Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2007, Article ID 54679, 7 pages
doi:10.1155/2007/54679

Research Article

Adaptation of Zerotrees Using Signed Binary Digit Representations for 3D Image Coding

Emmanuel Christophe,[1,2] Pierre Duhamel,[3] and Corinne Mailhes[2]

[1] *CNES, BPI 1219, 18 avenue Edourad Belin, 31401 Toulouse cedex 9, France*
[2] *TeSA / IRIT, 14 port St Etienne, 31000 Toulouse, France*
[3] *CNRS / LSS, Supelec Plateau de Moulon, 91192 Gif-sur-Yvette, France*

Received 15 August 2006; Revised 16 December 2006; Accepted 18 December 2006

Recommended by James E. Fowler

Zerotrees of wavelet coefficients have shown a good adaptability for the compression of three-dimensional images. EZW, the original algorithm using zerotree, shows good performance and was successfully adapted to 3D image compression. This paper focuses on the adaptation of EZW for the compression of hyperspectral images. The subordinate pass is suppressed to remove the necessity to keep the significant pixels in memory. To compensate the loss due to this removal, signed binary digit representations are 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.

1. INTRODUCTION

Since the publication of the Grossmann and Morlet paper [1], theory and applications concerning wavelets have improved. Theory on wavelets was progressively refined in several papers (e.g., [2, 3]). Originally, applications concerned mostly the data analysis field and more precisely in the time-scale analysis. However, their efficiency to represent complex signals with a limited number of generating functions raised an interest for image coding [4].

Research in wavelet-based image coding began focusing on the search for the most efficient wavelet form to represent the data as in [4] together with the most efficient decomposition [5]. The quasiorthogonal 9/7 wavelet for lossy compression and the 5/3 wavelet for lossless compression with a multiresolution decomposition exhibit good results for a wide range of natural images. Thus, these specifications were adopted in the latest still image compression standard: JPEG 2000 [6].

Efficient techniques to code these wavelet coefficients were then defined. EZW successfully made use of the relation of wavelet coefficients in zerotrees [7], a technique which was further refined with SPIHT [8]. EBCOT, the coder for JPEG 2000 focuses on the neighborhood of each coefficient using contextual arithmetic coding [9]. In this standard, a total of 18 different contexts are used according to the value of neighboring coefficients.

This paper looks at the zerotree-based compression techniques and improves them with the use of 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. The same methods can be applied to medical images as magnetic resonance (MR) or computed tomography (CT) which are also formed of several slices. Hyperspectral image involves observing the same scene at different wavelengths (Figure 1). 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 allows pixel identification (materials, mineral and gases, etc.). 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 for the pixel.

The highlight of this paper is not on 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

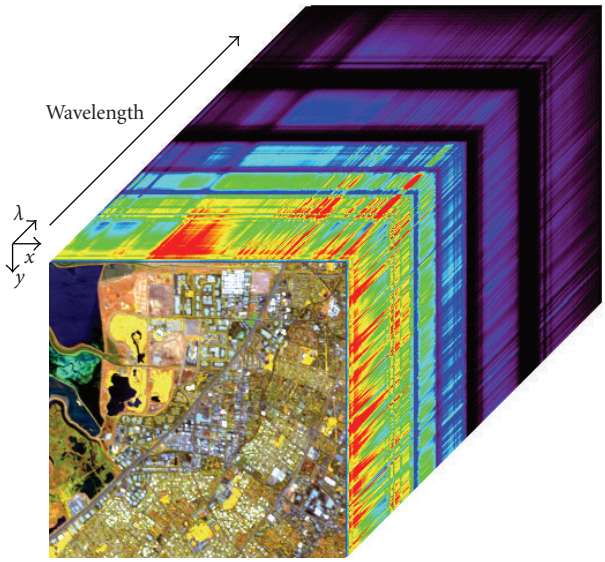


Figure 1: Example of a hyperspectral data cube (Moffett Field by AVIRIS): the front of the cube is a color composite of three spectral bands while the other sides display the spectra of the side pixels.

this decomposition was shown to be nearly optimal [10, 11]. The adaptation is done on EZW as the progression between coefficients is in raster order and less dependant of the data which is easier to adapt compared to SPIHT.

Details on the EZW algorithm are given in Section 2. Our EZW reference version is validated against results from other papers both in the 2D and 3D cases. This reference version, described in [11], exhibits slightly higher performances than those of the original EZW paper [7]. One 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 well as the original one without the use of the subordinate pass. Several results are given in this section to show the progression of the improvements. All the details concerning the measures: distortion, bit rate are given later in Section 4, but all are common.

2. EZW ALGORITHM

2.1. *Zerotree coding*

At the time of its publication, embedded zerotree coding of wavelet coefficients (EZW) from Shapiro [7] produced state-of-the-art compression performance at a modest level of complexity. This algorithm has some properties which make it particularly attractive in the context of 3D image compression. It produces an embedded bitstream: every prefix of a bitstream produced by EZW is a valid EZW bitstream, leading to a decompressed image with a lower quality. This algorithm manages to achieve this at a relatively modest level of complexity.

To ensure that the property of embedded bitstream is adhered to, the algorithm uses bitplane encoding of coefficients.

For each bitplane:

(i) dominant pass: For each coefficient which has not been found as significant before, output one of the symbol ZTR (zerotree: all coefficients corresponding to the same location in higher frequency subbands are insignificant), IZ (isolated zero: the coefficient in not significant and at least one coefficient corresponding to the same location in higher frequency subbands is significant), POS (positive significant coefficient) or NEG (negative significant coefficient);

(ii) subordinate pass: output one bit for all coefficients declared as significant before the current bitplane. This bit corresponds to the value of the coefficient in the current bitplane.

Algorithm 1: EZW.

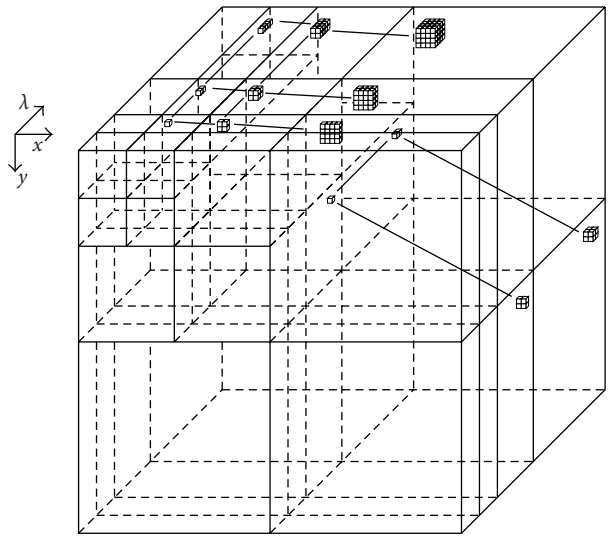


Figure 2: Illustration of the wavelet decomposition and tree structure.

As explained by Shapiro, the costly part in a bitplane encoding is to code the map of the significant coefficients. Zerotree coding is based on the assumption that if a coefficient in a given subband is insignificant, coefficients corresponding to the same location in higher-frequency subbands have a high probability to be also insignificant. All these coefficients are coded together with a single zerotree symbol (ZTR in the EZW denomination). After a coefficient has been declared as significant, the remaining bits will be output during the refinement pass (also called subordinate pass).

For the sake of clarity, see Algorithm 1. However all details can be found in the original paper [7].

In the 3D case, there are several possibilities to define the relationship between coefficients. For example, in [12], only the spatial link between pixels is used. However, it has been shown in [11] that the most efficient tree structure for EZW uses both spectral and spatial links. Finally, the tree structure illustrated on Figure 2 is used.

Table 1: Validation of our implementation of EZW used in this paper as a reference. Performance on Barbara image.

Rate (bpp)	Original EZW [7]		Reference used	
	MSE	PSNR	MSE	PSNR
1.0	19.92	35.14	18.52	35.45
0.5	57.57	30.53	55.63	30.68
0.25	136.8	26.77	138.21	26.73
0.125	257.1	24.03	246.10	24.22
0.0625	318.5	23.10	309.46	23.22
0.03125	416.2	21.94	382.68	22.30

Let us denote (i, j, k) the coordinate of one coefficient and (n_s, n_l, n_b), respectively, the number of samples (or columns), lines, and spectral bands of the image, these three numbers correspond to the size of the image for the three dimensions. Let $\mathcal{O}(i, j, k)$ the offspring of coefficient (i, j, k). We do not detail the case of the low frequency subband which is similar to the standard EZW. With the tree structure used here, we have

(i) if $i \geq n_s/2$ or $j \geq n_l/2$, $\mathcal{O}(i, j, k) = \emptyset$;
(ii) if $k \geq n_b/2$, $\mathcal{O}(i, j, k) = \{(2i, 2j, k), (2i + 1, 2j, k), (2i, 2j + 1, k), (2i + 1, 2j + 1, k)\}$;
(iii) else $\mathcal{O}(i, j, k) = \{(2i, 2j, k), (2i + 1, 2j, k), (2i, 2j + 1, k), (2i + 1, 2j + 1, k), (i, j, 2k), (i, j, 2k + 1)\}$.

It has to be noted that this structure leads to an overlapping tree structure. It has been found as being the most efficient in the case of EZW coding [11].

For the coefficients without descendant, there is no distinction to make between isolated zero (IZ) and zerotrees (ZTR), therefore the symbol Z is used. One modification is done for the high spatial frequency subband ($i \geq n_s/2$ or $j \geq n_l/2$) to make full use of the Z symbol when a coefficient has no descendant. With this modification, more coefficients are in this situation and the algorithm performs slightly better.

2.2. Validation of the reference implementation

The EZW algorithm used in this paper for reference is close to the original EZW in [7]. The wavelet transform and the arithmetic coder are performed using the latest version of the QccPack library [13]. The rest is programmed using ANSI C. Coding of the coefficients follows the details given in the original paper: initialization of the arithmetic model at the beginning of each new dominant and subordinate pass. It has to be noted that the tree structure chosen above is exactly the same as Shapiro's for 2D images. The only possible difference between the algorithms is the use of different symbol statistics for the arithmetic coder between the highest frequency subbands (where the three symbols POS, NEG, and Z are sufficient) and the lower frequency subbands (where four symbols are necessary: POS, NEG, IZ, and ZTR). This fact is not explicitly mentioned in Shapiro's paper. The performances of our reference are slightly better than the original EZW. PSNR and MSE values for Barbara image are given in Table 1.

Table 2: Effect on removing the subordinate pass. Results are for Moffett AVIRIS image (Figure 4(a)).

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

Lossless performance is also confirmed to be slightly better than the CB-EZW defined in [14]. In this latest paper, the lossless rate obtained for the $512 \times 512 \times 224$ scene 3 of Moffett Field from AVIRIS is 5.2605 bpppb (bit per pixel per band). With our reference, the rate is 5.1429 bpppb.

2.3. One drawback

One drawback of EZW is the memory required to store the coefficients already noticed 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 only for that purpose. For a $256 \times 256 \times 224$ hyperspectral image, counting 1 bit of memory to flag the position of significant coefficients, we need to keep an additional 14.7 Mbits in memory during compression. As a result, if the image is processed bitplane by bitplane (keeping only the current bitplane in memory), keeping this significance map in memory doubles the required amount of memory. One solution to remove the need for this memory is to remove the subordinate pass. In this situation only the significant pass is processed for each bitplane. Coefficients are considered as insignificant if the bit in the bitplane is 0 and significant otherwise. However, this simple change causes a loss in performances of more than 2 dB PSNR (see Table 2).

It is the purpose of this paper to propose an algorithm which does not require this additional memory (thus saving 14.7 Mbits in the previous example) without any significant loss in performance. This requires to increase the efficiency of the dominant pass for every bitplane, which is addressed in the next section.

3. IMPROVEMENT

3.1. Increasing the number of zeros

As we have seen, zerotrees high performance is to be credited mostly to their ability to code a great 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. One solution is to use a 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 $n = \sum_{i=0}^{\infty} a_i 2^i$.

$t = 0$
$a = (\dots, a_2, a_1, a_0)$, the standard binary notation for the number to convert
while $(\dots, a_{t+2}, a_{t+1}, a_t) \neq (\dots, 0, 0, 0)$
• if $a_t \neq 0$
– $b = (\dots, 0, \text{sgn}(a_t), -2^* \text{sgn}(a_t), 0, \dots, 0)$
(nonzeros at t, $t+1$)
– $c = a + b$
– if $c_{t+1} = 0$
$*a = c$
– endif
• endif
• $t = t + 1$
endwhile
return a

Algorithm 2: Signed binary digit representation.

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 -1 is used also instead of only 1 and 0, the number 119 can be noted $(1, 0, 0, 0, -1, 0, 0, -1)$ as it is equal to $1 * 2^7 - 1 * 2^3 - 1 * 2^0$.

The signed binary digit representation for a given number is not unique. Generally, the interest is in representations which have a maximum of 0s. This could be achieved considering the Hamming weight of a binary representation of a number. The Hamming weight of a number representation is equal to the number of nonzero elements in the representation a. In [15], an algorithm is given to find a signed binary digit representation of minimal Hamming weight (see Algorithm 2).

This algorithm is simple but not the most efficient in terms of complexity. Further research has been made on efficient algorithms to reach signed binary digit representation of minimal Hamming weight, we can cite [16–18] for example.

However, the signed binary digit representation of minimal Hamming weight is not unique. In general, the use of a signed binary digit representation is in fast exponentiation. The nonadjacent form (NAF), where nonzero digits are separated by at least one zero, is unique and provides the required properties for fast exponentiation. Hence, most of algorithms lead to the NAF form.

In our case, while the minimum Hamming weight is required, we cannot be sure that the NAF provides any advantage. Two different forms are compared in this paper using the transformation $(\dots, 1, 0, -1, \dots) \rightarrow (\dots, 0, 1, 1, \dots)$ and similarly $(\dots, -1, 0, 1, \dots) \rightarrow (\dots, 0, -1, -1, \dots)$. We denote this latest representation as AF (adjacent form). The two forms provide the same number of zeros. However, due to the different position of the first significant bit, the proportion of zeros can be slightly different. Examples of signed binary digit representation for number 349 are given in Table 3.

Table 3: Example of representation for number 349.

t	9	8	7	6	5	4	3	2	1	0
2^t	512	256	128	64	32	16	8	4	2	1
Binary	0	1	0	1	0	1	1	1	0	1
NAF	1	0	-1	0	-1	0	0	-1	0	1
AF	0	1	0	1	1	0	0	0	-1	-1

Table 4: Zero bits proportion after the first significant bit. Results are for Moffett AVIRIS image (Figure 4(a)).

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%
AF	2.85	25 507 573	61.03%

Table 5: EZW with independent processing of each bitplane (no subordinate pass).

Rate (bpppb)	Binary		NAF		AF	
	MSE	PSNR	MSE	PSNR	MSE	PSNR
1.0	193.73	73.46	149.07	74.60	151.76	74.52
0.5	685.49	67.97	549.56	68.93	553.10	68.90

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 extract of scene 3 of the Moffett Field data of $256 \times 256 \times 224$ coefficients (Figure 4(a)), 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 nonzero are detailed in Table 4. As shown in this table, the two signed binary digit representations managed to increase significantly the proportion of zeros for lower bitplanes: more than 60% of 0 s against 50% before. These results correspond to the value expected from the properties of signed binary digit representation.

EZW is implemented using signed binary digit representations (NAF and AF) 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 5), this improvement is not sufficient to recover from the loss due to the removal of the subordinate pass. We do not reach the original performance from Table 2. In this case, no major difference is noted between the NAF and the AF forms.

3.2. Using the spatial dependencies

This latest version of the EZW coder does not take into account the values of the neighboring coefficient in the same bitplane. A simple way to consider the neighboring coefficients is to use contextual arithmetic coding. Only three coefficients on the same bit plane are considered. These

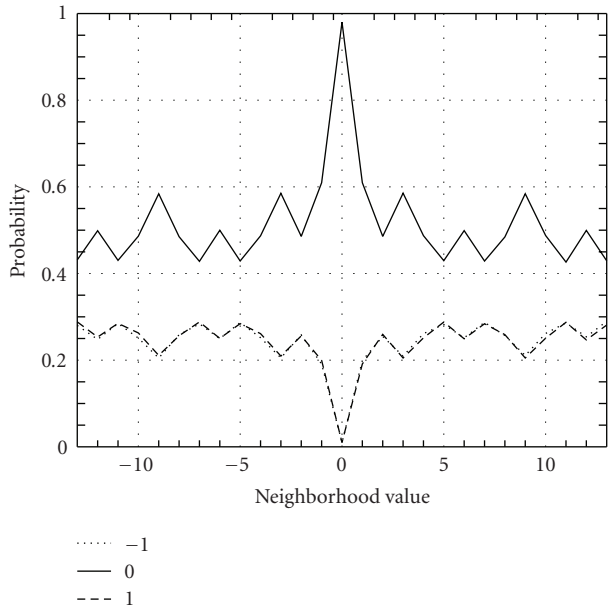


Figure 3: Probability of having value −1, 0, or 1 for the current coefficient according to the neighborhood value with the NAF form. The 27 possible neighborhoods are presented on abscissa according to the value of η.

coefficients are those preceding the current pixel in the three directions of the hyperspectral wavelet cube.

The coefficient at the same location on the previous bitplane is also considered. 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. In the case of the AF form, no such rule exists. We would have to double the number of contexts for the cases where the coefficient on the previous bitplane is 0 or is 1 or −1. Thus, the NAF form which leads to more simple contexts is chosen.

Let denote η_s, η_l, and η_b the preceding coefficients on the three directions. Thus, we have

(i) $\eta_s(i, j, k) = (i - 1, j, k)$,
(ii) $\eta_l(i, j, k) = (i, j - 1, k)$,
(iii) $\eta_b(i, j, k) = (i, j, k - 1)$.

As 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 neighborhoods and the integers between −13 and 13.

We can plot the probability of having the values −1, 0, or 1 according to the neighborhood values. The probability curves are presented on Figure 3. These probabilities are computed for the 256 × 256 × 224 Moffett image on all bitplanes for the NAF notation. Thus, several millions of data are taken into account. From these curves, we can see that one neighborhood clearly differs in terms of probability compared to the others, when $\eta = 0$, that is, $\eta_s = \eta_l = \eta_b = 0$.

Table 6: EZW with independent processing of each bitplane NAF with and without contextual coding.

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

With this neighborhood, the probability to have a 0 for the current coefficient is very high.

The context for the arithmetic coder will be separated in two cases: $\eta = 0$ and $\eta \neq 0$.

It can also be noted that in the case of the NAF, a nonzero value for a coefficient at a certain bitplane will be followed by one 0 at the next bitplane (hence the reason for the denomination *nonadjacent form*). In this case, it is not necessary to give any output for this 0. This advantage does not appear with the AF notation, and thus AF does not perform as well as NAF. Performance using the arithmetic coder with NAF are presented in Table 6.

This latest version of EZW without subordinate pass using NAF and contextual arithmetic coding is referred to as 3D-EZW-NAF.

4. RESULTS

3D-EZW-NAF is applied to a 256 × 256 × 224 extract of the scene 3 of f970620t01p02_r03 run from AVIRIS sensor on Moffett Field site. This part is shown in Figure 4(a) and is the most difficult part of the image to compress (urban area). Another image is a 256 × 256 × 224 extract of the scene 1 of f970403t01p02_r03 AVIRIS run over Jasper site. This part is shown in Figure 4(b). 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 compression.

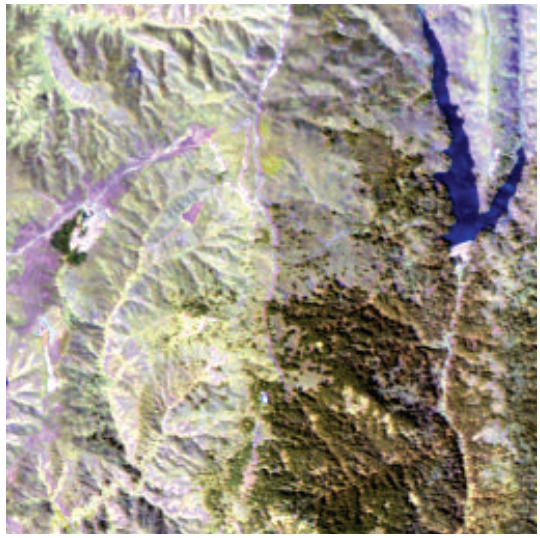
Mean-square error (MSE) and peak signal-to-noise ratio (PSNR) for different rates are given in Table 7 for Moffett Field image and in Table 8 for Jasper image. The rate is given in bit per pixel per band (bpppb), the PSNR in dB is calculated as

$$\mathrm{PSNR} = 10 \log_{10} \frac{(2^{16} - 1)^2}{\mathrm{MSE}}. \quad (1)$$

The use of 2^{16} as peak signal (signal dynamic) explains the unexpectedly high PSNR values, however to keep the scientific value of the data, the PSNR value has to be kept above 65 dB.

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, thus making the hardware implementation easier. The full rate-distortion curve is presented on Figure 5 for the Moffett image.

(a)



(b)

Figure 4: Hyperspectral images used during the experiment: (a) from Moffett Field site and (b) from Jasper site (b). Images are all in radiance.

Table 7: Comparison between 3D-EZW and the simplified version using signed binary digit representation (NAF) on AVIRIS image Moffet.

Rate (bpppb)	3D-EZW		3D-EZW-NAF	
	MSE	PSNR	MSE	PSNR
1.0	106.15	76.07	121.38	75.49
0.5	445.22	69.84	457.77	69.72
0.25	1407.34	64.85	1514.81	64.53
0.125	3933.86	60.38	4402.34	59.89

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 9). 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).

Table 8: Comparison between 3D-EZW and the simplified version using signed binary digit representation (NAF) on AVIRIS image Jasper.

Rate (bpppb)	3D-EZW		3D-EZW-NAF	
	MSE	PSNR	MSE	PSNR
1.0	40.56	80.25	43.49	79.95
0.5	139.31	74.89	140.76	74.84
0.25	391.31	70.40	411.75	70.18
0.125	981.79	66.41	1080.47	65.99

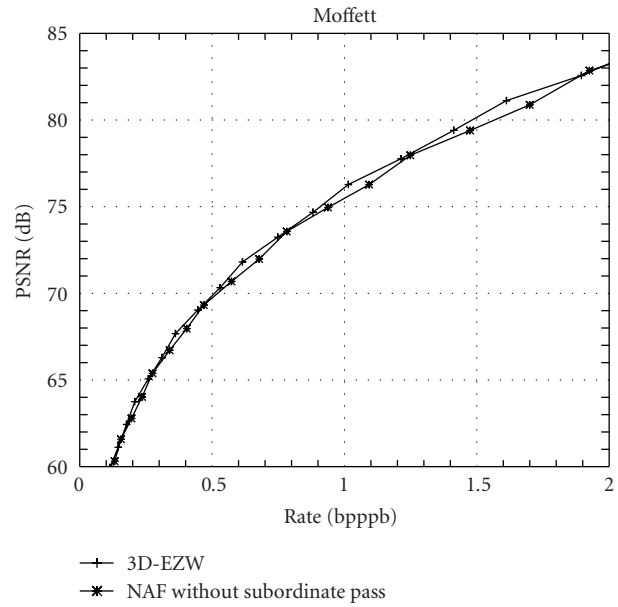


Figure 5: Comparison of compression performance between 3D-EZW and the NAF without subordinate pass.

In terms of complexity, a precise estimation would be required before hardware implementation. However a rough and simple way to measure the complexity is computation time. The coding time is similar between 3D-EZW and 3D-EZW-NAF: about 100 s for both versions. The conversion to signed binary digit representation is not optimized in our case (adding about 24 s) and could be greatly reduced with one of the smarter algorithms available in the literature. As one of the main applications of signed binary digit representations is on speeding exponentiation operation, fast convertion should not be a problem.

Note, however, that the main interest of the proposed solution is that it provides an algorithm which can be very easily parallelized, at almost no cost in terms of speed/complexity tradeoff. We can imagine for example to use separated coding units to encode each bitplane. Each of these coding units would be fed with one bitplane and would

Table 9: Comparison between 3D-EZW and 3D-EZW-NAF with subordinate pass.

Rate (bpppb)	3D-EZW		3D-EZW-NAF with subordinate pass	
	MSE	PSNR	MSE	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
0.125	3933.86	60.38	4001.42	60.30

output the portion of the bitstream corresponding to this bitplane.

The nature of the error caused by 3D-EZW-NAF is similar to the error caused by 3D-EZW and other wavelet-based compression algorithms. The compression introduced a quantization of the wavelet coefficients. For the given rates the degradation remains small and is similar to white noise.

5. CONCLUSION

Signed binary digit representations, particularly the NAF, have shown a good ability to compensate for the removal of the subordinate pass. However, this compensation is not as significant as expected but it enables a simplified algorithm to perform almost as well as the original one.

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.

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. The authors wish to thank NASA/JPL for providing the hyperspectral images used during the experiments.

REFERENCES

[1] A. Grossmann and J. Morlet, "Decomposition of hardy functions into square integrable wavelets of constant shape," *SIAM Journal of Mathematical Analysis*, vol. 15, no. 4, pp. 723–736, 1984.

[2] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[3] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961–1005, 1990.

[4] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, 1992.

[5] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 160–175, 1993.

[6] *Information technology—JPEG 2000 image coding system: core coding system*, ISO/IEC 15 444-1, 2002.

[7] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

[8] A. Said and W. 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*, vol. 6, no. 3, pp. 243–250, 1996.

[9] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, 2000.

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

[11] E. Christophe, C. Mailhes, and P. Duhamel, "Hyperspectral image compression: adapting SPIHT and EZW to anisotropic 3-D wavelet coding," submitted to *IEEE Transactions on Image Processing*.

[12] 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.

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

[14] A. Bilgin, G. Zweig, and M. W. Marcellin, "Three-dimensional image compression with integer wavelet transforms," *Applied Optics*, vol. 39, no. 11, pp. 1799–1814, 2000.

[15] S. Arno and F. S. Wheeler, "Signed digit representations of minimal Hamming weight," *IEEE Transactions on Computers*, vol. 42, no. 8, pp. 1007–1010, 1993.

[16] H. Prodinger, "On binary representations of integers with digit -1, 0, 1," *Integers Electronic Journal of Combinatorial Number Theory*, vol. 0, 2000.

[17] M. Joye and S.-M. Yen, "Optimal left-to-right binary signed-digit recoding," *IEEE Transactions on Computers*, vol. 49, no. 7, pp. 740–748, 2000.

[18] K. Okeya, K. Schmidt-Samoa, C. Spahn, and T. Takagi, "Signed Binary representations revisited," in *Advances in Cryptology—CRYPTO 2004*, vol. 3152 of *Lecture Notes in Computer Science*, pp. 123–139, Springer, New York, NY, USA, 2004.