A Simple and Robust Method for Image Compression based on the EZW Algorithm

Andrezza A. Gusmão and Max H. M. Costa
Department of Communications – FEEC - Unicamp
P.O. Box 6101 – 13083-970 - Campinas – SP- Brazil
andrezza, max@decom.fee.unicamp.br

Abstract: **In this article, a simple and robust algorithm for image compression is proposed. The algorithm is a modified version of Creusere´s Robust Embedded Zerotree Wavelet (REZW) algorithm [1], where arithmetic coding is replaced by Huffman coding. It is intended for use in noisy channels such as those found in wireless communications.**

I. Introduction

The current trend in telecommunications is towards digital networks with integrated services where voice, audio, still-image, video, and data signals share a common telecommunication network. This trend is observed both in wireless networks and in high-speed optical fiber networks. The transmission of image and video signals tend to be particularly demanding of the communication system, in terms of the high rate requirements. Thus, particularly in wireless applications, image and video compression are important operations for the feasibility of such integrated networks. Naturally, these operations must be sufficiently robust to deal with the possibility of noisy transmission environments.

In this paper we investigate the use of a simple and robust method of image compression that shows graceful degradation of performance in noisy environments. The motivation is its application in low speed, high bit error rate (BER) channels such as found in mobile communications. The method is based on the embedded zerotree wavelet (EZW) algorithm, proposed by Shapiro. More specifically, the method is a variation of Creusere´s robust embedded zerotree wavelet (REZW) algorithm [1], where arithmetic coding is replaced by Huffman coding. The advantage of this substitution is the simplicity of the resulting technique, and the fact that Huffman coding is not a proprietary technology as is arithmetic coding.

The EZW algorithm exploits the multiresolution nature of the wavelet decomposition and the correlation that exists among wavelet coefficients of similar spatial coordinates but different scales. This method is a powerful image compression technique that has the feature of progressive coding of the image data. This progressive nature is what is meant by the term “embedded”. It allows for the interruption of the encoded stream at any point, in a way that the information content of the truncated stream is maximized. Many extensions of the EZW algorithm have been proposed. One example is the SPIHT algorithm [3], that yields the best rate × distortion performance to date in many applications.

Another variation of the EZW algorithm is the cited REZW algorithm. This technique was proposed to increase the protection of EZW-encoded data against bit errors produced by the channel. EZW is very sensitive to bit errors because of the resulting error propagation. The increased robustness of REZW is achieved by separating the wavelet coefficients in groups that are quantized and encoded independently. When the encoded stream is faced with a bit error, only the data corresponding to the particular group of the affected bit is jeopardized. The algorithm has a mechanism to stop the decoding of the distorted group not far from the error position, to avoid error propagation. Like EZW, REZW uses arithmetic coding [4] for the entropy coding stage. Arithmetic coding was invented in the 1970’s as an alternative lossless source-coding scheme that self adapts to the source statistics, and that achieves asymptotically optimal performance. Nevertheless, there are reasons to consider a different entropy coding scheme. Arithmetic coding is not a particularly simple scheme, and has the disadvantage of being a proprietary technique.

In this article a modified REZW algorithm is proposed, where Huffman coding is used as the entropy coding operation. The modified algorithm is computationally simpler than REZW, at the cost of a small loss in rate × distortion performance. The Huffman encoding or decoding operation is usually performed by table look-up. Although table look-ups can be computationally intensive, in the present case the code tables are extremely small (of sizes 4 or 5), leading to a very simple Huffman operation.

The article is organized as follows. The REZW algorithm and the proposed modified version are described in

Section 2. The results of simulations and comparisons are presented in Section 3. Finally, conclusions are presented in Section 4.

II. THE REZW AND THE MODIFIED REZW ALGORITHMS

We start with a brief description of the EZW algorithm. Its general structure is depicted in Fig. 1.

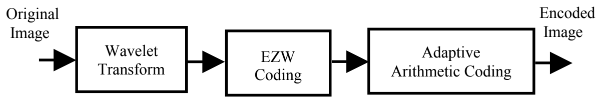

Fig. 1: Block diagram of the EZW algorithm.

In the first stage the data is wavelet transformed to reduce correlation among pixels. The second phase is known as EZW encoding. The transformed data is organized in tree structures that link the bits corresponding to wavelet coefficients of similar spatial locations, but different scales, on each bit plane. The purpose of these tree structures is to exploit the correlation that has been found to remain among these coefficients. The tree structures imposes a reordering of the data that tends to show long trees of zeros, the zero trees, that can be efficiently compressed. The third stage completes the process with arithmetic coding.

In order to exploit the residual correlation among wavelet coefficients, these are grouped in a structure known as descendence tree. Fig. 2 shows an illustration of such a tree. The lowest frequency subband is taken as the root of the tree. Each coefficient of this subband is a parent of three children, one in each of the subbands of the same scale but different orientation (cf. Fig. 2). Each of these children coefficients, in turn, have 4 children of similar position and orientation on the next scale, and so successively. In Fig. 2, the squares with "x" depict an entire descendence tree.

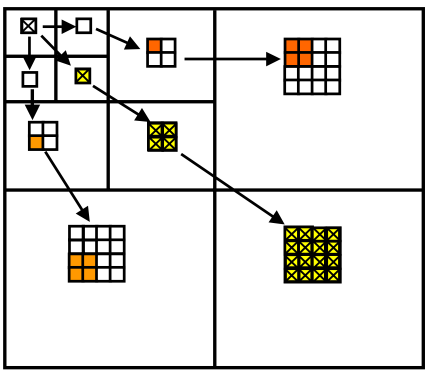

Fig. 2: Tree structures for EZW encoding

The EZW encoder scans the wavelet coefficients in a pre-established order so as to exploit correlation across scales. The scan is accomplished in successive bit planes, in a way that, for each bit plane, no child is considered before its parent, and all coefficients in a particular scale are scanned prior to those in a subsequent scale. This encoder uses the fact that wavelet coefficients tend to decrease in magnitude as the scales become finer. So, if a certain coefficient shows a small value of magnitude, there is a good chance that all its descendents will also be small.

As wavelet coefficients are scanned, they are automatically quantized at different bit planes [5], in a process known as successive approximation quantization (SAQ). The output of this process is encoded in 4 possible symbols. They are:

1) POS (Positive and Significant): the coefficient is larger that the current threshold and has positive sign;
2) NEG (Negative and Significant): the coefficient is larger that the current threshold and has negative sign;
3) IZ (Isolated Zero): the coefficient is smaller than the current threshold, but one or more of its descendents is larger;
4) ZTR (Zerotree Root): the coefficient and all of its descendents are smaller than the current threshold.

The sequence of these symbols and some auxiliary bits are finally encoded by the arithmetic encoder and transmitted over the channel.

The REZW algorithm was proposed by Creusere [1]. It is a variation of the EZW algorithm where the stream of transmitted bits is split into a number of subsequences that are interleaved prior to transmission. Therefore, when a bit error hits one of the subsequences, only that particular portion of the overall information is corrupted. The REZW decoder has a mechanism to recognize the error in that subsequence, and to interrupt the corresponding decoding process, so that error propagation is avoided. Naturally, the decoding of the other subsequences is unaffected by this mechanism.

The overall structure of the REZW algorithm is illustrated in Fig. 3. The wavelet coefficients are initially split into S groups, which are then independently quantized and encoded by a regular EZW encoder. The bit streams are then interleaved as appropriate, in bit, bytes or packets, prior to channel transmission. The interleaving operation is aimed at maintaining the embedded nature of the encoding algorithm.

For the REZW algorithm to be effective, the group of wavelet coefficients pertaining to each subsequence must be of equal sizes, and must homogeneously encode the image. This way, if a particular group is lost due to channel errors, the distortion caused by the partial loss of coefficients is uniformly distributed over the image.

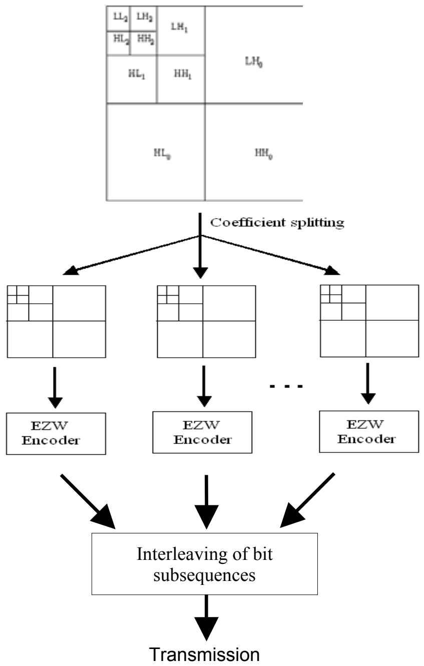

Fig. 3: The structure of the REZW algorithm.

Fig. 4 shows an example where the wavelet transform over 3 scales has its coefficients split into S=4 groups. The possibilities for the number of subsequences (S) is not arbitrary. The desired properties of the algorithm are maintained if S is an integer power of 4.

Let the wavelet coefficients of scale *j* and position *(x,y)* be denoted by W_j (x,y) The splitting of the wavelet coefficients into $S=4^k$ groups is governed by

$$W_{\Psi,j}(x,y) = W_j\left(\begin{matrix} 2^{M-j-1}\left\{\left\lfloor \frac{x}{2^{M-j-1}} \right\rfloor .(2^k-1)+n\right\}+x, \\ 2^{M-j-1}\left\{\left\lfloor \frac{y}{2^{M-j-1}} \right\rfloor .(2^k-1)+m\right\}+y \end{matrix}\right) \quad (2)$$

where $\Psi = n + 2^k.m$ specifies the group index, $\{n,m\} \in [0,2^k-1]$, and $\lfloor . \rfloor$ denotes the integer part of the argument (the floor function).

Scale and frequency are inversely proportional. For example, scales j=0 and j=M-1 correspond to the highest and the lowest frequency bands, respectively.

All the coefficients of any given zerotree are kept together in the same group. Therefore, the correlation of coefficients of different scales and similar positions and orientations are still explores as in the original EZW algorithm.
We note that S can be increased in powers of 4 until the point in which the encoder processes a single zerotree per subsequence. If the image size is XxY (assuming, for simplicity, that X and Y are powers of 2) and M scales of wavelet coefficients are used, then the maximum number of independent subsequences is $S=X.Y/4^M$. As S is increased the algorithm sensitivity to errors is decreased, that is, robustness is improved. On the other hand, if the channel BER is very small, the rate×distortion performance with larger values of S is somewhat inferior because of the accompanying overhead.

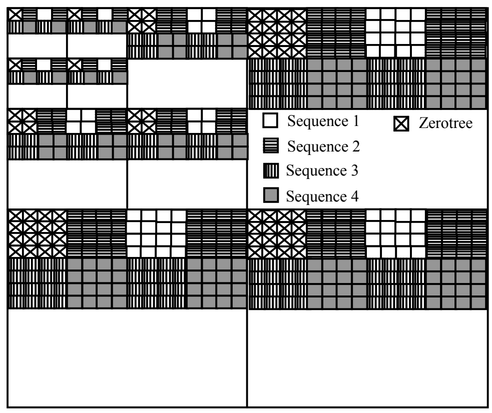

Fig. 4: Wavelet coefficients splitting into S=4 groups for M=3 scales.

The algorithm considered in this paper is a modified version of the REZW algorithm. The modification is basically the substitution of the arithmetic encoding stage (that is present in both, EZW and in REZW algorithms), by a simpler Huffman coding stage. We will refer to it as the modified algorithm. Our aim in replacing the entropy encoding stage of the algorithm was the increased computational simplicity and the avoidance of proprietary technology.

The outputs of the various EZW encoded subsequences are represented by symbols of the alphabet {POS, NEG, IZ e ZTR} (and by some additional auxiliary bits) in all subband, except for the last subband. Since the coeeficients in the last subband have no descendents, the IZ and ZTR symbols can be coalesced into a single symbol, denoted by Z (Zero). Therefore the code alphabet for the coefficients of the last subband is composed of the set {POS, NEG, Z}, which contributes for some reduction in the average length of the Huffman code.

An additional symbol must be added to the alphabets of the Huffman encoded sources. It is called the STOP symbol. The Huffman codeword for this symbol is never transmitted. But, in the event of an error in the transmitted subsequence, the corresponding Huffman decoder will lose its synchronism and after some time, with high probability, will read the codeword for the STOP symbol, and command the decoder to a halt. In these circumstances, the earlier the STOP codeword is met, the least amount of corrupt information will be incorporated in the reconstructed image. To increase the probability that the STOP word is encountered following a transmission error, a data scrambler can be incorporated at each encoder output, prior to the interleaver, with the corresponding descramblers (one for each subsequence) being added at the decoder.

The Huffman code was obtained from the empirical probabilities of the various symbols in the alphabets. The codes were the following. For all scales but the last: {ZTR = 1, IZ = 01, POS = 001, NEG = 0001, STOP = 0000}; for the last scale: {Z=1, POS = 01, NEG = 001, STOP = 000}. The choices on the STOP codeword were made so that no symbol codeword was longer than the STOP codeword.

III. SIMULATION RESULTS

This section describes the results of simulations of the proposed algorithm and compares them with Creusere's results.

In our simulations we have utilized the biorthogonal 9/7 wavelet transform (p=9 e q=7) with 5 levels of decomposition. These filters were selected because they yield one of the best rate × distortion performances in image compression, as investigated by Villasenor et al. [5].

All possible splittings of the original encoded sequence were considered, i.e., S = 1, 4, 16, 64 and 256. The simulations were done for the compressed rate of 1 bit per pixel. The chosen test images were the Lena and the Barbara images, with V=512 pixels per column, and H=512 pixels per line. The use of scramblers and descramblers to reduce the decoding of corrupt information, as suggested in the end of Section II, was not implemented in the simulations.

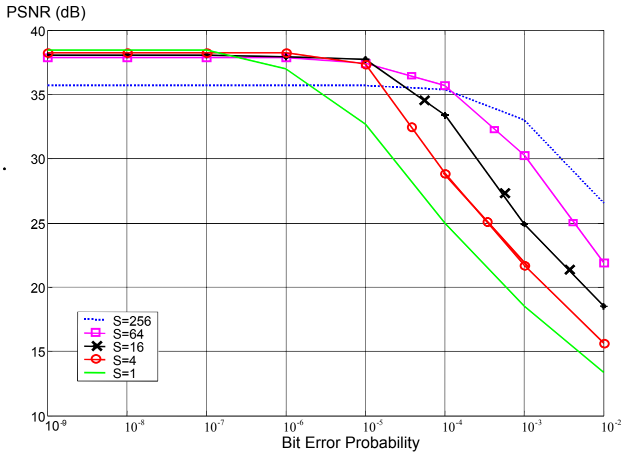

Fig. 5: PSNR versus BER for the Lena image.

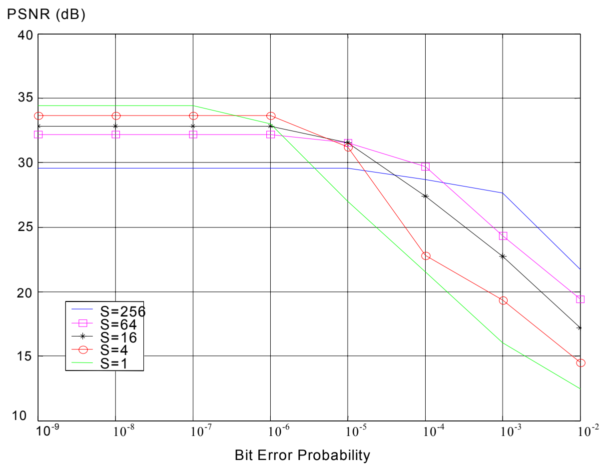

Fig. 6: PSNR versus BER for the Barbara image.

Figs. 5 and 6 present the simulation results in terms of the peak-signal-to-noise ratio (PSNR), in dB, plotted as a function of the channel bit error rate (BER), for the Lena and Barbara images, respectively. The curves obtained for the various values of S, with the modified algorithm, are shown. We note that, as the BER increases, the schemes with more subsequences (high values of S) are more robust. On the other hand, at very small values of BER, the added overhead of using more subsequences becomes apparent.

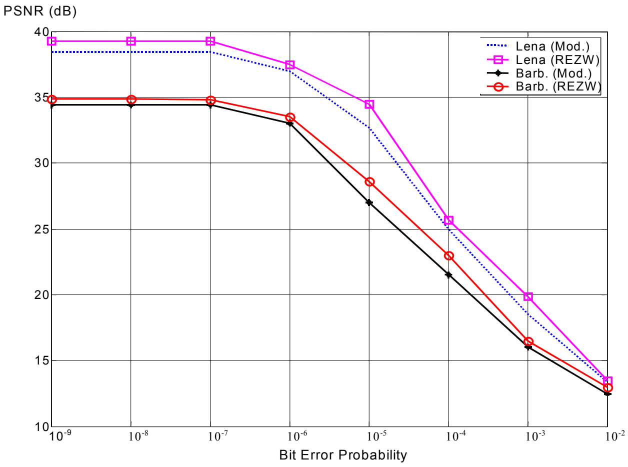

Fig. 7: PSNR versus BER with S=1 (REZW and modified algorithms).

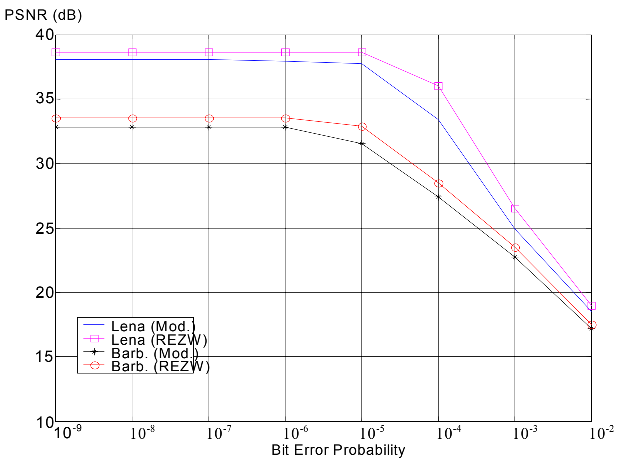

Fig. 8: PSNR versus BER with S=16 (REZW and modified algorithms).

(a) (b)

Fig. 9: Lena reconstruction with the modified algorithm:
(a) S=16, BER = 10^{-3}, PSNR=24.93 dB;
(b) S=1, BER = 10^{-3}, PSNR=18.53 dB.

Figures 7 through 13 present a comparison between the REZW and the modified algorithms for some values of S. In these figures, the PSNR values for the REZW algorithm were taken from [1]. Fig. 7 presents a comparison for S=1, i.e., when the data sequence is not divided into multiple subsequences. Naturally, no increase in robustness is observed in this case, but it serves to compare the overheads of the algorithms. Fig. 8 presents objective quality (PSNR) comparisons for S=16 subsequences. The added robustness achieved with the modified algorithm is illustrated by Fig. 9, where image reproductions are shown for BER=10^{-3}, S=16 and S=1. Similarly, Figs. 10 and 11 present comparisons to show the performances with S=64, and Figs. 12 and 13 illustrate the performances with S=256. It is found that the substitution of arithmetic code by Huffman codes represents a penalty of a fraction of a dB, in most cases. Also, the figures demonstrate the subjective quality of reproductions obtained with multiple values of S as compared to equivalent reproductions with S=1. The difference in subjective quality is that of an image with some noticeable distortion and a totally useless image.

IV. Conclusions

A simple and robust image compression algorithm has been presented. The method is a modified version of the REZW algorithm, proposed by Creusere [1]. The modification is the replacement of arithmetic coding by Huffman coding in the entropy coding stage. Application of this algorithm to the Lena and Barbara images shows that the PSNR × BER of the modified algorithm is, in most cases, a fraction of a dB inferior to that of the REZW algorithm (as taken from [1]). Thus, the modified algorithm pays a price in its rate × distortion performance, but, in compensation, it uses of a simpler and non-proprietary technology.

The PSNR × BER curves obtained for the Lena and Barbara images show that graceful degradation is achieved as the channel BER is increased. As expected, as the number (S) of encoded subsequences is increased, a higher level of robustness (less sensitivity to channel errors) is achieved. The results also show the cost of the algorithm overhead when the channel BER is very low. Naturally, this is higher for larger values of S (around 0.5 dB for S=64, and around 2.7 dB for S=256). It appears that intermediary values of S, such as 64, are a good compromise between an appropriate level of robustness and a low overhead cost.

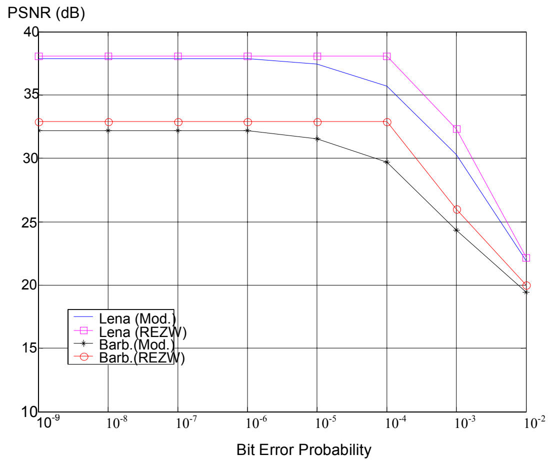

Fig. 10: PSNR versus BER with S=64 (REZW and modified algorithms).

(a) (b)

Fig. 11: Lena reconstruction with the modified algorithm:
(a) S=64, BER = 10^{-3}, PSNR=24.32 dB;
(b) S=1, BER = 10^{-3}, PSNR=16.01 dB.

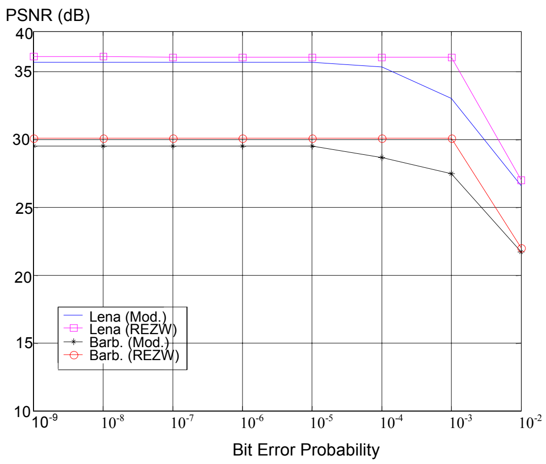

Fig. 12: PSNR versus BER with S=256 (REZW and modified algorithms).

(a) (b)

Fig. 13: Lena reconstruction with the modified algorithm:
(a) S=256, BER = 10^{-2}, PSNR=26.85 dB;
(b) S=1, BER = 10^{-2}, PSNR=13.36 dB.

REFERENCES

[1] C. D. Creusere, "A New Method of Robust Image Compression Based on the Embedded Zerotree Wavelet Algorithm", *IEEE Trans. Image Processing*, vol. 6, no 10, Oct. 1997.

[2] J. M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelets Coefficientes", *IEEE Trans. Signal Processing*, vol. 41, no. 12, Dec. 1993.

[3] A Said and W. A. Pearlman, "A New Fast and Efficient Image Codec based on Set Partitioning in Hierarchical Trees", *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, no. 3, June 1996.

[4] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression", Commun. ACM, v. 30, June 1987.

[5] M. Rabbani and P. W. Jones, Digital Image Compression Techniques. SPIE Optical Engineering Press, 1991.

[6] J. D. Villasenor, B. Belzer and J. Liao, "Wavelet Filter Evaluation for Image Compression", *IEEE Trans. Image Processing*, v. 4, no. 8, August 1995.