Hyperspectral Image Compression: Adapting SPIHT and EZW to Anisotropic 3-D Wavelet Coding

Emmanuel Christophe, *Member, IEEE*, Corinne Mailhes, *Member, IEEE*, and Pierre Duhamel, *Fellow, IEEE*

***Abstract*—Hyperspectral images present some specific characteristics that should be used by an efficient compression system. In compression, wavelets have shown a good adaptability to a wide range of data, while being of reasonable complexity. Some wavelet-based compression algorithms have been successfully used for some hyperspectral space missions. This paper focuses on the optimization of a full wavelet compression system for hyperspectral images. Each step of the compression algorithm is studied and optimized. First, an algorithm to find the optimal 3-D wavelet decomposition in a rate-distortion sense is defined. Then, it is shown that a specific fixed decomposition has almost the same performance, while being more useful in terms of complexity issues. It is shown that this decomposition significantly improves the classical isotropic decomposition. One of the most useful properties of this fixed decomposition is that it allows the use of zero tree algorithms. Various tree structures, creating a relationship between coefficients, are compared. Two efficient compression methods based on zerotree coding (EZW and SPIHT) are adapted on this near-optimal decomposition with the best tree structure found. Performances are compared with the adaptation of JPEG 2000 for hyperspectral images on six different areas presenting different statistical properties.**

***Index Terms*—Compression, EZW, hyperspectral, JPEG 2000, SPIHT, zerotrees.**

I. Introduction

IMAGE sensors, whether used to observe the Earth from space or to explore deep space and distant bodies, always seek better data quality to improve the scientific or the strategic value of the information provided. Improving the performance of such sensors often requires an increase in the spatial resolution, the radiometric precision and possibly the number of spectral bands. High-spectral resolution instruments, fall within this global evolution. Such sensors, named either imaging spectrometers or hyperspectral sensors, are becoming increasingly common nowadays.

Manuscript received May 13, 2007; revised August 05, 2008. Current version published November 12, 2008. This work was supported in part by the Centre National d'Études Spatiales (CNES), TeSA, the Office National d'Études et de Recherches Aérospatiales (ONERA), and by Alcatel Alenia Space. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Eli Saber.

E. Christophe is with the CNES (French Space Agency), Toulouse, France (e-mail: e.christophe@ieee.org; emmanuel.christophe@gmail.com).

C. Mailhes is with the University of Toulouse (ENSEEIHT/IRIT) and TESA, Toulouse, France (e-mail: corinne.mailhes@enseeiht.fr).

P. Duhamel is with the CNRS—LSS, Gif-sur-Yvette, France (e-mail: pierre.duhamel@lss.supelec.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2008.2005824

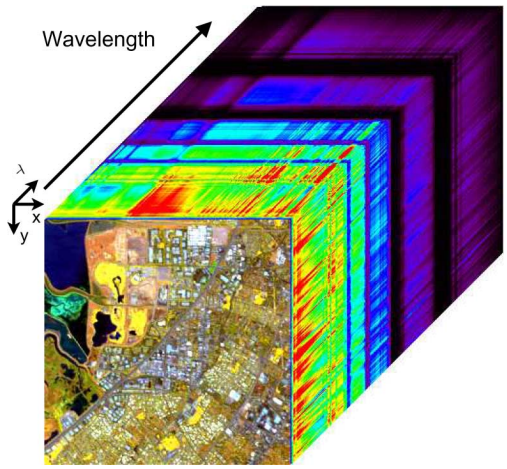


Fig. 1. Example of a hyperspectral data cube (Moffett Field): The front of the cube is a color composite of three spectral bands while the other sides display the spectra of the side pixels.

Significant constraints limiting the performance of a new instrument are the available transmission bandwidth and the on-board storage capacity. The compression step, therefore, becomes a crucial part of the acquisition system as it enhances the ability to store, access and transmit information. Ideally, the compression should be lossless to ensure preservation of the scientific value of data. However, lossless compression techniques provide compression ratios of about two or three, a limitation which is enforced in the hyperspectral case due to the noise inherently present is such high-resolution sensors [1]. Near lossless compression becomes an increasingly acceptable choice during the sensor definition.

Hyperspectral imagery, or spectral imagery, involves observing the same scene at different wavelengths (Fig. 1). Typically, each image pixel is represented by hundreds of values, corresponding to various wavelengths. These values correspond to a sampling of the continuous spectrum emitted by the pixel. This sampling of the spectrum at very high resolution allows pixel identification (materials, minerals, gases, etc.). The availability of the spectral information for each pixel leads to new applications in all fields that use remote sensing data (agriculture, environment, or military), and can help to improve the understanding of the solar system (mineral or gas identification). Hyperspectral data are in a way similar to video data, where wavelength corresponds to time, but their statistical properties are different: there is no motion between hyperspectral spectral planes but changes in *color*, as illustrated in Fig. 2.

(a) (b) (c) (d)

Fig. 2. Example of hyperspectral data (Moffett Field): the same scene at four different wavelengths (taken from the 224 bands of the original image) at 458 nm (a), at 664 nm (b), at 712 nm (c), and at 1211 nm (d). Strong similarities exist between the images, the relevant information is present in the details.

Due to the huge amount of information and specific properties involved, compressing hyperspectral images is a challenge [2]. A suitable and adapted compression system is still being awaited. Many actual hyperspectral instruments do not make use of compression, thus limiting the amount of data that can be stored and transmitted.

Existing work focuses mainly on two different techniques, namely vector quantization and wavelets. This paper concentrates on the wavelet approach and more precisely on zerotree based methods. Zerotree have demonstrated a good ability to compress hyperspectral data [3] and are successfully used in several space missions. Section II presents the JPEG 2000 compression standard and proposes various adaptations for hyperspectral images corresponding to several trade-offs between performance and complexity. Section III defines an algorithm to find the best 3-D wavelet decomposition for hyperspectral images in a rate-distortion sense. Section IV proposes two adaptations of the zerotree structure and bit plane coding based on the proposed anisotropic near-optimal wavelet decomposition. The first adaptation closely follows EZW and the second one is closer to SPIHT. The results obtained from various hyperspectral images are compared to the performance of JPEG 2000 algorithms in term of PSNR in Section V. We conclude and present perspectives in Section VI.

II. JPEG 2000 for Hyperspectral Data

JPEG 2000 is the latest international standard for still image compression, on which exhaustive bibliography can be found [4], [5].

Parts 1 and 2 of the JPEG 2000 standard [6], [7] are targeted at still images in grey level or with three color bands, and possibly, a fourth alpha band. In these parts, no interband transform is defined apart from color transforms. Part 2, however, makes provisions for arbitrary spectral decorrelation, including wavelet transform. Part 10, also known as JP3D, is targeted for 3-D images which are, however, *as isotropic as possible* [8]. This requirement does not suit hyperspectral images, for which the spectral dimension involves a much higher correlation than the spatial dimensions [9]. As a consequence, the JP3D part of the standard is not ideally suited to hyperspectral image compression. Instead, we propose to use the extensions of the JPEG 2000 standard by introducing transform on the spectral components before applying the JPEG 2000 encoder.

The reference implementation of the JPEG 2000 compression system is the *Verification Model* (VM). The VM is used by the JPEG 2000 committees as a vehicle for core experiments and will ultimately evolve into an implementation of the final JPEG 2000 standard [10]. During this study, VM 9.1, the latest version to date, is used to evaluate the performance of JPEG 2000 compression standard on hyperspectral images. With the options used here, performance is very similar to the one obtained with another well-known implementation of JPEG 2000: version 5.0 of Kakadu.

As already stated, Parts 1 and 2 do not include specifications for the case where the image has more than three bands (color transform). However, the VM let the user specify a component transform matrix, allowing Discrete Cosine Transform (DCT) or Karhunen–Loeve Transform (KLT) (*-Mlin* option of the VM). An option which applies a wavelet transform (DWT) through the components is available as well (*-Mtdt* option). In these cases, the 1-D transform is applied in the spectral dimension before JPEG 2000 encoding of each resulting image. The Lagrangian rate-distortion optimization is also a very useful ingredient, since the various components have very different statistical properties, due to the energy compaction property of the transform (*-Flra* option).

Fig. 3 compares the impact of various transforms, namely DCT, KLT interband decorrelations or wavelet transform in terms of Peak Signal to Noise Ratio (PSNR) (1) as a function of bit per pixel per band (bpppb)

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

MSE being the Mean Square Error. The high dynamics of the image (16 bits) explains the high values reached by the PSNR.

As expected, introducing a decorrelation prior to the JPEG 2000 encoder greatly improves the results. The relative position of the KLT and DWT are similar to the results obtained in [11]. However, the DCT provides smaller PSNR's in our study.

The KLT depends on the image and requires more complex operations: computation of the covariance matrix and eigenvectors. This high complexity is still a deterrent for on-board implementation, especially when the number of components increases even if some work is done to reduce its complexity [12], [13]. The wavelet transform approach seems more suitable, especially in the hyperspectral case where the number of components is important. The wavelet transform also provides higher performance than DCT.

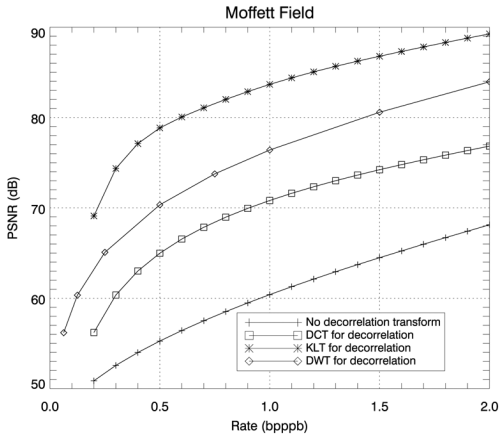


Fig. 3. Performances of the JPEG 2000 compression using either no interband decorrelation, DCT, KLT, or DWT decorrelation on the Moffett Field data set of Fig. 1.

Generally, the JPEG 2000 algorithm is computationally demanding, especially in the context of on-board spacecraft, where the implementation constraints are stiff. To the authors knowledge, there is only one hardware implementation of JPEG 2000 designed for space constraints [14]. In his paper, Van Buren admits that his solution performs significantly lower than Kakadu in the lossless case due to the absence of rate-distortion optimization. The Consultative Committee for Space Data Systems (CCSDS), a working group gathering the main space agencies (NASA, JAXA, ESA, CNES, CSA), issued some recommendations for onboard image compression systems in [15]. The adopted recommendation groups the wavelet coefficients in a structure similar to zerotrees instead of using the JPEG 2000 standard. The intent of this recommendation is to alleviate onboard implementation complexity. One should view the JPEG 2000 performance in the present article as an upper-bound to reach: our goal is to obtain performance close to this improved version of JPEG 2000 (in terms of PSNR versus bitrate tradeoff), while greatly reducing the algorithm complexity.

The complexity reduction is not obtained directly through the use of a wavelet transform in all dimensions, but it turns out that the use of wavelets enables the use of efficient, simple coding tools. Therefore, our study first involves the search for an optimal wavelet decomposition in a rate-distortion sense. This is the purpose of Section III. Then an efficient coding of the wavelet coefficients with zerotree methods is proposed in Section IV.

III. Optimal Decomposition

Before adapting the zerotrees to hyperspectral images, it is necessary to define the extention of the wavelet transform to hyperspectral images. Most current extensions are based on isotropic 3-D decomposition [16]–[18], however, as mentioned before, hyperspectral data are clearly not isotropic. In the domain of video processing some anisotropic structure has also been successfully used [19], [20]. However, no justification has been given concerning the particular choice of this structure for hyperspectral data, and more efficient decompositions could be available. The best choice for video is anyway not obviously the best choice for hyperspectral images, due to the differences in their statistical behavior and in requirements for data quality.

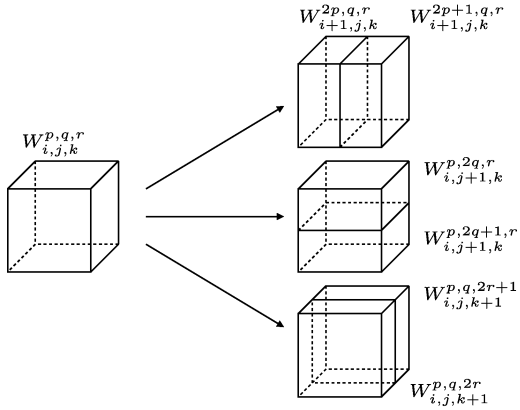


Fig. 4. Anisotropic decomposition and notations.

The problem of finding the optimal wavelet decomposition for 1-D signal has been explored in several publications [21], [22]. For natural 2-D images, decomposition possibilities were long restricted to quadratic transforms (leading to square subbands) but have since evolved to a more general framework with anisotropic decompositions [23]. Several criteria were used to characterize the optimal decomposition: entropy-based algorithms [24] or rate-distortion compromise [25] for example. The main advantage of the latest is that it provides simultaneously the bit allocation between the different subbands [26].

The 3-D anisotropic wavelet decomposition is detailed in Section III-A. Section III-A (Section III-B) recalls the principles of rate-distortion optimization looking for the optimal decomposition. The idea of the algorithm is illustrated on a well-known 2-D image in Section III-C. Section III-D details the algorithm designed to find the optimal wavelet decomposition for hyperspectral images and presents the results. The last subsection (Section III-E) finds a tradeoff between the execution time and performance. All results presented in this section are within the context of rate distortion theory and do not take into account headers necessary to decode the final bitstream. These results should not be compared directly with those from Sections II and V.

A. Three-Dimensional Anisotropic Wavelet Decomposition

Traditionally, on 2-D images, the wavelet decomposition is isotropic, i.e., for one given subband, the level of decomposition in the horizontal direction is the same as the level of decomposition in the vertical direction. This alternation between horizontal and vertical decompositions leads to square subbands (cubes, in the case of 3-D data). This is the case of the multiresolution decomposition of Mallat [27] or the wavelet packets decomposition. The term anisotropic is more general than the usual use of *wavelet packet*. In most case for image processing, wavelet packets correspond to a decomposition in a quadtree structure, leading to square subbands. This process is justified by the properties of traditional images: their statistical properties are quite similar in all directions.

Let us denote $W^{p,q,r}_{i,j,k}$ the wavelet subband space in the 3-D case (Fig. 4):

- i, j, k corresponding to the row, column and spectral levels, respectively (implying the size of the considered subband);
- p, q, r being the row, column and spectral indexes, respectively.

A relation can be defined between subbands. For a row decomposition, the anisotropic wavelet space satisfies

$$W_{i,j,k}^{p,q,r} = W_{i+1,j,k}^{2p,q,r} \oplus W_{i+1,j,k}^{2p+1,q,r}. \tag{2}$$

where $\oplus$ denotes the direct sum of vector spaces. Note the factor 2 in the indexes, since decomposition level $i+1$ has twice as much subbands than level i.

For a column decomposition

$$W_{i,j,k}^{p,q,r} = W_{i,j+1,k}^{p,2q,r} \oplus W_{i,j+1,k}^{p,2q+1,r} \tag{3}$$

and for a spectral decomposition

$$W_{i,j,k}^{p,q,r} = W_{i,j,k+1}^{p,q,2r} \oplus W_{i,j,k+1}^{p,q,2r+1}. \tag{4}$$

For any step of the decomposition, for all subbands, we are able to choose the direction of the next decomposition, thus increasing the flexibility of the space decomposition. Both multiresolution decomposition and wavelet packet decomposition are special cases of this representation.

B. Rate-Distortion Optimization

1) Allocation Problem: The problem of bit allocation, i.e., distributing optimally the given bit budget between the subbands, is a classical problem in data compression. Shoham and Gersho [26] address this problem within the framework of rate-distortion theory. Their solution consists in minimizing the distortion under the constraint of the available bit budget.

Within the context of wavelet decomposition, different quantizers can be used for different subbands. Let S be the finite set of the quantizer combination for the subbands, let B be one element of S. The problem is to minimize the total distortion $D(B)$ for the given combination of quantizers, B, with the total rate $R(B)$ within the bit budget R_c

$$\min_{B \in S}\{D(B)\} \quad \text{under} \quad R(B) \leq R_c. \tag{5}$$

Using the Lagrangian method, this constrained minimization becomes the minimization of the Lagrangian cost function J without constraint

$$J(\lambda_J) = D + \lambda_J R. \tag{6}$$

In the context of independent coded subbands, using additive measures for rate and distortion, it can be shown that R-D optimality is attained when *all subbands* operate at a constant slope point λ_J on their R-D curve. Thus, the problem becomes

$$\min\{D_k + \lambda_J R_k\} \text{ for each subband } k. \tag{7}$$

The proof of the equivalence between the constrained and the unconstrained problem is actually quite simple and can be found in [26].

Algorithm: The algorithm is defined to search for the best decomposition simultaneously with the best operating point. For a given subband, the algorithm computes the R-D points for different quantizers, thus leading to the R-D curve for the current subband. For a given quantizer q, the distortion D_k^q is computed using the *square error* as it has to be additive between subbands to comply with (7). Note that, strictly speaking, the additive property holds only for orthogonal wavelets, but that the ones that are used in the JPEG 2000 standard are very close to be orthogonal.

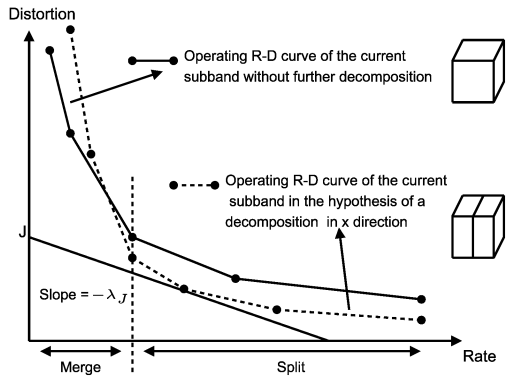


Fig. 5. Illustration of the split-merge decision during the algorithm. For sake of clarity, R-D curves in the hypothesis of decomposition in the two other directions (y or spectral) are not represented and are considered to be above the others. With the considered λ_J, the decision of splitting the current subband is taken.

The rate R_k^q of each subband is evaluated using the arithmetic coder from [28]. The choice of the coder is not critical here. What is important is the relative position between the different subbands, not the absolute performances. Simulations performed with other rate estimation as the entropy of the subband coefficient or using a combination of run length coding and Rice coder lead to similar results. The independent coding of the wavelet subbands is an implicit assumption here.

The R-D curve is also computed for the three possible further decompositions (corresponding to the three directions). A representation similar to Fig. 5 is obtained. For each value of λ_J, the cost function J is computed for each admissible R-D point. The decision of splitting or not the given subband is taken according to the minimum cost.

As an example, Fig. 5 shows a case for which the λ_J slope leads to take the decision of splitting the given subband in x direction.

Denote J_0 as the Lagrangian cost of coding the current subband without further decomposition. The Lagrangian costs J_1, J_2 and J_3 evaluate the costs corresponding to a decomposition of the current subband, respectively, on x, y or λ direction. The search for the best basis is done as follows:

Algorithm 1: Best-basis search

• Recursive function: $\text{cost}\left(W_{i,j,k}^{p,q,r}, \lambda_J\right)$

— compute the cost $J_0 = J(\lambda_J)$ using Shoham and Gersho algorithm for the current subband

— compute the cost J_1

• if the minimum size is not reach for the rows:
$J_1 = \text{cost}\left(W_{i+1,j,k}^{2p,q,r}, \lambda_J\right) + \text{cost}\left(W_{i+1,j,k}^{2p+1,q,r}, \lambda_J\right)$ by recursive calls

• otherwise $J_1 = \infty$

— compute the cost J_2: similar to J_1 for the columns

Fig. 6. Barbara image and the best anisotropic wavelet decomposition obtained for a bitrate of 0.9 bit per pixel (bpp).

— compute the cost J_3: similar to J_1 for the spectral dimension

— return the value $\min\{J_0, J_1, J_2, J_3\}$

• Global Function

— For each λ_J: call cost $\left(W_{0,0,0}^{0,0,0}, \lambda_J\right)$

— Full rate-distortion curve for the given image

■

This algorithm leads to a different decomposition for each image and each target bitrate. It has to be emphasized that due to the recursivity of the algorithm, the optimal R-D curve is first calculated for the smallest subbands then the algorithm gathers these values to take the splitting decision and finally to find the optimal decomposition for each subband.

This search for the optimal basis is similar in spirit to the one defined by Ramchandran in [25], with an extension to anisotropic decomposition in a 3-D space.

C. On 2-D Images

The search for the best wavelet decomposition has been first applied to natural 2-D images as an illustration. For some images (the well-known *Lena* for example), the best wavelet decomposition is not so far from the classical multiresolution decomposition. This result agrees with the choice of the multiresolution decomposition for 2-D images in standard like JPEG 2000. However, in the case of images containing strong frequency features, such as *Barbara* for example (Fig. 6), the decomposition manages to concentrate the energy in very few subbands (subband 1) and to group many coefficients within the same subband (subband 2). The gain can reach 1.5 dB compared to the classical multiresolution decomposition as shown in Fig. 7. However, in the general case, for 2-D images, the gain obtained with optimal decomposition is not sufficient to justify such an increase in the complexity.

D. Optimal Decomposition for Hyperspectral Images

The results presented on Fig. 10 are obtained on data from the AVIRIS hyperspectral sensor from JPL/NASA over the Moffett Field site in California (Fig. 8). Moreover, they were confirmed on different sites as well as on images acquired by the satellite sensor from NASA: Hyperion.

As can be seen from Fig. 10, the best basis decomposition brings a clear improvement, leading to an increase of the quality of 8 dB compared to the isotropic decomposition. If the limit

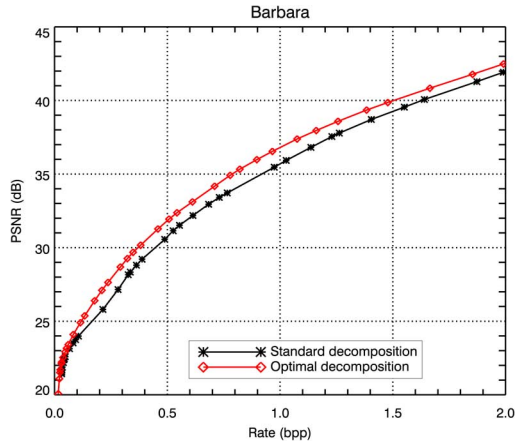


Fig. 7. Comparison between the classical decomposition and the best basis decomposition in the case of a 2-D image containing strong frequency features. For more natural images without strong frequency features, as Lena for example, the gain is negligible.

is fixed in term of quality, let say for example a PSNR greater than 70 dB, the necessary bit budget cuts down from 1 bit per pixel per band (bpppb) to 0.5 bpppb which doubles the compression rate. It is clearly seen that, due to the specific nature of hyperspectral images, the adaptation of the transform has a much larger impact.

E. Fixed Decomposition

There are two main drawbacks to this optimal search:

- the processing cost;
- the dependency from the image.

The processing cost is important, as an example, the processing of an hyperspectral cube of $256 \times 256 \times 224$ pixels with a minimum subband size of $8 \times 8 \times 7$ (5 decomposition levels) requires to process the full rate distortion curve for 250 047 subbands (arrangement of all possible subband sizes and positions in the 3 dimensions). The dependency from the image also poses a problem for the implementation of the transform. In general, data independent transforms are preferred.

The aim is to define a fixed transform close enough to the optimal one to give near-optimal results for a wide variety of images and bitrates. After observing the general structure of the optimal decomposition for different images at different bitrates, it appears that one regular decomposition is close to the optimal performance in many cases. This regular decomposition actually consists in decomposing first the spectra with a 1-D multiresolution wavelet and then using the standard multiresolution decomposition on the resulting components.

The resulting decomposition and the resulting coefficients values are illustrated in Fig. 9. This decomposition is compared with the standard isotropic decomposition. Coefficients in grey represent coefficients close to 0, white are positive and black are negative coefficients. As can be seen in Fig. 10, the chosen fixed transform is almost as efficient as the optimal one. The results are illustrated here on the Moffett data cube [Fig. 8(b)] but were confirmed on various hyperspectral images with very different characteristics (sea, forest, mineral, cities; see Fig. 8) and from different sensors (AVIRIS and Hyperion). The same experiment applied on the popular video sequence *tempete* and on the 3-D MR medical image *sag_head* confirm that this decomposition,

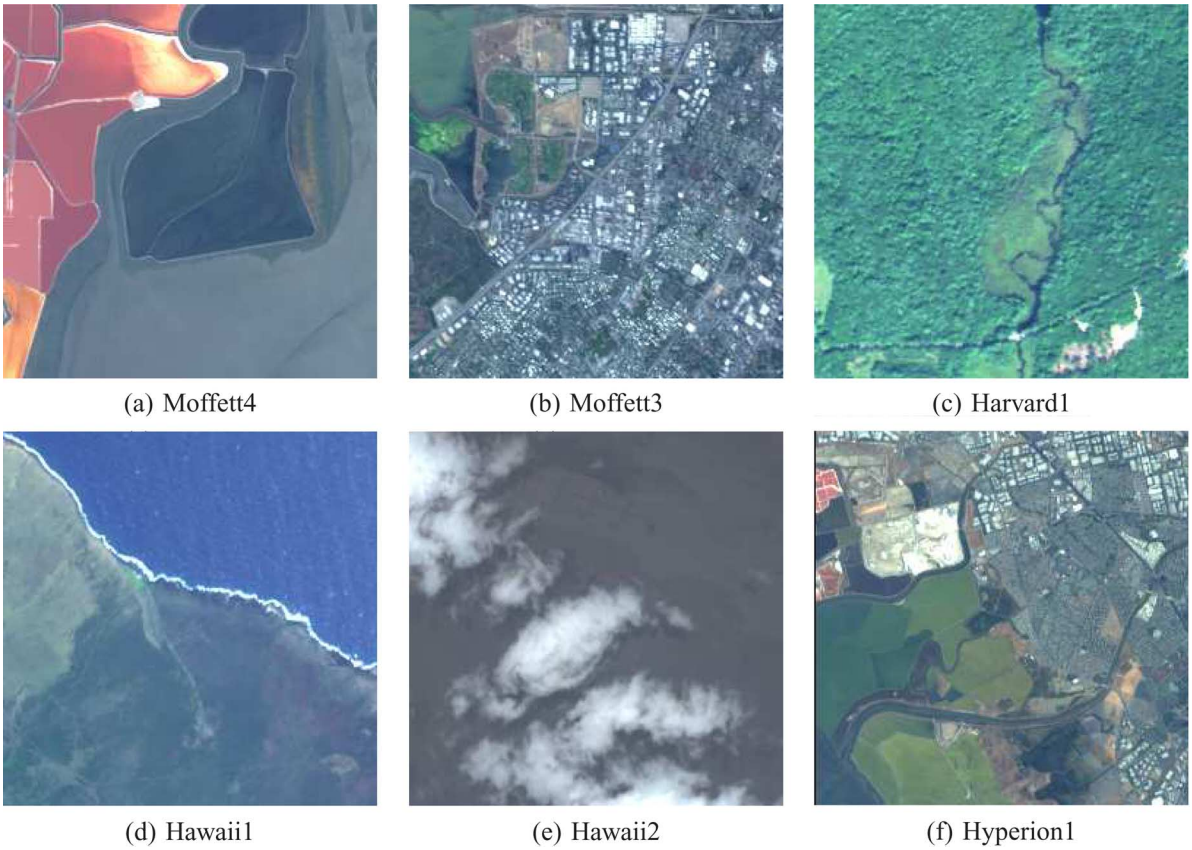


Fig. 8. Different hyperspectral images used during the experiments. (a), (b) Different parts from the f970620t01p02_r03 run from AVIRIS sensor on Moffett Field site. (a) Uniform spatial area with strong spectral features. (b) Mixed area with city (strong spatial frequency features). (c) From f010903t01p01_r03 AVIRIS run over Harvard Forest, it contains mostly vegetation. (d), (e) From f000414t01p03_r08 run over Hawaii. (d) Two areas (mineral and sea) delimited by a coast line. (e) Selected to illustrate a run containing clouds, thus a higher dynamic level and strong contrasts. (f) From Hyperion space sensor (EO1H0440342002212110PY) and cover an area similar to (a) and (b) but with a different sensor.

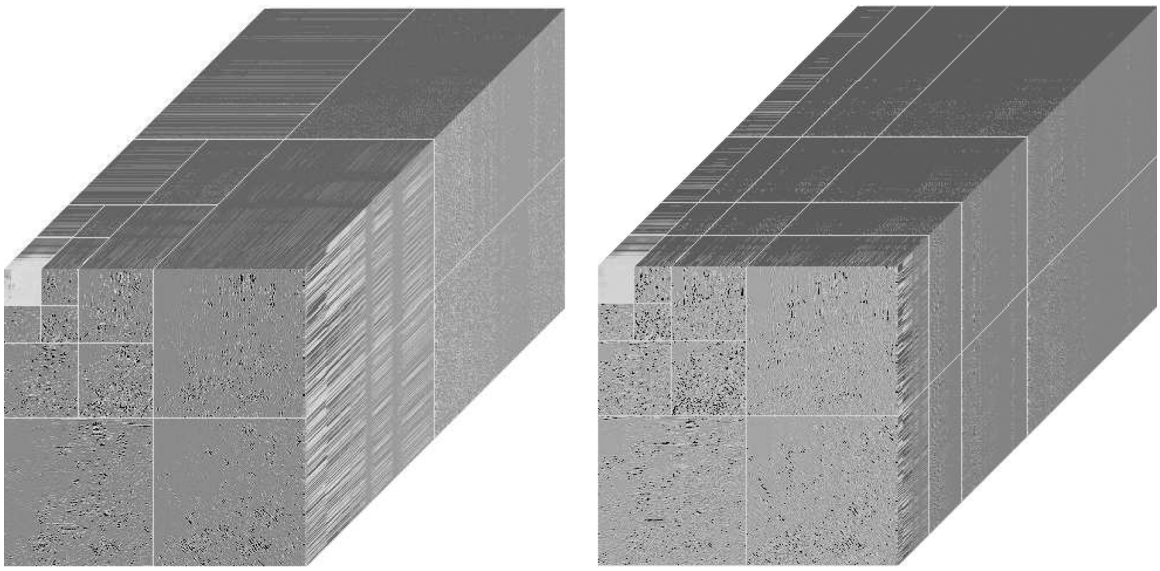


Fig. 9. Classical 3-D isotropic decomposition (a) and anisotropic decomposition (b) with three levels of decomposition (simulation are done with five levels of decomposition). For the isotropic decomposition, it can be seen easily that an important correlation remains between the coefficients in the lower spectral and spatial frequencies. The remaining correlation (e.g., lines of coefficients with similar values on the side of the cube) is lower for the anisotropic decomposition.

also used in [19] or [20], is near optimal in the context of video compression or medical image compression.

These results are also *a posteriori* justification for the transform used in [29] and [30] in the context of hyperspectral image

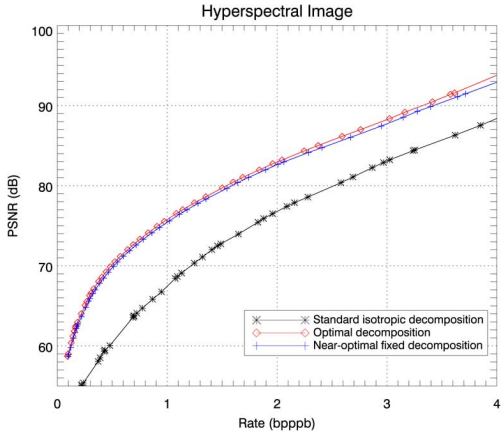


Fig. 10. Results on hyperspectral data, the anisotropic best-basis clearly improves the performance. The fixed anisotropic decomposition performs closely to the optimal decomposition.

compression. It should be noted also that this near optimal transform is actually similar to the one implicitly used by JPEG with the options described in Section II.

IV. Zerotrees

One possible weakness of JPEG 2000 is that it does not make use of the relation that exists between the locations of significant subband coefficients. According to Taubman and Marcellin [4], the benefits from choosing the truncation point compensates for the fact that the parent-child relations are not used. This may no longer be true for hyperspectral images as the correlation between subbands is unusually high.

Zerotrees of wavelet coefficients were developed to make use of the relationships between the locations of significant subband coefficients. After the wavelet transform, it can be observed that the location of the insignificant coefficients is similar within the different subbands, even if their value is decorrelated. The idea behind zerotrees of wavelet coefficients is that if a coefficient is insignificant in one subband, the coefficient in the same location and in a higher frequency subband is probably also insignificant. This idea has been successfully used by Shapiro in [31], developing the embedded zerotree coding of wavelet coefficients (EZW), as well as, few years later, by Said and Pearlman in their SPIHT algorithm [32].

The main benefits of zerotree coding are presented in Section IV-A and B, which also discusses their adaptation to the anisotropic decomposition, as defined above. The application of EZW algorithm on hyperspectral images is explained in Section IV-C. Another adaptation closer to SPIHT is developed in Section IV-D.

A. Main Ideas

At the time of its publication, embedded zerotree coding of wavelet coefficients (EZW) from Shapiro [31] produced state-of-the-art compression performance at a modest level of complexity. Later, zerotrees were generalized by Said and Pearlman in [32] and a set partitioning method was introduced to efficiently encode these zerotrees. This generalization is known as set partitioning in hierarchical trees (SPIHT).

TABLE I
Bitplane Coding for Three Different Example Coefficient Values: 7, 30, and 180. Coding Process From Top Down

Bitplane	T_k	7	30	180
7	128	0	0	1
6	64	0	0	0
5	32	0	0	1
4	16	0	1	1
3	8	0	1	0
2	4	1	1	1
1	2	1	1	0
0	1	1	0	0

These two algorithms share some properties which make them particularly attractive in the context of on-board hyperspectral image compression. Both of them produce an embedded bit-stream: every prefix of a bitstream produced by EZW (resp. SPIHT) is a valid EZW (resp. SPIHT) bitstream, leading to a decompressed image with a lower quality. Both algorithms manage to achieve this at a relatively modest level of complexity.

Low complexity is one of the main constraints for on-board implementation of compression algorithms since the processing power is limited. This is especially true in the case of spacecraft. Buffer constraints make the embedded property useful as they help to avoid overflow and to assure the best image quality given the available on-board memory and processing resources.

To ensure that the property of embedded bit stream is adhered to, the algorithm uses bit-plane encoding of coefficients. Let us define $c_{x,y,\lambda}$ as the wavelet coefficient at column x, line y and spectral plane λ. Also, define a sequence of thresholds, $T_{K-1},\ldots,T_0$, such as $T_k = T_{k+1}/2$. The initial threshold is chosen such that $|c_{x,y,\lambda}| < 2T_{K-1}$ for all wavelet coefficients. For convenience of binary representation, we also choose T_{K-1} as a power of 2. The coefficient $c_{x,y,\lambda}$ is considered significant in bitplane k if $|c_{x,y,\lambda}| \geq T_k$. Bit planes are encoded one after the other, reducing the distortion at each step.

An example of bitplane coding is presented in Table I for three coefficient values. The value 7 is considered as insignificant until bitplane 2, the value 30 is insignificant until bitplane 4 and the value 180 is significant from bitplane 7. After a coefficient is marked as significant, bits following the first significant bit have to be coded. Coefficient signs are coded separately.

B. Tree Structure

Given the near optimal decomposition found on the previous section, several tree structures can be defined. We can use the link between the spatial subbands, between the spectral subbands or use both. One key advantage of zerotree coder is to encode an important amount of zero coefficients using very few symbols. Thus, the best tree structure is the one which maximizes the length of the zerotrees, leaving only a small amount of isolated zeros. In fact, the significant coefficients are always the same ones, and do not depend on the definition of the trees. The difference in efficiency between various tree structures is only due to the gathering of the zeros.

There are three different regular tree structures to chose from.

TABLE II
NUMBER OF COEFFICIENTS FOR A 3-D TREE STRUCTURE AT EACH BITPLANE

Bit plane	Significant	IZ	ZTR	Average
19	99	0	745	19704.65
18	183	0	997	14724.05
17	249	2	1195	12263.84
16	610	94	1960	7146.76
15	1143	264	3595	3803.81
14	2975	975	9619	1385.41
13	9052	2678	28191	475.27
12	25525	6797	66564	199.76
11	64646	14587	122718	106.44
10	141869	29824	189039	66.43
9	276589	56712	268876	44.24
8	499196	104615	372197	29.74
7	860482	189221	517928	19.54
6	1461127	376246	718656	12.07
5	2439789	691967	926441	7.56
4	3940555	1299399	1015978	4.64
3	6053237	2130317	1098716	2.67
2	8950166	2860847	800219	1.57
1	11795923	2079228	341052	1.19
0	13673387	777902	156241	1.08

TABLE III
NUMBER OF COEFFICIENTS FOR A SPATIAL TREE STRUCTURE AT EACH BITPLANE

Bit plane	Significant	IZ	ZTR	Average
19	99	0	14534	1010.04
18	183	0	14702	998.50
17	249	2	14842	988.94
16	610	41	15704	932.32
15	1143	294	17256	836.41
14	2975	1877	22564	600.51
13	9052	10995	35115	318.17
12	25525	39488	53933	156.87
11	64646	95909	78453	83.82
10	141869	179019	112395	49.89
9	276589	297178	164097	31.23
8	499196	480696	245862	19.52
7	860482	787705	358326	12.06
6	1461127	1323093	473480	7.36
5	2439789	2019657	543211	4.78
4	3940555	2783606	543079	3.23
3	6053237	3597503	583521	2.06
2	8950166	4056970	316510	1.31
1	11795923	2557907	111127	1.08
0	13673387	918152	56742	1.03

- Use only spatial relationship between coefficients: spatial trees.
- Use only spectral relationship between coefficients: spectral trees.
- Use both relationships between coefficients: 3-D trees.

The optimal choice is not straightforward. Including more zero coefficients in one tree potentially allows more coefficients to be encoded with a single symbol, but also increases the risk to find a significant symbol destroying the tree.

Statistics computed on a transformed hyperspectral image are presented on Tables II–IV. Tables present the number of significant coefficients for each bitplane (these coefficients have to be coded and are independent of the choice of the tree structure), the number of isolated zero and the number of zerotrees. For a given bitplane, a certain amount of coefficients are set to 0. The purpose of zerotree structure is to encode most of these 0 coefficient in one zerotree using one symbol (ZTR in EZW terminology detailed later). 0 coefficients which cannot be included in zerotrees are coded with a single symbol (IZ). The two different symbols, ZTR and IZ, are used to code the 0 coefficient. The purpose of the tree structure is to minimize the number of symbols used or to maximize the number of 0 coefficient encoded within one symbol. In order to compare the different tree structures, the average number of 0 coefficients coded by ZTR+IZ is provided on Tables II–IV.

From these tables, it appears that the choice of the 3-D structure is more efficient and uses less symbols to code the 0 coefficients at any bitplane. This corresponds to the tree structure illustrated in Fig. 12. This structure is different from the one defined in [16] (illustrated in Fig. 11).

C. Embedded Zerotree of Wavelet Coefficients (EZW)

The critical part in bit plane encoding is the encoding of the *significance map*, i.e., the binary decision as to whether a sample is significant with respect to a given threshold T_k. EZW algorithm provides an efficient way of encoding the significant map.

In the general case, with the optimal tree structure found in the previous section, one coefficient, $c_{x,y,\lambda}$, has two spectral, $c_{x,y,2\lambda}$ and $c_{x,y,2\lambda+1}$, and four spatial children, $c_{2x,2y,\lambda}$, $c_{2x+1,2y,\lambda}$, $c_{2x,2y+1,\lambda}$ and $c_{2x+1,2y+1,\lambda}$. The lower frequency subbands, either spatial or spectral, are special cases which differ between EZW and SPIHT.

Each magnitude bit plane is encoded in two passes. The first pass, called the *refinement pass*, encodes a bit for every coefficient that is significant, where the significance of the coefficient has been declared by a previous pass. The second pass, called the *significance pass*, encodes the significance map of the coefficients for the current magnitude bit plane. Zerotree structures help to reduce the cost of the significance map encoding by using self-similarities between the subbands.

As shown by Shapiro, it is useful to encode the sign of significant coefficients along with the significant map. In practice, four different symbols are used: Zero Tree Root (ZTR), Isolated Zero (IZ), Positive significant (POS) and Negative significant (NEG). Each of these four symbols can be coded using two bits. IZ symbol indicates that the current coefficient is below the current threshold and that at least one of its descendants that has not been processed in the significant pass is above the threshold. ZTR symbol indicates that the current coefficient is below the current threshold and all its descendants are also below the threshold or already noticed as significant (thus

TABLE IV
NUMBER OF COEFFICIENTS FOR A SPECTRAL TREE STRUCTURE AT EACH BITPLANE

Bit plane	Significant	IZ	ZTR	Average
19	99	0	458752	32.00
18	183	0	458752	32.00
17	249	0	458755	32.00
16	610	57	459065	31.97
15	1143	124	459261	31.95
14	2975	566	459575	31.90
13	9052	1251	460365	31.78
12	25525	2659	462871	31.48
11	64646	4930	472529	30.61
10	141869	12290	501134	28.31
9	276589	32602	568224	23.97
8	499196	84838	696563	18.15
7	860482	210635	895402	12.49
6	1461127	511498	1145031	7.98
5	2439789	1016474	1336894	5.20
4	3940555	1751396	1346870	3.47
3	6053237	2510335	1216711	2.31
2	8950166	3104591	763072	1.48
1	11795923	2154989	323541	1.16
0	13673387	804094	142971	1.06

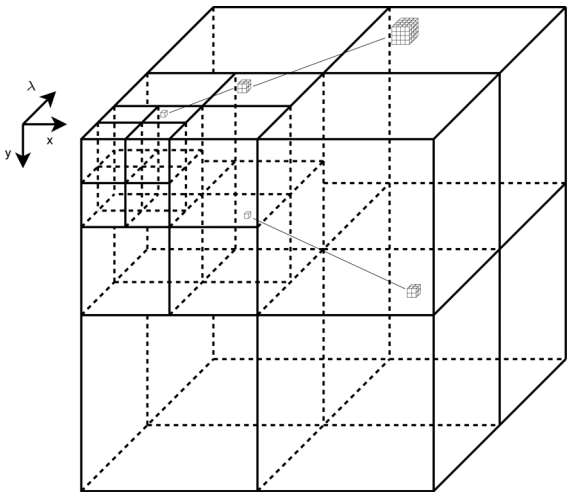


Fig. 11. Parent-child dependencies of subbands for the classical isotropic case.

processed during the refinement pass). It is important to note that coefficients already noticed as significant will be processed during the next refinement pass and could be considered as a part of a zerotree. Particular attention should be paid during the definition of the EZW algorithm as one given coefficient can have spatial and spectral parents, leading to tree crossings as we can see in Fig. 13. This phenomenon is particular to the tree structure used here.

The scanning of the coefficients is done by subbands in such a way that no child node is scanned before any of its parents. Within each subband, coefficients are visited in raster order, beginning with the subband with the lowest spatial and spectral frequency and continuing with subbands of higher frequency.

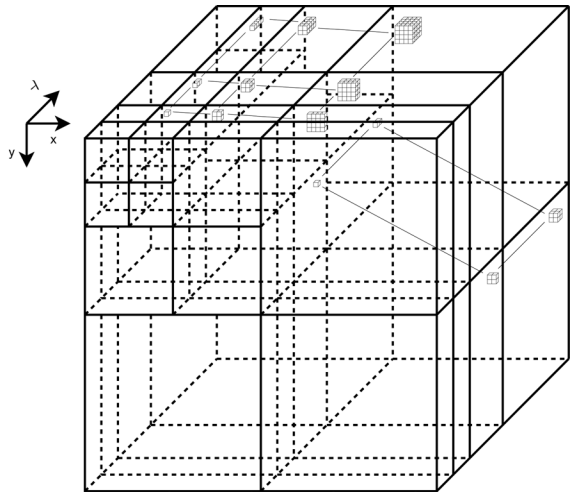


Fig. 12. Parent-child dependencies of subbands. Note that spectral and spatial descendants interfere, leading to a redundant relationship. The parent-child relation is no longer single-parent.

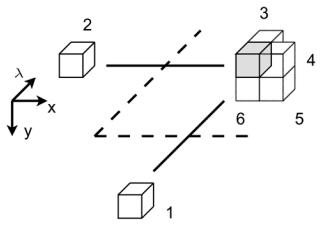


Fig. 13. Example of a tree crossing: the coefficient in grey can be considered as the spectral child of coefficient 1 together with coefficient 3 or as the spatial child of coefficient 2 together with 4, 5, and 6.

TABLE V
EZW PERFORMANCES ON MOFFETT3 (PSNR). 3-D CORRESPONDS TO THE USE OF THE 3-D TREE STRUCTURE, SPAT TO THE USE OF THE POPULAR SPATIAL TREE STRUCTURE, AC DENOTES THE USE OF AN ARITHMETIC CODER. THE 3-D TREE STRUCTURE BRINGS A CLEAR IMPROVEMENT

	1.0 bpppb	0.5 bpppb
EZW-3D	73.77 dB	67.98 dB
EZW-3D-AC	75.54 dB	69.25 dB
EZW-SPAT	71.94 dB	66.31 dB
EZW-SPAT-AC	74.87 dB	68.84 dB

Coding is done by a 4-ary alphabet of the previously defined symbols (POS, NEG, ZTR, and IZ).

Performance on the image *moffett3* are detailed in Table V for the 3-D tree structure as well as for the more popular spatial tree structure. An optional arithmetic coder as described in [28] can be applied on the EZW stream. This coder is available for research purposes at http://www.cs.mu.oz.au/~alistair/arith_coder/. The arithmetic coder (for the results denoted AC) is applied directly on the EZW stream and does not consider any context. As we can see from Table V, the 3-D structure leads to a clear improvement compared to the spatial tree.

D. Adaptation of SPIHT

The SPIHT algorithm was first described by Said and Pearlman in [32]. The main properties of EZW are preserved: progressivity and low-complexity. Some differences, however, lead to improved performance for classical 2-D images. The

TABLE VI
SPIHT PERFORMANCES ON MOFFETT3 (PSNR). 3-D CORRESPONDS TO THE USE OF THE 3-D TREE STRUCTURE, SPAT TO THE USE OF THE POPULAR SPATIAL TREE STRUCTURE, AC DENOTES THE USE OF AN ARITHMETIC CODER. THE TREE-CROSSING PHENOMENON CAUSES MORE PROBLEM HERE THAN IN THE CASE OF EZW, THE 3-D TREE STRUCTURE DOES NOT PERFORM PROPERLY

	1.0 bpppb	0.5 bpppb
SPIHT-3D	73.55 dB	68.35 dB
SPIHT-3D-AC	73.89 dB	68.83 dB
SPIHT-spat	75.74 dB	69.97 dB
SPIHT-spat-AC	75.92 dB	70.05 dB

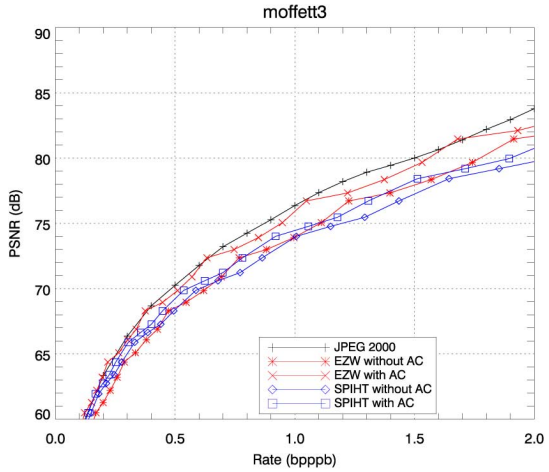


Fig. 14. Comparison of compression performance between JPEG 2000 and EZW (resp. SPIHT) with and without arithmetic coder for the 3-D tree.

SPIHT algorithm maintains three lists of coefficients: List of Significant Pixels (LSP), List of Insignificant Pixels (LIP) and the List of Insignificant Sets (LIS). $\mathcal{O}(x,y,\lambda)$ represents the children of (x,y,λ) (only one level), $\mathcal{D}(x,y,\lambda)$ are all the descendants and $\mathcal{L}(x,y,\lambda) = \mathcal{D}(x,y,\lambda) - \mathcal{O}(x,y,\lambda)$ (descendants without the children). The function $S_k(x,y,\lambda)$ is 0 if all the descendants of (x,y,λ) are below T_k (zerotree) and 1 otherwise.

The first main difference to be noted with the EZW algorithm is the fact that every output is binary. The second difference is that the coefficient order is more dependent on the data. Whereas coefficient are processed in a raster order for EZW in each subband, the list system of SPIHT leaves the order entirely dependent on the data. Coefficients are processed according to their order in the list. The parent-child relation is also different in the case of the lower subband. In the 2-D case, for EZW, each coefficient from the lower subband (LL) has three descendants (in LH, HL, and HH). For SPIHT, one out of four coefficients from the lower subband has no child whereas the three others have four children. The definition is also slightly different as SPIHT considers two different types of zerotree: type A where all the descendants are not significant and type B where all descendants *except the children* are not significant. It has to be noted that the zerotree root can be significant in both cases.

On hyperspectral images, with the previously defined parent-child relationship, the SPIHT algorithm works as follows.

Algorithm 2: Anisotropic SPIHT

Initialization step:

• $k = 0$

• $LSP = \emptyset$

• LIP: all the coefficients without any parents (coefficient from the LLL)

• LIS: all coefficients from the LIP with descendants (as type A)

Sorting pass:
For each entry (x,y,λ) of the LIP

• Output $S_k(x,y,\lambda)$

• If $S_k(x,y,\lambda) = 1$, move (x,y,λ) in LSP and output the sign of $c_{x,y,\lambda}$

For each entry (x,y,λ) of the LIS

• If the entry is type A

— Output $S_k(\mathcal{D}(x,y,\lambda))$

— If $S_k(\mathcal{D}(x,y,\lambda)) = 1$ then

$*$ For all $(x',y',\lambda') \in \mathcal{O}(x,y,\lambda)$: output $S_k(x',y',\lambda')$; if $S_k(x',y',\lambda') = 1$, add (x',y',λ') to the LSP and output the sign of $c_{x',y',\lambda'}$ else, add (x',y',λ') to the end of the LIP. Critical point for tree-crossing !

$*$ If $\mathcal{L}(x,y,\lambda) \neq \emptyset$, move (x,y,λ) to the end of the LIS as a type B entry

Else, remove (x,y,λ) from the LIS

• If the entry is type B

— Output $S_k(\mathcal{L}(x,y,\lambda))$

— If $S_k(\mathcal{L}(x,y,\lambda)) = 1$

$*$ Add all the $(x',y',\lambda') \in \mathcal{O}(x,y,\lambda)$ to the end of the LIS as a type A entry

$*$ Remove (x,y,λ) from the LIS

Refinement pass:

• For all entries (x,y,λ) of the LSP except those included in the last sorting pass

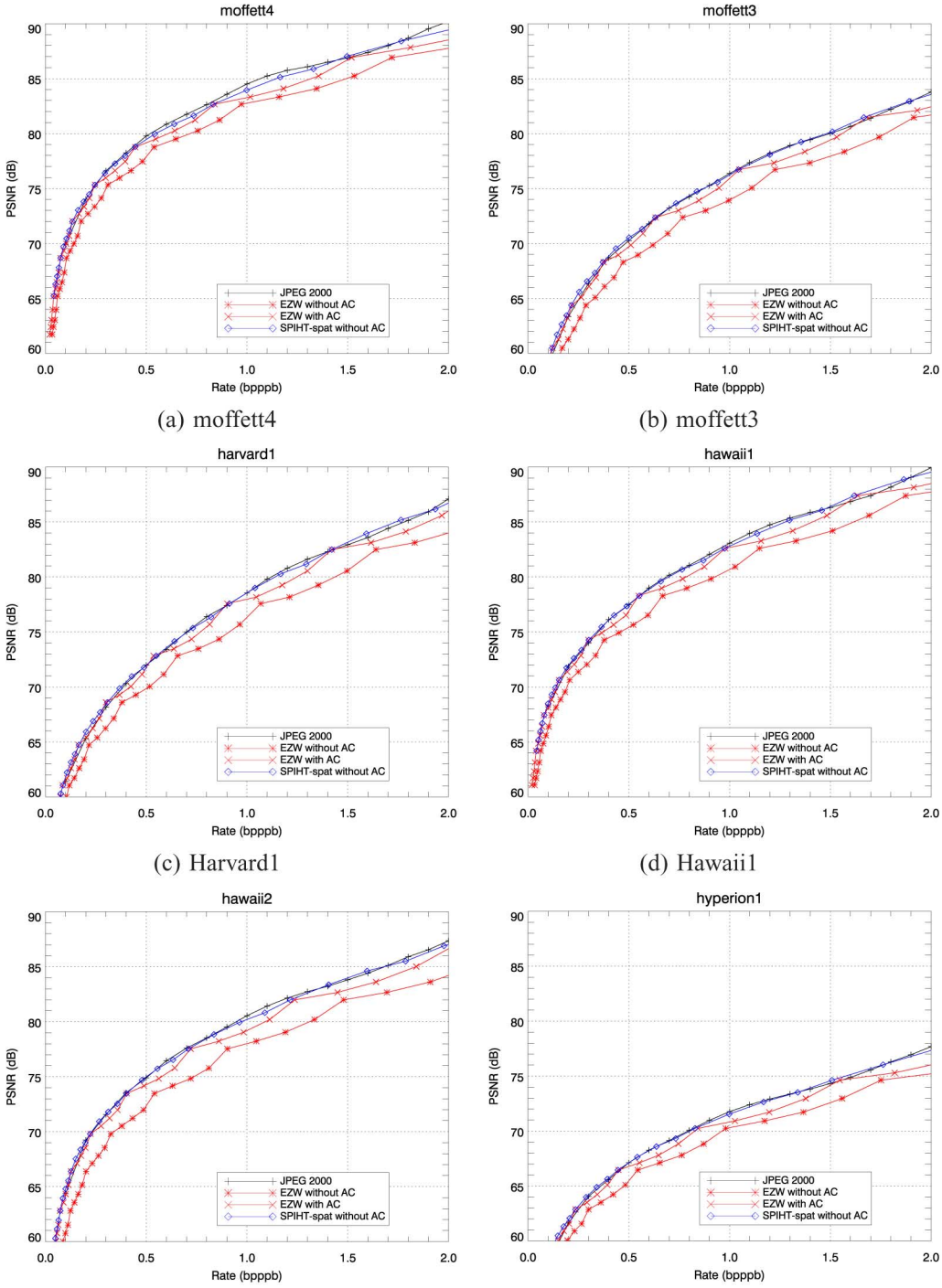
• Output the kth most significant bit of $c_{x,y,\lambda}$

Increment of k and return to the sorting pass. ■

The decoder is obtained replacing *output* by *input* in the previous algorithm.

The tree-crossing phenomenon has a greater impact on SPIHT than on EZW. During the algorithm, one coefficient can be processed before one of his parent has been seen by the algorithm, special care has to be taken not to process the same coefficient twice.

The same arithmetic coder than in the EZW case can be used on the SPIHT stream.



(a) moffett4 (b) moffett3

(c) Harvard1 (d) Hawaii1

(e) Hawaii2 (f) hyperion1

Fig. 15. Results on six different hyperspectral data sets. SPIHT2 on the spatial tree structure provides performance very similar to JPEG 2000 in all the different situations. The raw arithmetic coder does not improve the results significantly for SPIHT. EZW does not perform as well, but with the arithmetic coder and for the optimal truncation point, performances are almost the same as JPEG 2000 and SPIHT.

From Table VI, we can observe that, surprisingly, the 3-D tree structure performs lower than expected and does not bring any improvement compared to the spatial tree. This is due to the tree crossing phenomenon which is more problematic in SPIHT case. In Fig. 14, performances of SPIHT using the 3-D tree are illustrated whereas on Fig. 15, SPIHT2 corresponds to a coder using the spatial tree structure.

V. Results

JPEG 2000 with the multicomponent wavelet transform as well as the adaptation of EZW and SPIHT algorithms are applied to hyperspectral data cubes from NASA/JPL airborne sensor AVIRIS and from NASA spaceborne sensor Hyperion. The chosen images are presented in Fig. 8. One of the dataset is the same as the one used in [16].

Comparisons of our proposed utilization of JPEG 2000 (presented in Section II) and the adaptation of [33] show similar performance. The proposed adaptations of EZW and SPIHT algorithms presented above are compared with JPEG 2000. The results obtained with JPEG 2000 consists in one compression step using several quality layers as defined in [6]. 20 quality layers are set from 0.1 to 2.0 bpppb. The decompression is done using the corresponding quality layers. EZW and SPIHT performances are plotted both with and without arithmetic coder. Performances are plotted in term of PSNR (1). The image is fully compressed with the bitstream saved on disk and then the bitstream is decoded writing the output image on disk. PSNR is measured between the original image and the output image.

EZW performs significantly better when the decompression reaches the end of a bit-plane. This can be seen by the inflexion point in Fig. 14. This particularity is not so visible for SPIHT. When used with an arithmetic coder, EZW gives similar results as JPEG 2000 and even outperforms it at some bitrate. Performance of SPIHT is surprisingly low at high bitrate and tends to improve over EZW at small bitrates. This can be explained by the fact that SPIHT used with the 3-D tree structure is particularly sensitive to the tree-crossing effect.

One simple solution to avoid the tree-crossing effect consists in using the spatial tree defined before. This modification, leading to a structure similar to the one found in [19], yields improved results as shown in Fig. 15. In this case, the arithmetic coder provides only a limited improvement; thus, the results are presented without arithmetic coder to further reduce the coder complexity. However, this solution is not fully satisfactory as one can feel that the spectral relationship is not fully used. Yet, keeping the 3-D tree structure would require an important modification of the SPIHT algorithm to make full use of the tree-structure.

VI. Conclusion

In this paper, different hyperspectral image compression algorithms are proposed and compared. First, a method is defined to find the optimal (in a rate-distortion sense) 3-D anisotropic wavelet decomposition for hyperspectral images. This method leads to the justification for the use of a particular fixed transform. On this decomposition, several tree structures are possible. A statistical study on the proportion of zerotree is made on the different structures, leading to the choice of a 3-D tree. Then, different compression algorithms based on this wavelet transform and this tree are defined. The first method is based on an adaptation of the EZW algorithm. The second one is an adaptation of the well-known SPIHT. Their results are compared to the JPEG 2000 standard. Even if the rate-distortion optimization included in JPEG 2000 makes it difficult to be considered for on-board implementation on spacecraft, its performance can be seen as an upper bound to be reached.

This study demonstrates the potential of using zerotrees of wavelet coefficients in the compression of hyperspectral images. With a low complexity, EZW manages to give performance similar to those given by JPEG 2000, while providing a fully embedded bit stream. SPIHT performs well, even if used without arithmetic coder. These properties are particularly appealing for the on-board processing of hyperspectral images on space systems. Further improvement of the presented algorithm have since been developed on EZW [34], [35] and on SPIHT [36], [37]. Making full use of the double linked tree structure or adding an adapted arithmetic coder would yield improved results.

Of course, before using such a compression algorithm for a real mission, an extensive quality study, beyond the scope of the present paper, needs to be performed accordingly to the mission objectives and sensor characteristics.

Acknowledgment

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

References

[1] R. E. Roger and J. F. Arnold, "Reversible image compression bounded by noise," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 1, pp. 19–24, Jan. 1994.

[2] E. Christophe, "Compression Des Images Hyperspectrales Et Son Impact Sur La Qualité Des Données," Ph.D., École Nationale Supérieure de l'Aéronautique et de l'Espace, France, 2006.

[3] Y. Langevin and O. Forni, "Image and spectral image compression for four experiments on the ROSETTA and mars express missions of ESA," in *Proc. SPIE Applications of Digital Image Processing XXIII*, 2000, vol. 4115, pp. 364–373.

[4] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Boston, MA: Kluwer, 2002.

[5] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 18, no. 5, pp. 36–58, Sep. 2001.

[6] Information Technology—JPEG 2000 Image Coding System: Core Coding System 2002, ISO/IEC Std. 15444-1.

[7] Information Technology—JPEG 2000 Image Coding System: Extensions 2004, ISO/IEC Std. 15444-2.

[8] JP3D Working Draft V5.0 2004, ISO/IEC Std. WG1N3467.

[9] E. Christophe, D. Léger, and C. Mailhes, "Quality criteria benchmark for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 5, pp. 2103–2114, Sep. 2005.

[10] JPEG2000 Verification Model 9.1 (Technical description) 2001, ISO/IEC Std. WG1N2165.

[11] H. S. Lee, N. H. Younan, and R. L. King, "Hyperspectral image cube compression combining JPEG-2000 and spectral decorrelation," in *Proc. IEEE Int. Geoscience and Remote Sensing Symp.*, Jun. 2002, vol. 6, pp. 3317–3319.

[12] Q. Du and J. E. Fowler, "Hyperspectral image compression using JPEG2000 and principal component analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 4, pp. 201–205, Apr. 2007.

[13] Q. Du and J. E. Fowler, "Low-complexity principal component analysis for hyperspectral image compression," *Int. J. High Performance Comput. Appl.*, to be published.

[14] D. Van Buren, "A high-rate JPEG2000 compression system for space," in *Proc. IEEE Aerospace Conf.*, Mar. 2005, pp. 1–7.

[15] P.-S. Yeh, P. Armbruster, A. Kiely, B. Masschelein, G. Moury, C. Schaefer, and C. Thiebaut, "The new CCSDS image compression recommendation," presented at the IEEE Aerospace Conf., Mar. 2005.

[16] X. Tang, W. A. Pearlman, and J. W. Modestino, "Hyperspectral image compression using three-dimensional wavelet coding," in *Proc. SPIE Image and Video Communications and Processing*, 2003, vol. 5022, pp. 1037–1047.

[17] H. Kim, C. Choe, and J. Lee, "Fast implementation of 3-D SPIHT using tree information matrix," in *Proc. IEEE Int. Geoscience and Remote Sensing Symp.*, Jul. 2003, vol. 6, pp. 3586–3588.

[18] X. Tang and W. A. Pearlman, "Scalable hyperspectral image coding," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2005, vol. 2, pp. 401–404.

[19] C. He, J. Dong, and Y. F. Zheng, "Optimal 3-D coefficient tree structure for 3-D wavelet video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 10, pp. 961–972, Oct. 2003.
[20] S. Cho and W. A. Pearlman, "Error resilient video coding with improved 3-D SPIHT and error concealment," in *Proc. SPIE Image and Video Communications and Processing*, 2003, vol. 5022, pp. 125–136.
[21] R. R. Coifman, Y. Meyer, S. Quake, and M. V. Wickerhauser, "Signal processing and compression with wavelet packets," Tech. Rep., Numerical Algorithms Research Group, Yale Univ., New Haven, CT, 1990.
[22] E. Christophe, C. Mailhes, and P. Duhamel, "Best anisotropic 3-D wavelet decomposition in a rate-distortion sense," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May 2006, vol. 2, pp. II–17.
[23] D. Xu and M. N. Do, "Anisotropic 2D wavelet packets and rectangular tiling: theory and algorithms," in *Proc. SPIE, Wavelets: Applications in Signal and Image Processing X*, Nov. 2003, vol. 5207, pp. 619–630.
[24] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inf. Theory*, vol. 38, no. 3, pp. 713–718, Mar. 1992.
[25] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. Image Process.*, vol. 2, no. 2, pp. 160–175, Apr. 1993.
[26] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 9, pp. 1445–1453, Sep. 1988.
[27] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, Jul. 1989.
[28] A. Moffat, R. Neal, and I. H. Witten, "Arithmetic coding revisited," *ACM Trans. Inf. Syst.*, vol. 16, no. 3, pp. 256–294, 1998.
[29] J. E. Fowler and J. T. Rucker, *3D Wavelet-Based Compression of Hyperspectral Imagery, in Hyperspectral Data Exploitation: Theory and Applications*, C. I. Chang, Ed. Hoboken, NJ: Wiley, 2007, ch. 14, pp. 379–407.
[30] X. Tang, S. Cho, and W. A. Pearlman, "3D set partitioning coding methods in hyperspectral image compression," in *Proc. Int. Conf. Image Processing*, 2003, vol. 2, pp. 239–242.
[31] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.
[32] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 243–250, Jun. 1996.
[33] J. T. Rucker, J. E. Fowler, and N. H. Younan, "JPEG2000 coding strategies for hyperspectral data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2005, vol. 1, pp. 128–131.
[34] E. Christophe, P. Duhamel, and C. Mailhes, "Signed binary digit representation to simplify 3D-EZW," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Apr. 2007, vol. 1, pp. 1025–1028.
[35] E. Christophe, P. Duhamel, and C. Mailhes, "Adaptation of zerotrees using signed binary digit representations for 3 dimensional image coding," *EURASIP J. Image Video Process.*, 2007.
[36] E. Christophe and W. A. Pearlman, "Three-dimensional SPIHT coding of hyperspectral images with random access and resolution scalability," in *Proc. 40th Annu. Asilomar Conf. Signals, Systems, and Computers*, Oct. 2006, pp. 1897–1901.
[37] E. Christophe and W. A. Pearlman, "Three-dimensional SPIHT coding of volume images with random access and resolution scalability," *EURASIP J. Image Video Process.*, 2008.

Emmanuel Christophe (M'06) received the Engineering degree in Telecommunications from ENST Bretagne, France, the DEA in telecommunications and image processing from the University of Rennes 1, France, in 2003, and the Ph.D. degree in image processing from Supaero, France, in 2006.

He was a Visiting Scientist at the National University of Singapore, Singapore, in 2003, and at the Rensselaer Polytechnic Institute, Troy, NY, in 2006. Since 2006, he has been a Research Engineer for CNES (French Space Agency) where he is developing image processing algorithms for efficient exploitation of high-resolution remote sensing images. His research interests include image and video compression, object recognition, as well as technologies for remote sensing.

Corinne Mailhes (M'87) was born in France in 1965. She received the Eng. degree in electronics and signal processing and the Ph.D. degree in signal processing from the University of Toulouse (ENSEEIHT, France) in 1986 and 1990, respectively.

She is currently a Professor with the University of Toulouse (ENSEEIHT) and a member of the IRIT Laboratory (UMR 5505 of the CNRS) and of TeSA Lab (http://www.tesa.prd.fr). Her research activities are centred on statistical signal processing, with particular interests in spectral analysis, data compression, and biomedical signal processing.

Dr. Mailhes was member of the Organizing Committee for ICASSP 2006, held in Toulouse.

Pierre Duhamel (F'98) was born in France in 1953. He received the Eng. degree in electrical engineering from the National Institute for Applied Sciences (INSA) Rennes, France, in 1975, the Dr. Eng. degree in 1978, and the Doctorat ès sciences degree in 1986, both from Orsay University, Orsay, France.

From 1975 to 1980, he was with Thomson-CSF, Paris, France, where his research interests were in circuit theory and signal processing, including digital filtering and analog fault diagnosis. In 1980, he joined the National Research Center in Telecommunications (CNET), Issy les Moulineaux, France, where his research activities were first concerned with the design of recursive CCD filters. Later, he worked on fast algorithms for computing Fourier transforms and convolutions, and applied similar techniques to adaptive filtering, spectral analysis and wavelet transforms. From 1993 to September 2000, he was a Professor at ENST, Paris (National School of Engineering in Telecommunications), with research activities focused on signal processing for communications. He was head of the Signal and Image processing Department from 1997 to 2000. He is now with CNRS/LSS (Laboratoire de Signaux et Systemes, Gif sur Yvette, France), where he is developing studies in signal processing for communications (including equalization, iterative decoding, multicarrier systems, cooperation) and signal/image processing for multimedia applications, including source coding, joint source/channel coding, watermarking, and audio processing. He is currently investigating the application of recent information theory results to communication theory.

Dr. Duhamel was Chairman of the DSP committee from 1996 to 1998, and a member of the SP for Com committee until 2001. He was an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 1989 to 1991, an Associate Editor for the IEEE SIGNAL PROCESSING LETTERS, and a Guest Editor for the special issue on wavelets of the IEEE TRANSACTIONS ON SIGNAL PROCESSING. He was Distinguished Lecturer, IEEE, for 1999, and was Co-General Chair of the 2001 International Workshop on Multimedia Signal Processing, Cannes, France. He was also Co-Technical Chair of ICASSP 2006, Toulouse, France. The paper on subspace-based methods for blind equalization, which he coauthored, received the "Best paper award" from the IEEE TRANSACTIONS ON SIGNAL PROCESSING in 1998. He was awarded the "grand prix France Telecom" by the French Science Academy in 2000.