levels decoded at the receiver side since the elements at the lower levels of the quad tree correspond to high frequency content of the image. This means that in the coarse level replacing isolated elements with values less than the threshold will not create a major visual effect. To replace the large coefficient values in the low levels of the tree when encoding at the coarse stage, we propose a weighted form of representing the coefficients. Assume the quad tree has a height of n. For each level a weight factor $w_i$is defined as a positive number so that $w_1 = 1$ and $w_i < w_j$ if i>j. During the dominant pass the values are multiplied by their corresponding weight factors before comparing to the threshold. As the weight factors are less than one, the large coefficients will not appear in encoding with large threshold values (coarse stage). In the subordinate pass the bit strings are obtained using the weighted coefficients. The method applied to the sample data will produce the following results:

Weights: w_1= 1, w_2= 0.8, w_3= 0.6

Initial threshold value = 32

Dominant Pass 1: PNZttttt

Coefficients in FIFO : 61, -40

Subordinate Pass 1: 10

Pass 2, Threshold = 16

Dominant Pass 2: ZZNP Pttt tZtt tttt tttt tPtt

Coefficients in FIFO : 29, -8, -29, 25, 30(51x0.6), 20(52x0.4)

Subordinate Pass 2: 101110

The large coefficients which do not appear in the first pass emerge in the second pass. This justifies the long symbol string in the second pass. In fact the method defers the processing of some of the symbols which contain large frequency information to subsequent passes which is more suitable for a scalable video coding scheme. Also the coefficients in the FIFO have been truncated after being multiplied by their respective weight factors.

- **Decoding**

To avoid loss of information the extracted coefficients are divided by the corresponding weight factors. This reconstruction of the wavelet coefficients introduces some error due to the truncation process in the subordinate passes. In the above example the reconstructed values after the second pass are as follows.

EZW:

Original coefficient	Reconstructed coefficient
61	56
-40	-32
51	48
52	48
-29	-24
25	24

Weighted EZW:

Original coefficient	Reconstructed coefficient
61	56
-40	-32
51	40
52	40
-29	-24
25	24

Despite the fact that the reconstructed values for the coefficients in the first passes is farther from their true value, closer approximation is obtained in the next passes. This is also compliant with the main idea of the proposed method which aims at deferring the exact reconstruction to the late passes and reducing the string length in the first passes.

V. Experimental Results

To evaluate the performance of the proposed method we have created the zero trees for two video sequences. The videos use gray scale values so a single pixel value component is available. Number of levels in the zero trees has been considered as a parameter. An important issue in the proposed method is adjusting the weight values for each tree level. The larger weight values create a zero tree which is close to the EZW tree. Smaller weight values on the other hand pushes the larger list of coefficient values to the lower levels of the tree which means the smaller the weight factor is, the more compact the frame size will be in the upper levels. This feature is specially important since reducing frame size when the bandwidth is limited for a given connection or display quality is low, should be performed by increasing the number of levels in the hierarchical tree. However, increasing the number of levels in the generated zero trees means more processing time will be needed due to wavelet transform which is a serious consideration for real time applications. A suitable choice of weight value can adequately solve this problem. Figures 2 and 3 shows the one frame of each sample video used for evaluation.

Fig. 2. Sample frame of the test video with low frequency content

Fig. 3. Sample frame of the test video with high frequency content

We have also compared the bit per pixel rate with respect to the peak signal to noise ratio (PSNR). In Figures 4 and 5 the reconstructed video quality in terms of PSNR is depicted versus bit-per-pixel for both original EZW and the proposed weighted EZW methods. As our aim is comparing the performance of the original EZW with the proposed weighted EZW, we have not considered encoding the difference between the frames and each frames undertakes DWT transform, EZW coding, and entropy coding of the coefficients. The important difference between the examples at Figures 4 and 5 is that the video shown at Figure 3 includes more high frequency content. Therefore the difference between the original EZW and the proposed weighted EZW methods is more. The compression rates using EZW and Weighted-EZW methods have been compared with respect to the number of levels in the generated trees. The results of comparison are given in table I. We have only considered the range of 3 to 5 for tree levels however, the results show that there exists an increase in the compression rate with increase in the number of levels.

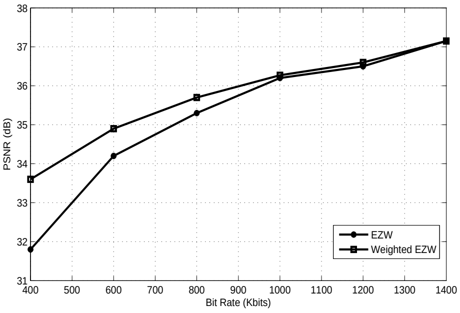


Fig. 4. Reconstructed images using first two level of zero trees. **Left:** Original EZW **Right:** Weighted EZW

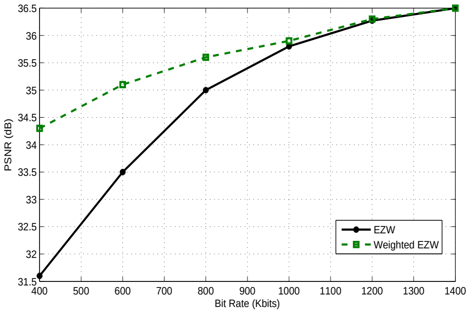


Fig. 5. Reconstructed images using first two level of zero trees. **Left:** Original EZW **Right:** Weighted EZW

TABLE I
RATE COMPARISON FOR EZW (LEFT) AND WEIGHTED-EZW (RIGHT)

Levels	Pass 1	Pass2	Pass 1	Pass2
3	69.3	50	87.5	54.4
4	73.2	71.1	90.2	80.33
5	78.8	76.43	94.11	89.6

VI. CONCLUSION

Large fluctuations in the communication bandwidth and heterogeneity of the Internet are two main reasons for using scalability in encoding video data. The modification introduced here for Embedded Zero Tree algorithm further reduces the data size in the low frequency and coarse stage of video coding and makes quick transmission of the data feasible. The method eliminates the need for a large number of layers of wavelet tree when higher compression rate is desired. This feature is the result of pushing the details to the lower layers of the tree and therefore large data sizes are seen only at the lower levels of the tree. The experimental results show that using only a single level, 22.6% improvement can be achieved with almost no visually noticeable decrease in video quality. In case of using two layers for reconstruction the improvement rate drops to 9.13% which indicates that the method is suitable for low resolution display screens or low bandwidth channels where high rate of compression is necessary.

REFERENCES

[1] G. Conklin, G. Greenbaum, K. Lillevold, A. Lippman, and Y. Reznik, Video Coding for Streaming Media Delivery on the Internet, IEEE Trans. Circuits and Systems for Video Technology, March 2001.

[2] D. Wu, Y. Hou, W. Zhu, Y.-Q. Zhang, and J. Peha, Streaming Video over the Internet: Approaches and Directions, IEEE Trans. Circuits and Systems for Video Technology, March 2001.

[3] Coding of Audio-Visual Objects, Part-2 Visual, Amendment 4: Streaming Video Profile, ISO/IEC 14 496-2/FPDAM4, July 2000.

[4] H. Schwarz, D. Marpe, and T. Wiegand, ”Overview of Scalable Video Coding Extension of the H264/AVC Standard”, IEEE Transaction on Circuits and Systems for Video Technology, Vol. 17, No. 9, September 2007.

[5] B. G. Haskell, A. Puri, and A. N. Netravali, ”Digital Video: An Introduction to MPEG-2”, New York: Chapman and Hall, Sept. 1996.

[6] A. Segall, CE 8: SVC-to-AVC Bit-Stream Rewriting for Coarse Grain Scalability, Joint Video Team, Doc. JVT-V035, Jan. 2007.

[7] R. Aravind, M. R. Civanlar, and A. R. Reibman, Packet loss resilience of MPEG-2 scalable video coding algorithm, IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 426–435, Oct. 1996.

[8] H. Kirchhoffer, H. Schwarz, and T. Wiegand, CE1: Simplified FGS, Joint Video Team, Doc. JVT-W090, Apr. 2007.

[9] S. F. Chang and A. Vetro, ”Video adaptation: Concepts, technologies, and open issues,” Proc. IEEE, vol. 93, no. 1, pp. 148–158, Jan. 2005.

[10] ”Video coding for low bit rate communication,” Int. Telecommun. Union-Telecommun. (ITU-T), Geneva, Switzerland, Recommendation H.263, 1998.

[11] G. Strang, ”Wavelets”, American Scientist, Vol. 82, 1992, pp. 250–255.

[12] S. Nogaki, M. Ohta, ”An overlapped block motion compensation for high quality motion picture coding” Proc. IEEE Int. Symp. Circuits and Systems, vol. 1, pp. 184–187, 1992

[13] J. M. Shapiro, ”Embedded image coding using zerotrees of wavelets coefficients”, IEEE Transactions on Signal Processing, vol. 41, no. 12, pp. 3445–3462, December 1993.

[14] S. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, IEEE Trans. Pattern Analysis and Machine Intelligence 11 (7), pp. 674–693, 1989.