SIGNED BINARY DIGIT REPRESENTATION TO SIMPLIFY 3D-EZW

Emmanuel Christophe[†] [*], *Pierre Duhamel*[‡] *and Corinne Mailhes*[*]

[†] CNES, BPI 1219, 18 avenue Edouard Belin, 31401 Toulouse Cedex 9, France
[*] TeSA/IRIT, 14 port St Etienne, 31000 Toulouse, France
[‡] CNRS/LSS, Supelec Plateau de Moulon, 91192 Gif-sur-Yvette Cedex, France

e.christophe@ieee.org, pierre.duhamel@lss.supelec.fr, corinne.mailhes@enseeiht.fr

ABSTRACT

Zerotree based coders have shown a good ability to be successfully adapted to 3D image coding. This paper focuses on the adaptation of EZW for the compression of hyperspectral images with reduced complexity. The subordinate pass is removed so that the location of significant coefficients does not need to be kept in memory. To compensate the quality loss due to this removal, a signed binary digit representation is used to increase the efficiency of zerotrees. Contextual arithmetic coding with very limited contexts is also used. Finally, we show that this simplified version of 3D-EZW performs almost as well as the original one.

Keywords: Image coding, EZW, Signed binary digit representation.

1. INTRODUCTION

In the context of image compression, wavelet transform has shown a good ability to decorrelate image pixels. Efficient techniques to code these wavelet coefficients have been proposed. EZW successfully makes use of the relation of wavelet coefficients in zerotrees [1]. EBCOT, the coder for JPEG 2000 focuses on the neighborhood of each coefficient using contextual arithmetic coding [2]. In this standard, a total of 18 different contexts are used according to the value of neighboring coefficients.

This paper considers the zerotree-based compression techniques in conjunction with signed-binary representations and arithmetic coding, particularly in the context of 3D image encoding. The 3D images used here are hyperspectral images from the JPL/NASA airborne sensor AVIRIS (http://aviris.jpl.nasa.gov). The same methods remain valid for medical images as magnetic resonance (MR) or computed tomography (CT) which are also formed of several slices. Hyperspectral images involve observing the same scene at different wavelengths. Typically, each image pixel is represented by hundreds of values, corresponding to various wavelengths. These values correspond to a sampling of the continuous light spectrum emitted by the pixel. This sampling of the spectrum at very high resolution enables pixel identification from its physical characteristics. Hyperspectral images can be seen as three dimensional data where two dimensions correspond to the spatial scene observed and the third dimension to the light spectrum of the pixel.

The highlight of this paper is not on the choice of the wavelet form, thus the popular 9/7 wavelet is chosen for lossy compression and the 5/3 for lossless compression. The decomposition is first done for each spatial plane in a Mallat's decomposition scheme and then for each spectrum (the third dimension) as this decomposition was shown to be nearly optimal [3, 4] (Figure 1).

A short description of the EZW algorithm [1] is given in section 2. A drawback coming from the use of the subordinate pass is explained. In section 3, successive improvements are described to finally reach a version of EZW performing almost as efficient as the original one in terms of rate/distortion tradeoff, while offering simpler implementation, since it does not require the use of the subordinate pass. The progression of the proposed improvements is highlighted on a particular hyperspectral image in terms of PSNR and MSE. Final results on different hyperspectral images are given in section 4 showing that the proposed simplified EZW algorithm leads to positive conclusions on the use of signed binary digit representation within compression algorithms.

2. EZW ALGORITHM

2.1. Zero-tree coding

The original EZW algorithm is described in [1] and a 3D adaptation is provided in [5]. The idea is to perform successive encoding of the different bitplanes of the wavelet coefficients. For each bitplane, a tree structure is defined. The coding algorithm involves two steps. The first step is called the *dominant pass* (or *significant pass*). Each bit is encoded using one of the four symbols: ZTR for zerotree root (the current coefficient and all its descendants in the tree are non-significant), IZ for isolated zero (at least one descendant is significant), POS or NEG (the current coefficient is significant and either positive or negative). The second step, called *subordinate pass* (or *refinement pass*), concerns the coefficients previously declared as significant and encodes the corresponding bit in the current bitplane. In the 2D-EZW case, the tree structure of the coefficients is induced by the wavelet decomposition. However, in the 3D case, there are several possibilities to define the relationship between coefficients. For example, in [6], only the spatial link between pixels is used. However, it has been shown in [4] that the most efficient tree structure for EZW uses both spectral and spatial link. Therefore, the overlapping tree structure illustrated on figure 1 is used. The EZW algorithm used in this paper for reference is programmed using ANSI C and produces similar performance to the original paper. The wavelet transform and the arithmetic coder are performed using the latest version of the QccPack library [7].

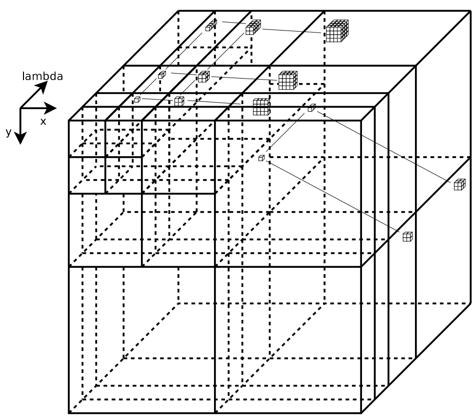

Fig. 1. Illustration of the wavelet decomposition and tree structure.

2.2. One drawback

A drawback of EZW is the memory required to store the coefficients already classified as significant. These coefficients are processed during the subordinate pass and should not be processed during the dominant pass. One bit of memory at least is required for every coefficient of the image. One solution to remove the need for this memory is to remove the subordinate pass. In this situation only the dominant pass is processed for each bitplane which also eases the dependancy between bitplanes coding and makes multithreading possible. Coefficients are considered as insignificant if the bit in the bitplane is 0 and significant otherwise. However this simple change causes a loss in performance, since the average length of the codes is increased. This results in a loss of more than 2 dB PSNR (Table 1).

The next part focuses on providing coding of bitplanes without any significant losses i.e. increasing the efficiency of the dominant pass for every bitplane.

Rate	3D-EZW		Without subordinate pass	
(bpppb)	MSE	PSNR	MSE	PSNR
1.0	106.15	76.07	193.73	73.46
0.5	445.22	69.84	685.49	67.97

Table 1. Effect of removing the subordinate pass. Results are for Moffett Field AVIRIS image (image details in section 4).

3. IMPROVEMENT

3.1. Increasing the number of zeros

Zerotrees high performance is due to their ability to encode a large number of zero coefficients using only one symbol. However, if all bitplanes are processed with a dominant pass, when going down the bitplanes, the probability of having 0 on a lower bitplane for a given coefficient tends to be close to 0.5. Moreover, these zeros tend to be randomly distributed, thus hurting the capabilities of zerotrees to efficiently gather these coefficients.

One strategy to increase the compression capability is to increase the proportion of zeros in each bitplane. A possible solution is the use of signed digit representation. A signed binary digit representation of a number n is a sequence of digits $a = (\ldots, a_2, a_1, a_0)$ with $a_i \in \{-1, 0, 1\}$ such as $n = \sum_{i=0}^{\infty} a_i 2^i$.

The number 119, for example in classical binary notation is $(0,1,1,1,0,1,1,1)$ as it is equal to $1*2^6+1*2^5+1*2^4+1*2^2+1*2^1+1*2^0$. If a ternary alphabet $\{-1,0,1\}$ is used instead of a binary one, the number 119 can be noted $(1,0,0,0,-1,0,0,-1)$ as it is equal to $1*2^7-1*2^3-1*2^0$ thus increasing the proportion of zeros.

The signed binary digit representation for a given number is not unique. Generally, the interest is in representations which have a maximum of 0s. A simple algorithm is proposed in [8] to convert standard binary to signed binary digit representation. This algorithm leads to the non-adjacent form (NAF) where a non zero digit is necessarily followed by a 0. This form is widely used for fast exponentation in cryptographic systems.

To measure the efficiency in increasing the amount of zero coefficients, we compute the proportion of zeros after the first significant bit. For the wavelet transform of an hyperspectral image of $256 \times 256 \times 224$ coefficients, the average number of bits after the first significant bit, the number of zero bits after this first significant bit and the proportion of zero bits versus non zero are detailed in table 2. As shown in this table, the binary-signed digit representation managed to increase significantly the proportion of zeros for lower bitplanes: more than 60% of 0s against 50% before.

EZW is implemented using binary signed digit representation (NAF) and each bitplane is processed separately with a dominant pass. However, even if we can observe a gain of 1 dB using any of the signed binary digit representation (Table 3), this improvement is not sufficient to recover from the

Notation	Average num. of bits after the first sig.	Number of zero bits	Proportion of zero bits
Binary	2.72	20 490 955	51.28%
NAF	3.12	29 263 791	63.83%

Table 2. Zero bit proportion after the first significant bit.

Rate	Binary		NAF	
(bpppb)	MSE	PSNR	MSE	PSNR
1.0	193.73	73.46	149.07	74.60
0.5	685.49	67.97	549.56	68.93

Table 3. EZW with independent processing of each bitplane (without subordinate pass).

loss due to the removal of the subordinate pass. We do not reach the original performance of table 1.

3.2. Using the local dependencies

However, this strategy does not take into account the values of the neighboring coefficients or bits, even if the number representation now provides useful information. First consider the coefficient at the same location on the previous bitplane. In the case of the NAF, this dependency is easy to take into account: if this coefficient on the previous bitplane is 1 or -1, we know that the coefficient on the current bitplane is 0.

Consider also the values of the neighboring coefficients in the same bitplane. A simple way to take them into account is to use contextual arithmetic coding. Only three coefficients on the same bit plane are considered. These coefficients are those preceding the current pixel in the three directions of the hyperspectral wavelet cube.

Denote η_s, η_l and η_b, the preceding coefficients on the three directions. Thus, we have:

- $\eta_s(i,j,k)$ the value at position $(i-1,j,k)$,
- $\eta_l(i,j,k)$ the value at position $(i,j-1,k)$,
- $\eta_b(i,j,k)$ the value at position $(i,j,k-1)$.

Since the bitplanes are considered separately, η_s, η_l and η_b are within the set $\{-1,0,+1\}$. We consider the valuation function for the neighborhood η defined as $\eta = \eta_s + 3\eta_l + 9\eta_b$. This function is a bijection between all possible neighboorhoods and the integers between -13 and 13.

We can plot the probability of a given coefficient at location (i,j,k) to take the values -1, 0 or 1 according to the neighborhood values represented by the function η. The probability curves are presented on figure 2. These probabilities are computed for the $256 \times 256 \times 224$ Moffett Field image on all bitplanes. Thus, several millions of data are taken into account. From these curves, we can see that one neighborhood clearly differs in term of probability compared to the others, when $\eta = 0$ i.e. $\eta_s = \eta_l = \eta_b = 0$. With this neighborhood, the probability to have a 0 for the current coefficient is very high.

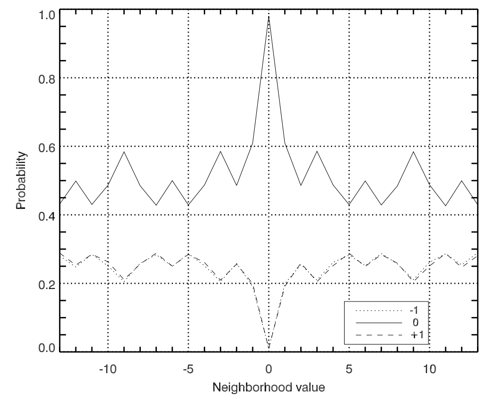

Fig. 2. Probability of having value -1, 0 or 1 for the current coefficient according to the neighborhood value with the NAF form. The 27 possible neighborhoods are presented on abscissa according to the value of η.

Rate	Non contextual		Contextual	
(bpppb)	MSE	PSNR	MSE	PSNR
1.0	149.07	74.60	121.38	75.49
0.5	549.56	68.93	457.77	69.72

Table 4. EZW with independent processing of each bitplane NAF with and without contextual coding.

As the differences between other cases remain low, the context for the arithmetic coder will be separated in two cases: $\eta = 0$ and $\eta \neq 0$.

For the NAF, a non zero value for a coefficient at a certain bitplane will be followed by one 0 at the next bitplane (hence the reason for the denomination *non-adjacent form*). In this case, it is not necessary to give any output for this 0. Note that this strategy does not require to store the locations of these *ones* in the previous bit plane : the following *zero* is generated together with the *one* at the decoder.

This latest version of EZW without subordinate pass using NAF and contextual arithmetic coding is referred to as 3D-EZW-NAF. First results of this 3D-EZW-NAF coding are given in table 4.

4. RESULTS

3D-EZW-NAF is applied to a $256 \times 256 \times 224$ extract of the scene 3 of f970620t01p02_r03 run from AVIRIS sensor on Moffett Field site. This part is from the lower right angle of the scene and is the most difficult part of the image to compress (urban area). Another image is a $256 \times 256 \times 224$ extract of the scene 1 of f970403t01p02_r03 AVIRIS run over Jasper site. This part is from the top left angle of the scene. These two images are in radiance and correspond to the signal received by the airborne sensor. These two scenes are widely available and popular in experiments on hyperspectral image

Rate (bpppb)	3D-EZW MSE	3D-EZW PSNR	3D-EZW-NAF MSE	3D-EZW-NAF PSNR
Moffett 1.0	106.15	76.07	121.38	75.49
Moffett 0.5	445.22	69.84	457.77	69.72
Moffett 0.25	1407.34	64.85	1514.81	64.53
Jasper 1.0	40.56	80.25	43.49	79.95
Jasper 0.5	139.31	74.89	140.76	74.84
Jasper 0.25	391.31	70.40	411.75	70.18

Table 5. Performance comparison between 3D-EZW and the simplified version using NAF on AVIRIS image Moffett and Jasper.

Rate (bpppb)	3D-EZW MSE	3D-EZW PSNR	3D-EZW-NAF with subordinate pass MSE	3D-EZW-NAF with subordinate pass PSNR
1.0	106.15	76.07	112.42	75.82
0.5	445.22	69.84	427.36	70.02
0.25	1407.34	64.85	1399.51	64.87

Table 6. Comparison between 3D-EZW and 3D-EZW-NAF with subordinate pass.

coding.

Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) for different rates are given in table 5 for Moffett Field and Jasper images. The rate is given in bit per pixel per band (bpppb), the PSNR in dB is calculated as $\text{PSNR} = 10\log_{10}(2^{16}-1)^2/\text{MSE}$ [9].

The use of the NAF enables us to recover more than 2 dB from the loss resulting from the removal of the subordinate pass. The performance of EZW without subordinate pass comes very close to the original EZW without the need to keep the list of significant coefficients in memory (saving 1 bit of memory per coefficient during the compression: 14.7 Mbits), and making the hardware implementation easier. The full rate-distortion curve is presented on figure 3 for the Moffett image.

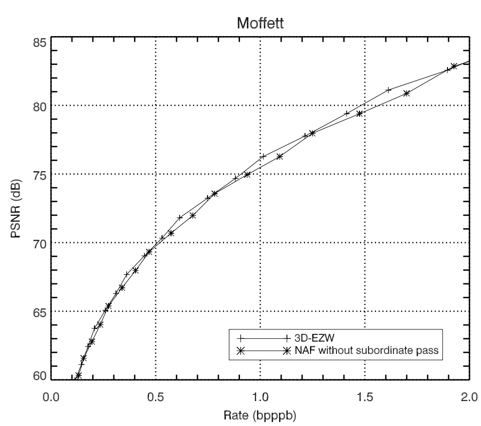

Fig. 3. Comparison of compression performance between 3D-EZW and the 3D-EZW-NAF version without subordinate pass.

Even if the original purpose was to remove the subordinate pass to ease the memory requirements, we can check the performance of the signed binary digit representation with the subordinate pass (Table 6). The quality obtained is very close to the reference version of EZW and even exceeds it for some rates (0.5 bpppb and 0.25 bpppb).

5. CONCLUSION

Signed binary digit representations, particularly the NAF, have shown a good ability to compensate for the removal of the subordinate pass in the EZW algorithm. This compensation is not as significant as expected but its use in conjunction with contextual arithmetic coding enables a simplified algorithm to perform almost as well as the original one. On top of the reduction in memory requirements, this new version of the algorithm allows an easy multithread implementation.

The use of signed binary digits is typically to enable fast exponentiation and it is not common to use it to increase the proportion of zeros. Binary signed digit representations have shown a good ability for that and such a use could be applied to other compression algorithms.

6. ACKNOWLEDGMENTS

This work has been carried out under the financial support of *Centre National d'Études Spatiales* (CNES), *Office National d'Études et de Recherches Aérospatiales* (ONERA) and *Alcatel Alenia Space*.

7. REFERENCES

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

[2] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, pp. 1158–1170, July 2000.

[3] E. Christophe, C. Mailhes, and P. Duhamel, "Best anisotropic 3-D wavelet decomposition in a rate-distortion sense," in *International Conference on Acoustics, Speech, and Signal Processing. ICASSP'06*, vol. 2, pp. II–17 – II–20, IEEE, May 2006.

[4] E. Christophe, C. Mailhes, and P. Duhamel, "Hyperspectral image compression: Adapting SPIHT and EZW to anisotropic 3D wavelet coding," *IEEE Transactions on Image Processing*, 2006. (submitted).

[5] A. Bilgin, G. Zweig, and M. W. Marcellin, "Three-dimensional image compression with integer wavelet transforms," *Applied optics*, vol. 39, pp. 1799–1814, Apr. 2000.

[6] C. He, J. Dong, and Y. F. Zheng, "Optimal 3-D coefficient tree structure for 3-D wavelet video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 10, pp. 961–972, 2003.

[7] J. E. Fowler, "QccPack - Quantization, Compression, and Coding Library." http://qccpack.sourceforge.net/, 2006.

[8] S. Arno and F. S. Wheeler, "Signed digit representations of minimal hamming weight," *IEEE Transactions on Computers*, vol. 42, pp. 1007–1010, Aug. 1993.

[9] E. Christophe, D. Léger, and C. Mailhes, "Quality criteria benchmark for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, pp. 2103–2114, Sept. 2005.