Tuning and optimizing the performance 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

During the last decade a lot of research and develop efforts have been made to design competitive still image coders for several kinds of applications. Some popular standards have emerged [7, 5] and the work is not finished yet [8].

We are interested in wavelet-based still-image coders. This kind of coders have a lot of interesting properties that make them very attractive for most applications. 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 optimized version of the Shapiro's EZW algorithm [12]. We have identified up to fifteen different implementation choices related with all the wavelet coder stages, from wavelet decomposition until the last entropy coding stage. After evaluating the proposed implementation options, we have chosen the ones that better performance results show, resulting in a wavelet coder that improves the original one in up to 0.8 dB at low bit rates (0.25 bpp) with the Lena standard image.

Keywords: Image compression, wavelet-based coders, EZW, performance evaluation.

1 Introduction

A wide variety of wavelet-based image compression schemes have been reported in the literature, ranging from simple entropy coding to more complex techniques such as vector quantization, adaptive transforms, tree encodings, edge-based coding, joint space-frequency coding schemes, etc. In this section we are going to introduce the most significative techniques that have been proposed for this kind of coders.

The early wavelet-based image coders [16, 1] were designed in order to exploit the ability of compacting energy on the typical wavelet decomposition. They used quantizers and variable-length entropy coders, showing little improvements

*Work partially funded by *UPV*

with respect to the popular DCT-based algorithms. In [4], some early wavelet coder proposals were compared with JPEG, concluding that wavelet coders obtain better results than JPEG only when low bit rates are used (bellow 0.25 bpp).

However, the properties of wavelet coefficients can be exploited more efficiently. In that sense, Shapiro [12] developed a wavelet-based coder that considerably improves the previous proposals. The coder, called Embedded Zero-tree Wavelet coder (EZW), is mainly based on two questions (a) the similarity between the same kind of sub-bands in a wavelet decomposition, and (b) a quantization based on a successive-approximation scheme that can be adjusted in order to get a specific bit rate. The own coder includes an entropy encoder (typically an adaptive arithmetic encoder) as its final stage.

Said and Pearlman [11] proposed a variation of EZW, called SPIHT (Set Partitioning In Hierarchical Trees). It is able to achieve better results than EZW, even without 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 [14], known as the stack-run algorithm. This algorithm has a similar structure than JPEG coders. That is, after wavelet decomposition, the wavelet 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. The originality of this algorithm resides on the use of a symbol set that allows an efficient storage of each pair of values supplied by the RLE stage. The stack-run authors state that it achieves better results than those obtained with EZW in a range from 1 to 2 dB [14].

In [17], a joint space-frequency quantization scheme was proposed. It uses a spatial quantization, like zero-tree, in combination with a standard scalar quantizer. The idea is based in the fact that natural images are perfectly modeled by a lineal combination of compacted energy in both domains the frequency and space. In [18] a wavelet packet coder is proposed using the principles of the joint space-frequency quantization.

Finally, in [6], another wavelet coder proposal is presented. It is based on the use of a scalar quantizer and the construction of a coefficient map with a trellis state coder based on the Viterbbi algorithm [15]. By using the Trellis Coded Quantization (TCQ), a very good image quality is obtained with a relatively low complexity. In fact, this was the algorithm that best results achieved in the Sidney JPEG 2000 meeting [13].

In general, when a new wavelet coder is proposed, the associated algorithm use to need a set of optimizations in order to be competitive in performance. These optimizations are mainly related to the final algorithm implementation. So, it is very important to evaluate the different implementation choices before the final version of the wavelet coder, in order to achieve the highest performance results.

In this paper, we are going to implement a version of EZW wavelet still image coder based on the one proposed by Shapiro [12]. Then, we are going to test different implementation alternatives in order to show their impact on the overall coder performance. Also, we will test the performance results obtained with our implementation comparing them with those published by the EZW author in order to check its correctness.

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

2 2D DWT wavelet transform and the EZW algorithm

Coders based on the Discrete Cosine Transform (DCT), like MPEG, JPEG and H.261, present several drawbacks. Images are divided into regular small blocks that are processed separately, so when high compression rates are required, blocking artefacts appear in the reconstructed image, degrading the objective and subjective quality considerably. On the other hand, the DCT uses a fixed orthonormal basis (the DCT) which seems not to be always the best choice.

The Discrete Wavelet Transform (DWT) is another mathematical tool that relies on a recently developed theory [10, 3, 9], which has resulted of interest on several fields of engineering and computer science. In particular, it offers very good results when it is applied to image and video coding algorithms, improving significantly the performance of DCT-based coders. To implement wavelet decomposition, filter banks are commonly used. In this case, symmetric extension is performed when using symmetric filters, otherwise periodic extension is used.

In the first decomposition level, high and low-pass filters are applied to both columns and rows by separate. 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 implemented a dyadic decomposition, where the next coarser scale of wavelet coefficients are obtained making a recursive decomposition of the LL sub-band, until the desired decomposition level is achieved.

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

As we said in prior section, the Embedded 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-approximation scheme. Some of the coefficients from different sub-bands represent the same spatial location, in the sense that one coefficient in a scale corresponds with four in the previous one. 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 from the root node a coefficient is, the larger magnitudes it has. So if a node of a coefficient tree is lower than a threshold, its descendent coefficients will probably be lower as well. 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 that a threshold, just a symbol is used to code that branch.

The EZW algorithm is performed in several steps, with two fixed stages per step: the dominant pass and the subordinate pass. Successive-approximation can be implemented as a bit-plane encoder, so that the method can be outlined as follow (notice that an implementation outlook is taken).

Consider we need n bits to code the highest coefficient of the image (in absolute value). The first step will be focused on all those 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 fall (in absolute value) in this range are labeled as 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 the 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, and 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. More details about the EZW algorithm can be found in the original paper [12].

3 Tuning the EZW algorithm.

Shapiro's EZW is a relatively complex algorithm, with several stages and parameters that can be optimized. We have implemented the EZW algorithm in order to get its best performance by tuning the coder. So, 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. All these options can be grouped in four categories, (a) filters, (b) coefficient preprocessing, (c) improvements on the EZW, and (d) improvements related to the adaptive arithmetic encoder.

Notice that when results are presented (in tables or curves), all configuration options are assumed to be set to its default value with the exception of those explicitly mentioned (the default image will be the standard Lena).

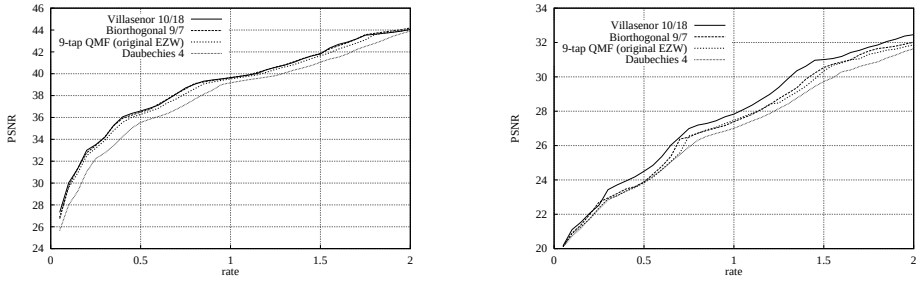
3.1 Choosing the best filters.

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 and the standard image Lena, he obtains the results shown in Table 1 (Orig column). Our implementation, with the same image and filter (Adel column), throws similar results. We think that

these results validate our implementation. However, biorthogonal filters, B9/7 and Villasenor 10/18 (Vil), which make a better energy compactation, provide better results. Daubechies 4-tap filter (D4) gets poorer compactness and hence lower PSNR values. Similar results are obtained with the standard image Baboon. However, with this image, Villasenor 10/18 achieves remarkably better performance, showing a great capability to efficiently decompose full-detailed images.

	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	43.90	31.86	32.46	32.02
1	39.55	39.53	39.64	39.63	39.17	27.46	27.83	27.39
0.5	36.28	36.28	36.59	36.49	35.54	23.84	24.50	23.88
0.25	33.17	33.18	33.50	33.43	32.23	22.37	22.54	22.70

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



(a) (b)

Figure 1: Evaluating different filters for (a) Lena, and (b) Baboon.

All these results are also shown in Figure 1. Notice that both Lena and Baboon images have been downloaded from [2].

Another important aspect in wavelet processing is the number of decomposition levels performed. It mainly depends on the image size and the number of filter taps. As we can see in Figure 2, for a 512x512 image and a 9-tap QMF filter, it is highly interesting decomposing the LL sub-band up to four times. However, less improvements are attained with more than four decomposition levels. By default, we, as Shapiro does, will perform a six level dyadic decomposition with Adelson 9-tap QMF filter on the 512x512 standard image Lena.

3.2 Coefficient preprocessing.

Shapiro proposes that the image mean can be removed before the EZW algorithm is applied. Figure 3 shows the effect of this idea. It was performed in two different manners: (a) removing simultaneously the mean of all the bands, and (b) removing it only from the LL band (similar to remove the original image mean). As wavelet sub-bands are expected to be zero mean, trying to remove the mean of these bands does not seem to be a good idea, as Figure 3 shows.

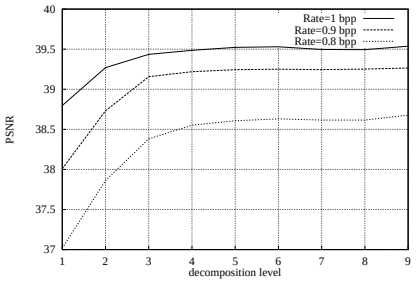


Figure 2: Impact on performance of the number of decomposition levels.

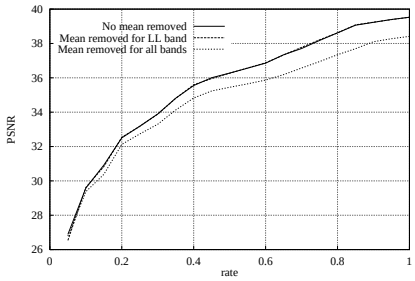


Figure 3: Mean value removing option.

On the other hand, the effect of removing the mean of the LL band is nearly negligible (default value is no mean removed).

An important effect that appears in all the Rate/Distortion curves based on the EZW algorithm is its scalloped aspect; it can be easily noticed the peaks in the performance which correspond with the end of a full EZW iteration. That is due to its embedded nature: the EZW presents its best performance when the algorithm finishes its bit budget just at the end of a subordinate pass. A uniform quantization of the coefficients can move these peaks along different rates. This effect is shown in Figure 4.a, where at established bit rates, different quantization factors (q values) have been used. In this case, with 1 bpp, best performance is obtained at $q = 0.2 * k$ being k integer. These peaks are shifted to the right when the bit rate decreases, until a 0.5 bpp value is reached. Then, a full pass is completed and peaks repeat again at $q = 0.2 * k$. Figure 4.b shows that with $q = 0.8$ peaks are achieved at $1/2^n$ rates.

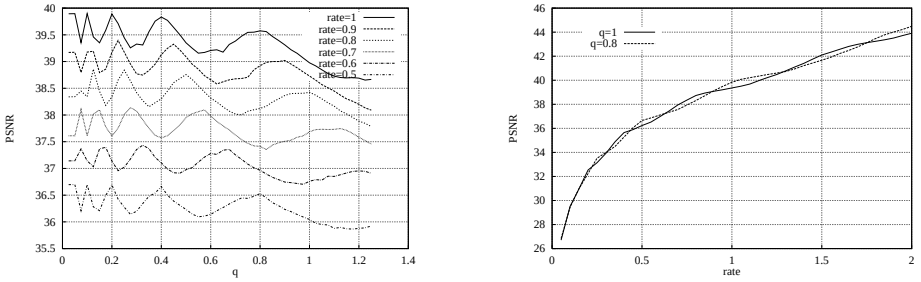
Therefore, with suitable values of q, results from Table 1 can be significantly improved. These results are shown in Table 2 (default value: no uniform quantization used).

	PSNR 9-tap QMF filter			PSNR Villasenor 10/18 filter	
Bit Rate	Orig	No quant.	q = 0.8	No quant.	q = 0.4
2	n/a	44.03	44.49	44.05	44.79
1	39.55	39.53	39.83	39.64	40.20
0.5	36.28	36.28	36.87	36.59	37.04
0.25	33.17	33.18	33.52	33.50	33.97

Table 2: Optimized results improving the quantization of coefficients.

3.3 Improvements on the main EZW algorithm.

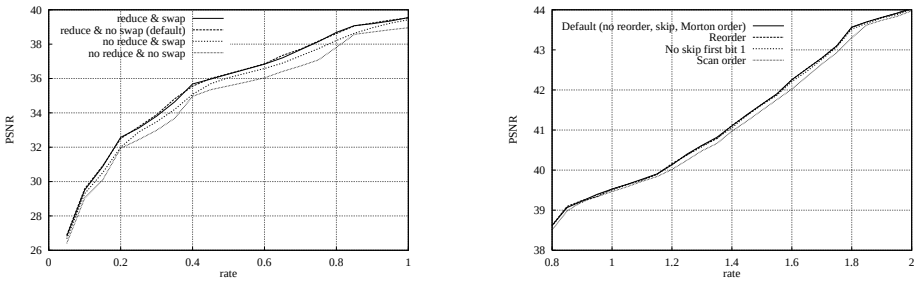
Some options can be established in the main algorithm. Curve "no reduce & no swap", in Figure 5.a, shows the different gradient existing between dominant and subordinate passes. This could mean that bits from subordinate passes are more valuable than those from dominant passes. Hence, performing a swap between the order of those stages could be a good idea. Curve "no reduce & swap" shows the results of performing firstly the subordinate pass and then the



(a) (b)

Figure 4: Uniform quantization option (a) PSNR for different quantization factors at constant rates, and (b) Rate/Distortion curves at different quantizations.

dominant pass for every EZW iteration. 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 5: EZW improvements: (a) Swapping dominat and subordinate passes (b) Coefficient scanning order.

Another improvement consist on reducing the uncertainty interval at the decoder. The decoder must predict the bits which the coder could not send because he finished its bit budget. It can assume that the rest of bits are 0, or maybe that all are 1. But the best option seems to suppose that, for every coefficient, the more significative predicted bit is 1 and the rest 0, so we would have a lower uncertainty interval and, as consequence, less error. Curves "reduce" from Figure 5.a shows the improvement of this action, and how performing a swap is not actually significative when the uncertainty interval is already reduced.

Other options on the EZW coder are shown in Figure 5.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 for the decoder, so bigger coefficients are coded before than smaller ones. Figure 5.b shows that, evaluating these options, only the scan order seems to be important (Morton order is better than regular order).

3.4 Improvements on the arithmetic encoder.

Several actions can be tackled on the adaptive encoder. First, four histograms can be used in the dominant pass, depending on the significance of the previously coded coefficient and its parent coefficient in the current pass. Second, all the histograms can be restarted at the end of a full pass, to improve the adaptivity. Finally, as last sub-bands do not have offspring, we do not need to use a four-symbol alphabet for these bands, and another arithmetic encoder (without the isolated-zero symbol) can be used. By default, all these improvements are tackled. Figure 6 shows the contribution of every option to the performance of the algorithm (removing them one by one). Only the first option seems to be of interest (notice the smaller scale of this graph).

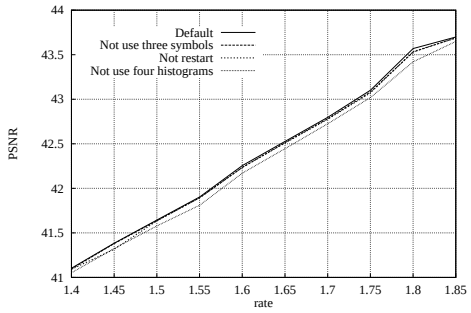


Figure 6: Arithmetic encoder evaluation.

It is also important the maximum frequency count in the adaptive arithmetic code and how many the histogram is increased with every symbol (default values, 512 and 6 respectively).

Notice that although most results have been presented for the Adelson 9-tap QMF bank filter (because it was the filter used by Shapiro), the rest of the filters from subsection 3.1 behave similarly.

4 Conclusions and future work

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

The main contribution of this paper resides in a deep study of the EZW implementation, evaluating many alternatives (up to 15) in the different coder stages. Also, we have shown that it is possible to get better results (around 0.4 dB) than those published by authors under the same conditions, and if more efficient filter banks are used, the EZW performance significantly increases (up to 0.8 dB).

As future work, we are planning to improve the EZW performance by adding a new stage behind the wavelet decomposition, in order to reduce the entropy of the wavelet coefficients before the quantization stage.

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] CityU-UPL. Image Database. http://www.image.cityu.edu.hk/imagedb/

[3] I. Daubechies. Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992

[4] M. L. Hilton, B. D. Jawerth, A. Sengupta. Compressing still and moving images with wavelets. Multimedia Systems, vol2, n3, 1994

[5] Coded representation of picture and audio information: progressive bi-level image compression. ISO/IEC 11544, ISO 1993.

[6] R.L. Joshi, V.J. Crump, T.R. Fischer. Image subband coding using arithmetic coded trellis coded quantization. IEEE Trans. on circuits and systems for video technology, vol5, no6, December 1995

[7] Joint Photographic Expert Group. Digital Compression and Coding of Continuous Tone Images (Part 1: Requirements and Guidelines). ISO/IEC 10918-1, 1992

[8] D. Lee. Jpeg 2000: Developments in still-image coding system. In slides from presentation at EUSIPCO '98 (http://eurostill.epfl.ch/eusipco-j2k-pres/sld001.htm).

[9] S. Mallat. A Theory for Multiresolution Signal Decomposition. IEEE Trans. Pattern Anal. Mach. Intel., Vol. 11, pp. 669-718, July 1989

[10] Y. Meyer. Principe D'incertitude. Bases Hilbertiennes et Algèbres D'Opérateurs, Séminaire Bourbaki, No. 662, 1985-1986

[11] 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

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

[13] O. Strmne. On the applicability of wavelet transforms to image and video compression. Thesis, Department of Computer Science, University of Strathclyde, Feb. 1999

[14] 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

[15] Jr.G.D. Forney. The viterbi algorithm. Proceedings of the IEEE (invited paper), 61, pag 268-278, March 1973

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

[17] 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

[18] Z. Xiong, K. Ramchandran, M.T. Orchard. Wavelet packet image coding using space-frequency quantization. IEEE Trans. on image processing, vol7, no6, pp892-898, June 1998