# An implementation of the EZW algorithm *

J. Oliver and M.P. Malumbres
Universidad Politécnica de Valencia. DISCA Department.
Camino de Vera 17, 46071 Valencia
E-mail: {joliver,mperez}@gap.upv.es

**Abstract**

Wavelet-based coders outperforms the DCT-based ones in terms of rate-distortion and subjective quality performance metrics. A lot of wavelet coders were proposed until now. Many of them were candidates for the JPEG 2000 still-image standard. However, the work is not finished and the research in this area still goes on.

In this paper, we present an implementation of a wavelet coder based on the Shapiro's EZW algorithm. Also, we show the importance of choosing an adequate filter bank in the wavelet decomposition stage. Several evaluation results show that our implementation achieves the performance of the original EZW algorithm presented by Shapiro [3].

*Keywords*: Image compression, wavelet transform, zero-tree quantizers.

## 1 Introduction

The early wavelet-based image coders [5, 1] were designed in order to exploit the ability of compacting energy on the typical wavelet decomposition by entropy coding its coefficients.

However, the properties of wavelet coefficients can be exploited more efficiently. In that sense, Shapiro developed a wavelet-based coder [3] that considerably improves the previous wavelet proposals. The coder, called Embedded Zero-tree Wavelet coder (EZW), is mainly based on two observations (a) the similarity between the same kind of sub-bands in a wavelet decomposition, and (b) a quantization based on a special kind of successive-approximations scheme that can be adjusted in order to get a specific bit rate. The coder includes an entropy encoder (typically an arithmetic coder) as its final stage.

Said and Pearlman [2] proposed a variation of EZW, called SPIHT (Set Partitioning In Hierarchical Trees). It is able to achieve better results than EZW without

*Work partially funded by *UPV*

taking into account the final arithmetic encoding stage. The improvements are mainly due to the way it groups the wavelet coefficients and how it stores the significant information.

A different approach to the previous algorithms is the one proposed by Tsai, Villasenor and Chen [4], known as the stack-run algorithm. This algorithm has a similar structure than JPEG coders. That is, after wavelet decomposition, the resulting coefficients are quantized using a classic quantization scheme. Then, quantized coefficients are entropy coded using a run-length encoder (RLE) and finally an arithmetic encoder is used.

In [6], a joint space-frequency quantization scheme was proposed. It uses a spatial quantization, like zero-tree, combined with a standard scalar quantizer. The idea is based in the fact that natural images are perfectly modeled by a linear combination of energy in both domains the frequency and space.

In section 2 details about the implementation of both the 2D DWT transform and EZW algorithm are given. In section 3 some performance evaluation results are shown. Finally, in section 4 some conclusions and future work are drawn.

## 2 The Embedded Zero-tree Wavelet Algorithm (EZW)

Most popular standards for image and video compression (MPEG, JPEG, H.261) are based on the Discrete Cosine Transform (DCT), a mathematical tool that transforms the signal domain from space to frequency.

Codecs based on the DCT present several drawbacks. Images are divided into regular small blocks that are processed separately, so when high compression rates are required, blocking artifacts appear in the reconstructed image, degrading the subjective quality considerably. On the other hand, the DCT uses a fixed orthonormal basis, the coefficients from the DCT, which could not be always the best choice.

The Discrete Wavelet Transform (DWT) is another mathematical tool that offers very good results when it is applied to image and video coding algorithms, improving significantly the performance of DCT-based codecs. We have implemented the wavelet decomposition using filter banks. Then, we perform symmetric extension when using symmetric filters, otherwise periodic extension is used.

In the first stage of the decomposition we apply the high and low-pass filters separately to both columns and rows. Thus, it divides the image into four sub-bands: one representing the low frequencies (LL) and which corresponds with the scaled version of the original image, and the others containing the horizontal (HL), vertical (LH) and diagonal (HH) high frequency bands. We have implemented a dyadic decomposition. This kind of wavelet decomposition obtains the next coarser scale of wavelet coefficients making a recursive decomposition of the LL sub-band until the desired level of decomposition is achieved.

Since all coefficients are perfectly allocable in a sub-band and at a specific position, this type of decomposition is said to present both spatial and frequency location. Moreover, the image is processed entirely and no block artifacts appear. Another advantage of the DWT with respect to the DCT, is the chance to choose the preferred filter (wavelet family), in this way you can select the orthonormal basis.

As we said in prior section, the Embbebed Zero-tree Wavelet (EZW) algorithm is considered the first really efficient wavelet coder. Its performance is based on the similarity between sub-bands and a successive-approximations scheme.

Coefficients in different sub-bands of the same type represent the same spatial location, in the sense that one coefficient in a scale corresponds with four in the prior level. This connection can be settled recursively with these four coefficients and its corresponding ones from the lower levels, so coefficient trees can be defined.

In natural images most energy tends to concentrate at coarser scales (higher levels of decomposition), then it can be expected that the nearer to the root node a coefficient is, the larger magnitudes it has. So if a node of a coefficient tree is lower than a threshold, it is likely that its descendent coefficients will be lower too. We can take profit from this fact, coding the sub-band coefficients by means of trees and successive-approximation, so that when a node and all its descendent coefficients are lower than a threshold, just a symbol is used to code that branch.

The successive-approximation can be implemented as a bit-plane encoder. The EZW algorithm is performed in several steps, with two fixed stages per step: the dominant pass and the subordinate pass. In Shapiro's paper [3] the description of the original EZW algorithm can be found. However, the algorithm specification is given with a mathematical outlook. In this paper we present how to implement it, showing some implementation details and their impact on the overall codec performance.

Consider we need $n$ bits to code the highest coefficient of the image (in absolute value). The first step will be focused on all the coefficients that need exactly $n$ bits to be coded (range from $2^{n-1}$ to $2^n-1$). In the dominant pass, the coefficients which falls (in absolute value) in this range are labeled as a significant positive/negative (sp/sn), according to its sign. These coefficients will no longer be processed in further dominant passes, but in subordinate passes. On the other hand, the rest of coefficients (those in the range $[0, 2^{(n-1)}[$) are labeled as zero-tree root (zr), if all its descendants also belong to this range, or as isolated zero (iz), if any descendant can be labeled as sp/sn. Notice that none descendant of a zero-tree root need to be labeled in this step, so we can code entire zero-trees with just one symbol. In the subordinate pass, the bit $n$ of those coefficients labeled as sp/sn in any prior step is coded. In the next step, the $n$ value is decreased in one so we focus now on the following least significative bit. Compression process finishes when a desired bit rate is reached. That is why this coder is so called embedded.

In the dominant pass four types of symbols need to be coded (sp, sn, zr, iz), whereas in the subordinate pass only two are needed (bit zero and bit one). Finally, an adaptive arithmetic encoder is used to get higher entropy compression.

## 3 Performance evaluation of our implementation.

Shapiro's EZW is a relatively complex algorithm, with several stages and parameters that can be optimized. In this section, we present different implementations alternatives that we found in the algorithm, some of them mentioned by Shapiro and others not, and evaluate its contribution to the performance of the EZW. Basically we have chosen an adequate filter bank and performed improvements on the EZW. Notice that when results are presented (in tables or curves), all options but those pointed out are assumed to be set to its default value, that are provided when an option is introduced in this paper (default image will be the standard Lena).

### 3.1 Choosing an adequate filter bank.

Choosing a good filter set is crucial to achieve a good compactness of the image in the LL band, thus we reduce the amount of nonzero coefficients and its magnitude, and therefore the image entropy. Shapiro uses an Adelson 9-tap QMF bank filter. With this filter, he obtains the results shown in Table 1 (row orig). Our implementation with the same image and filter (row Adel) shows similar results, so these results validate our implementation. But biorthogonal filters (B9/7) or Villasenor 10/18 (Vil), which compact the energy better, offer improved results. Daubechies 4 (D4), with only four taps, is the option that worst work. Similar results are obtained with the standard image Baboon, a monkey full of hair and high frequency details, but in this case Villasenor 10/18 achieves remarkably better performance, showing a great capability to efficiently decompose full-detailed images.

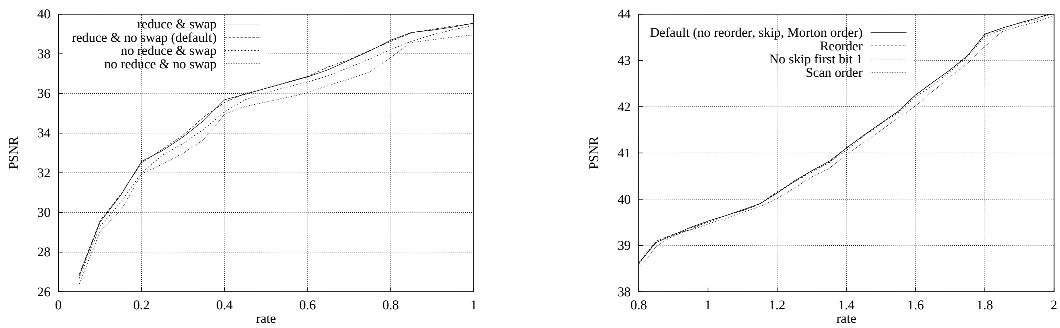
### 3.2 Evaluating the main EZW algorithm.

Some options can be established in the main algorithm. Curve "no reduce & no swap", in Figure 1.a, shows the different gradient existing between dominant and subordinate passes. This could mean that bits from subordinate passes are more valuable than those from dominant passes. Hence, performing a swap between the order of those stages could be a good idea. Curve "no reduce & swap" shows the

|  | PSNR Lena |  |  |  |  | PSNR Baboon |  |  |
|---|---|---|---|---|---|---|---|---|
| Bit Rate | Orig | Adel | Vil | B9/7 | D4 | Adel | Vil | B9/7 |
| 2 | n/a | 44.03 | 44.05 | 44.18 | 31.86 | 32.46 | 32.02 | 31.63 |
| 1 | 39.55 | 39.53 | 39.64 | 39.63 | 27.46 | 27.83 | 27.39 | 27.43 |
| 0.5 | 36.28 | 36.28 | 36.59 | 36.49 | 23.84 | 24.50 | 23.88 | 23.88 |
| 0.25 | 33.17 | 33.18 | 33.50 | 33.43 | 22.37 | 22.54 | 22.70 | 22.31 |

Table 1: Filter comparison with Lena and Baboon source images.

results when performing, in every pass, firstly the subordinate pass and then the dominant pass. In this way, when we run out of bits, no bit from the dominant pass is processed prior than one from the subordinate pass with the same threshold.




(a) (b)

Figure 1: EZW improvements: (a) Changing the passes order (b) Coefficient scanning order.

Another improvement is based on reducing the uncertainty interval at the decoder. The decoder must predict the bits which the coder could not send, because it finished its bit budget. It can assume that the rest of bits are 0 or maybe that all they are 1. But the best option seems to suppose that, for every coefficient, the more significative predicted bit is and the rest 0, so we would have a lower uncertainty interval. Curves "reduce" from Figure 1.a shows that by performing a swap, no significative improvements are achieved.

Other options on the EZW codec are shown in Figure 1.b. One of them is the scanning order of the coefficients in the dominant pass. We can see that a Morton order, that performs the scan in small groups, improves the performance of the algorithm, due to the best adaptivity achieved in the arithmetic encoder. Another improvement could be not to code the first bit of a coefficient, because the decoder can deduce it from the significative symbols in the dominant pass. The last option is to sort the coefficients in the subordinate pass, according to its magnitude, so bigger coefficients are coded before than smaller ones. Figure 1.b shows that, evaluating these options, only the scan order seems to be important (Morton order outperforms regular order).

## 4 Conclusions and future work

An implementation of a wavelet-based still-image coder was presented. We have proved its correctness and we have compare its performance with the one stated by the EZW authors.

We have shown that the performance of our EZW implementation is similar to the one achieved by Shapiro's EZW and if more efficient filter banks are used, the EZW performance increases.

As future work, we are planning to improve the EZW (or similar) performance by including a preprocessing wavelet coefficients stage, and reduce the temporal complexity of the EZW in order to implement a real-time Motion Wavelet video coder.

## References


[1] Antonini M, Barlaud M, Mathieu P, Daubechies I. Image coding using wavelet transform. IEEE Trans Image Processing, 1(2):205-220, 1992

[2] A. Said, A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. IEEE Transactions on circuits and systems for video technology, vol6, no3, June 1996

[3] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. on Signal Processing, vol41, n12, December 1993

[4] M.J. Tsai, J. Villasenor, F. Chen. Stack-run image coding. IEEE Trans. on Circuits and Systems for Video Technology, vol6, no10, pag 519-521, Oct. 1996

[5] J.W. Woods, S. O'Neil. Subband coding of images. IEEE Trans. Acoust., Speech, Signal Processing, vol.34, pp1278-1288, Oct. 1986

[6] Z. Xiong, K. Ramchandran, M.T. Orchard. Space-frequency quantization for wavelet image coding. IEEE Trans. on image processing, vol6, no5, pp677-693, May 1997