Zerotree-based Compression Algorithm for Spaceborne Hyperspectral Sensor

CIPR Technical Report TR-2008-4

E. Christophe, C. Thiebaut, W. A. Pearlman, C. Latry, and D. Lebedeff

On-Board Payload Data Compression Workshop – OBPDC
June 2008

Center for Image Processing Research

Rensselaer Polytechnic Institute
Troy, New York 12180-3590
http://www.cipr.rpi.edu

Zerotree-based Compression Algorithm for Spaceborne Hyperspectral Sensor

E.Christophe[(1)], C. Thiebaut[(1)], W. A. Pearlman[(2)],
C. Latry[(1)] and D. Lebedeff[(3)]

[(1)]CNES
18 av. E. Belin
31401 Toulouse Cedex 9, France
emmanuel.christophe@cnes.fr

[(2)]Rensselaer Polytechnic Institute
Troy, NY 12180-3590 USA, USA

[(3)]Thales Alenia Space
100 Blvd. du midi BP. 99
06156 Cannes-la-Bocca Cedex, France

1. INTRODUCTION

The compression step 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 guaranty the full preservation of the scientific value of data. However, lossless compression techniques are limited to compression ratios of about two or three given the noise inherently present in the data.

Hyperspectral images have some specific properties 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. This presentation focuses on the full optimization of a complete compression algorithm for hyperspectral images. Zerotree based compression algorithms (EZW, SPIHT) are among the best for image compression and are compatible with space constraints (some implementation are already flying with adaptations of these algorithms). With the increase of remote sensing images, fast access to some features of the image is becoming critical. This access could be some part of the spectrum, some area of the image, high spatial resolution. These properties are critical for on-board applications. Reduction of the memory print necessary during the compression is also important.

The compression algorithm is optimized starting from the 3D wavelet transform, including the zerotree definition and to the bitstream construction to include scalability and random access properties[1]. Performances are compared with the adaptation of JPEG 2000 for hyperspectral images: extensions of the JPEG 2000 standard are used to produce the best possible results.

2. GETTING THE MAXIMUM FROM JPEG 2000

The JPEG 2000 standard provide state of the art performances for image compression[2]. Parts 1 of the JPEG 2000 standard ([3]) is 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[4], however, makes provisions for arbitrary spectral decorrelation, including wavelet transform. We propose to use the extensions of the JPEG 2000 standard by introducing transform on the spectral components before applying the JPEG 2000 encoder.

Four different possibilities for multicomponent transform are compared:

- No transform
- Discrete Cosine Transform (DCT)
- Karhunen-Loeve Transform (KLT)

- Wavelet transform (DWT)

In these cases, the 1D transform is applied in the spectral dimension before JPEG 2000 encoding of each resulting image.

Fig. 1 compares the impact of these transforms in terms of Peak Signal to Noise Ratio (PSNR) (Eq. 1) as a function of bit per pixel per band (bpppb). PSNR is probably not the best measure for hyperspectral image quality [5] but it is commonly admitted and easy to compute.

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

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 spectral decorrelation prior to the JPEG 2000 encoder greatly improves the results compared to the basic algorithm. The KLT depends on the image data and requires more complex operations: computation of the covariance matrix and eigenvectors. This high complexity is a deterrent for on-board implementation, especially when the number of components increases. Further investigation showed that using a pre-computed KLT on several data sets does not provide good performance as those obtained on multispectral images [6]. 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.

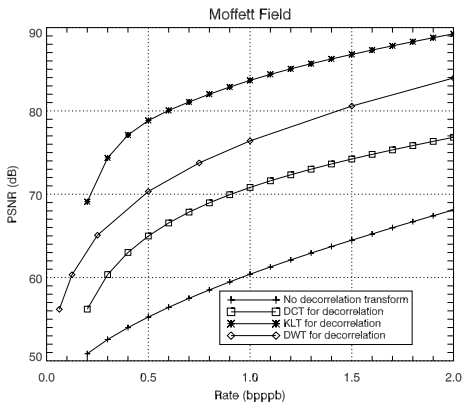


Figure 1. Comparison of multicomponent transform with the JPEG 2000 compression standard using either no interband decorrelation, DCT, KLT or DWT decorrelation on the Moffett Field data set.

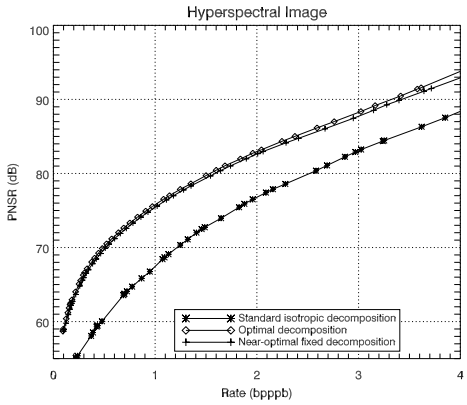


Figure 2. Results on hyperspectral data, the anisotropic best-basis clearly improves the performance. The fixed anisotropic decomposition performs closely to the optimal decomposition.

To the authors knowledge, there is only one hardware implementation of JPEG 2000 designed for space constraints [7]. In his paper, Van Buren admits that his solution performs significantly lower than Kakadu in the lossy case. The Consultative Committee for Space Data Systems (CCSDS), a working group gathering the main space agencies (NASA, JAXA, ESA, CNES), issued some recommendations for on-board image compression systems in [8]. 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 on-board 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.

3. BEST 3D WAVELET TRANSFORM

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 3D decomposition [9, 10], however, as mentioned before, hyperspectral data are clearly not isotropic. In the domain of video processing

some anisotropic structure has also been successfully used [11, 12]. 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.

The problem of finding the optimal wavelet decomposition for 1D signal has been explored in several publications [13]. For natural 2D 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 [14]. Several criteria were used to characterize the optimal decomposition: entropy-based algorithms [15] or rate-distortion compromise [16] for example. The main advantage of the latest is that it provides simultaneously the bit allocation between the different subbands [17].

An optimal decomposition search algorithm is defined in [18]. The results presented on Fig. 2 are obtained on data from the AVIRIS hyperspectral sensor from JPL/NASA over the Moffett Field site in California and remain valid on different sites as well as on images acquired by the satellite sensor from NASA: Hyperion.

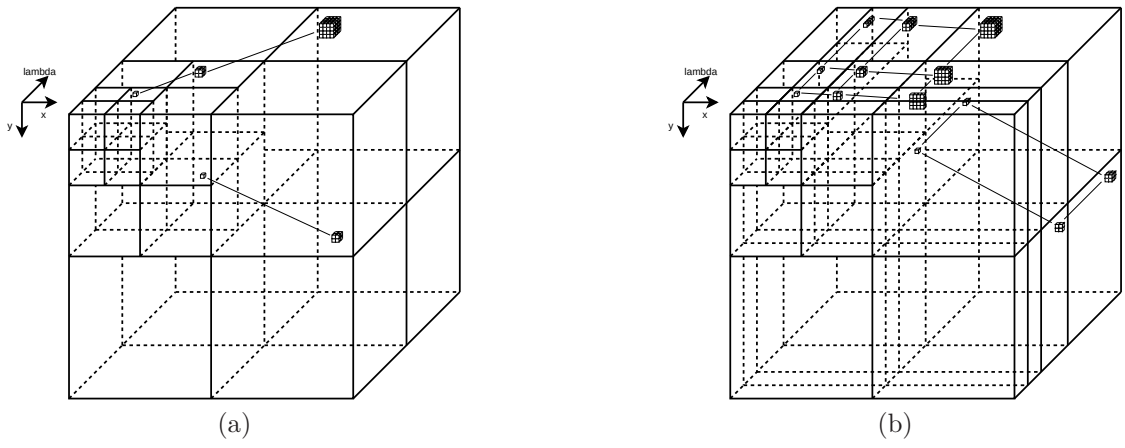
As can be seen from Fig. 2, the best basis decomposition brings a clear improvement, leading to an increase of the quality of 8 dB compared to the isotropic decomposition. If the limit is fixed in term of quality, let say for example a PSNR greater than 70 dB, the necessary bit budget cuts down from 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.

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 1D multiresolution wavelet and then using the standard multiresolution decomposition on the resulting components.



(a) (b)

Figure 3. Isotropic 3D decomposition (a) and nearly optimal anisotropic decomposition (b) with 3 levels of decomposition (simulation are done with 5 levels of decomposition).

The resulting decomposition and the resulting coefficients values are illustrated on Fig. 3. The results are illustrated here on the Moffett data cube but were confirmed on various hyperspectral images with very different characteristics (sea, forest, mineral, cities) and from different sensors (AVIRIS and Hyperion).

These results are also *a posteriori* justification for the transform used in [19] in the context of hyperspectral image 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 2.

On-the-flow hardware implementation of the wavelet transform are well-known and the adaptation of this 3D wavelet transform will not present any specific difficulty. However, the processing order has to be adapted to the acquisition process as illustrated on Fig. 4.

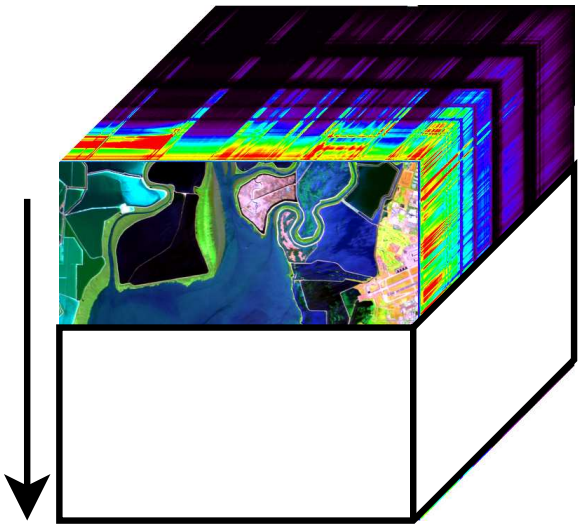


Figure 4. Acquisition process: as the second spatial dimension is acquired through the satellite motion, on-the-flow wavelet transform should process the spectral dimension first.

4. TWEAKING SPIHT

4.1. Local coding

One of the difficulty in adapting SPIHT [20] for hardware implementation is the unpredictable scan order of coefficients. Random coefficient access is particularly difficult problem for hardware implementation in a matrix containing millions of values. However the classical representation of the wavelet coefficient is misleading, especially when on-the-flow processing is performed. Coefficient corresponding to the same area in the data cube are actually gathered together as illustrated on Fig. 5 in 2D for simplicity.

This organization is a blessing for zerotree coding as all the tree coefficients are near each others. Coding separately these trees will ease the constraints and provide random access capabilities as is it detailed in [21]. Encoding separately portions of the image provides several advantages. First, scan-based mode compression is made possible as the whole image is not necessary. Secondly, encoding parts of the image separately also provides the ability to use different compression parameters for different parts of the image, enabling the possibility of high quality region of interest (ROI) and the possibility of discarding unused portions of the image. An unused portion of the image could be an area with clouds. Third, transmission errors have a more limited effect in the context of separate coding; the error only affects a limited portion of the image.

Direct transformation and coding different portions of the image results in poor coding efficiency and blocking artifacts visible at boundaries between adjacent portions. However, if we encode portions of the full image transform corresponding to image regions that together constitute the whole, coding efficiency is maintained and boundary artifacts vanish in the inverse transform process. This strategy has been used for this particular purpose on the EZW algorithm in [22] and in [23] for 3D-SPIHT in the context of video coding. Finally, one limiting factor of the SPIHT algorithm is the complicated list processing requiring a large amount of memory. If the processing is done only on one part of the transform at a a time, the number of coefficients involved is dramatically reduced and so is the memory necessary to store the control lists in SPIHT.

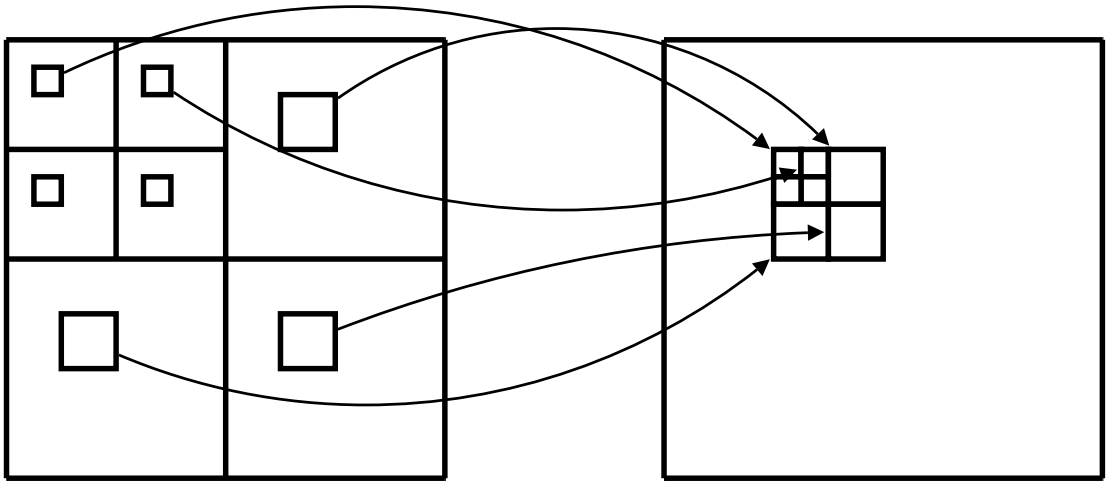


Figure 5. During on-the-flow wavelet transform, wavelet coefficients are not organized as the traditional wavelet representation show. Coefficients are gathered in groups corresponding to the same area in the data cube.

The granularity of the random access obtained with this method is very small. Spatially, the grain size is 2×2, compared to JPEG2000's 32×32 or 64×64, which are the typical sizes of the encoded subblocks of the subbands. Using subblocks smaller than 32×32 in JPEG2000 results in considerable loss of coding efficiency. JPEG2000 encodes the spectrally transformed slices or spectral bands independently, so its grain size in the spectral direction is 1 versus 2 for our method. With the $2 \times 2 \times 2$ root group, it is possible to retrieve almost only the required coefficients to decode a given area. Moreover every coefficient can be retrieved only to the bit plane necessary to give the expected quality.

4.2. Resolution scalability

The original SPIHT algorithm does not provide distinction between coefficients from different resolution levels. To provide resolution scalability, we need to provide the ability to decoded only the coefficients from a selected resolution. A resolution comprises 1 or 3 subbands. To enable this capability, we keep three lists for each resolution level r. When $r = 0$ only coefficients from the low frequency subbands will be processed. Resolution levels must be processed in increasing order, because to reconstruct a given resolution, all the lower order resolution levels are needed. Coefficients are processed according to the resolution level to which they correspond. For a 5-level wavelet decomposition in the spectral and spatial direction, a total of 36 resolution levels will be available. Each level r keeps in memory three lists: LSP_r, LIP_r and LIS_r. The interested reader can refer to [21] for a detailled description of the algorithm.

This adaptation of SPIHT, which processes all bit planes at a given resolution level, provides strictly the same amount of bits as the original SPIHT. The bits are just organized in a different order. With the block structure, memory footprint during compression is dramatically reduced. The resolution scalability with its several lists does not increase the amount of memory necessary as the coefficients are just spread onto different lists.

4.3. Rate allocation

The problem of processing different areas of the image separately always resides in the rate allocation for each of these areas. A fixed rate for each area is usually not a suitable decision as complexity most probably varies across the image. If quality scalability is necessary for the full image, we need to provide the most significant bits for one block before finishing the previous one. This could be obtained by cutting the bitstream for all blocks and interleaving the parts in the proper order. With this solution, the rate allocation will not be available at the bit level due to the block organization and the spatial separation, but a trade-off with quality layers organization can be used.

An algorithm similar to the PCRD-opt of JPEG 2000 can be used. However, several facts make it particularly easy to use in the case of 3D-SPIHT-RARS:

- as no further entropy coding is done on the output of the SPIHT algorithm, the exact rate is known at any point of the encoding process.

- as the encoding is done on bit plane and as the 9/7 wavelet transform is nearly orthogonal, image quality improvement impact is known for every single coding symbol.

4.4. Results

Performances are presented in terms of SNR for easy comparison with other publications and compared with JPEG 2000 with the multicomponent transform in table 1.

Table 1. SNR for popular data sets.

	Rate (bpppb)	2.0	1.0	0.5	0.1
Moffett sc3	JPEG2000 MT	54.90	48.63	43.52	33.16
	3D-SPIHT-RARS	54.07	47.84	42.49	32.28
Moffett sc1	JPEG2000 MT	51.99	45.48	40.18	29.75
	3D-SPIHT-RARS	50.87	44.27	39.32	28.82
Jasper sc1	JPEG2000 MT	51.60	44.85	39.31	28.99
	3D-SPIHT-RARS	50.52	43.55	38.34	28.03
Cuprite sc1	JPEG2000 MT	56.72	50.99	46.71	38.72
	3D-SPIHT-RARS	55.35	50.15	46.04	37.98

Performances of both algorithms are also compared in term of classification. The classification is compared before and after compression with a ground truth. The dataset used is the dataset from the DFTC Fusion Contest 2008 with the on-line classification evaluation system. The classification is done on the PCA transform keeping the 8 bands with most energy. The classification algorithm is a supervised SVM from the Orfeo Toolbox [24]. The same area are used for the SVM training. The classification performance on the original image is 83.19%. Results are presented in table 2.

Table 2. Classification performances with SVM method applied on the PCA keeping the 8 bands with the highest energy. The performance without compression is 83.19%

Rate (bpppb)	2.0	1.0	0.5	0.1
JPEG2000 MT	83.05%	88.38%	87.14%	78.29%
3D-SPIHT-RARS	82.32%	86.41%	**88.59%**	78.21%

It is worth noting that is some case, the classification performances after compression is better than on the original data. This is mostly due to the fact that most errors occur on area boundaries due to some residual noise in data. The best classification performance is obtained with the 3D-SPIHT-RARS algorithm at a 0.5 bpppb.

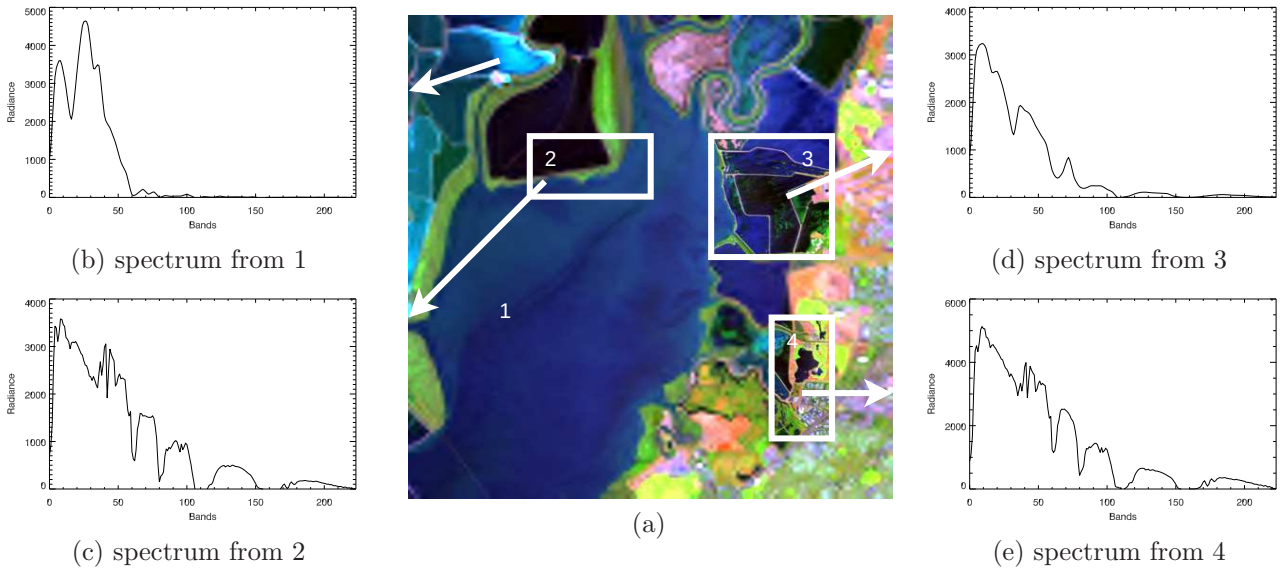
The final bitstream is particularly flexible as illustrated on Fig. 6.

5. CONCLUSION

This adaptation of SPIHT for hyperspectral images works relatively well in terms of SNR and very well in terms of classification. The hardware implementation is not done yet, but as a particular emphasis was on the algorithm simplification, it should be possible to comply with harsh space constraints. Other coding techniques can be used to further simplify the algorithm as in [25] and [26] were the signed binary digit representation prove to be valuable for EZW method.

ACKNOWLEDGMENTS

This work has been carried out under the financial support of *Centre National d'Etudes Spatiales* (CNES), TeSA, *Office National d'Etudes et de Recherches Aérospatiales* (ONERA) and *Thales Alenia Space*. Partial support was also provided by the Office of Naval Research under Award No. N0014-05-10507. The authors wish to thank their supporters and *NASA/JPL* for providing the hyperspectral images used during the experiments.



(b) spectrum from 1 (d) spectrum from 3

(c) spectrum from 2 (a) (e) spectrum from 4

Figure 6. Region-of-interest (ROI) decompression of an hyperspectral image with different spatial and spectral resolution for different areas. Background (area 1) is with low spatial resolution and low spectral resolution as is can be seen on the spectrum (b). Area 2 has low spatial resolution and highspectral resolution (c), area 3 has high spatial resolution but low spectral resolution (d). Finally, area 4 has both high spectral and spatial resolutions. This decompressed image was obtained from a generic bitstream, reading less than 5.5% of the compressed bit stream.

REFERENCES

1. E. Christophe, "Compression des images hyperspectrales et son impact sur la qualité des données," Ph.D. dissertation, École Nationale Supérieure de l'Aéronautique et de l'Espace, Oct. 2006.
2. D. S. Taubman and M. W. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Boston, MA: Kluwer Academic Publishers, 2002.
3. *Information technology – JPEG 2000 image coding system: Core coding system*, ISO/IEC Std. 15 444-1, 2002.
4. *Information technology – JPEG 2000 image coding system: Extensions*, ISO/IEC Std. 15 444-2, 2004.
5. E. Christophe, D. Léger, and C. Mailhes, "Quality criteria benchmark for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 09, pp. 2103–2114, Sep. 2005.
6. C. Thiebaut, E. Christophe, D. Lebedeff, and C. Latry, "CNES studies of on-board compression for multi-spectral and hyperspectral images," in *SPIE, Satellite Data Compression, Communications, and Archiving III*, vol. 6683. SPIE, Aug. 2007.
7. D. Van Buren, "A high-rate JPEG2000 compression system for space," in *IEEE Aerospace Conference*. IEEE, Mar. 2005, pp. 1–7.
8. P.-S. Yeh, P. Armbruster, A. Kiely, B. Masschelein, G. Moury, C. Schaefer, and C. Thiebaut, "The new CCSDS image compression recommendation," in *IEEE Aerospace Conference*. IEEE, Mar. 2005.
9. X. Tang, W. A. Pearlman, and J. W. Modestino, "Hyperspectral image compression using three-dimensional wavelet coding," in *Image and Video Communications and Processing*, vol. 5022. SPIE, Jan. 2003, pp. 1037–1047.
10. H. Kim and C. Choe, Jand Lee, "Fast implementation of 3-D SPIHT using tree information matrix," in *IEEE International Geoscience and Remote Sensing Symposium, IGARSS'03*, vol. 6, Jul. 2003, pp. 3586–3588.
11. 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.
12. S. Cho and W. A. Pearlman, "Error resilient video coding with improved 3-D SPIHT and error concealment," in *Image and Video Communications and Processing*, vol. 5022. SPIE, Jan. 2003, pp. 125–136.
13. R. R. Coifman, Y. Meyer, S. Quake, and M. V. Wickerhauser, "Signal processing and compression with wavelet packets," Numerical Algorithms Research Group, Yale University, New Haven, Tech. Rep., 1990.

14. D. Xu and M. N. Do, "Anisotropic 2D wavelet packets and rectangular tiling: theory and algorithms," in *SPIE, Wavelets: Applications in Signal and Image Processing X.*, vol. 5207, Nov 2003, pp. 619–630.
15. R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 713–718, Mar. 1992.
16. 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, Apr. 1993.
17. Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 9, pp. 1445–1453, Sep. 1988.
18. E. Christophe, C. Mailhes, and P. Duhamel, "Best anisotropic 3-D wavelet decomposition in a rate-distortion sense," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'06*, vol. 2, May 2006, pp. II–17 – II–20.
19. X. Tang, S. Cho, and W. A. Pearlman, "3D set partitioning coding methods in hyperspectral image compression," in *International Conference on Image Processing*, vol. 2, 2003, pp. 239–242.
20. 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, Jun. 1996.
21. E. Christophe and W. A. Pearlman, "Three-dimensional SPIHT coding of volume images with random access and resolution scalability," *EURASIP Journal on Image and Video Processing*, 2008.
22. C. D. Creusere, "A new method of robust image compression based on the embedded zerotree wavelet algorithm," *IEEE Transactions on Image Processing*, vol. 6, no. 10, pp. 1436–1442, Oct. 1997.
23. S. Cho and W. A. Pearlman, "Multilayered protection of embedded video bitstreams over binary symmetric and packet erasure channels," *Journal of Visual Communication and Image Representation*, vol. 16, no. 3, pp. 359–378, Jun. 2005.
24. "The ORFEO toolbox software guide," http://otb.cnes.fr, 2008.
25. E. Christophe, P. Duhamel, and C. Mailhes, "Signed binary digit representation to simplify 3D-EZW," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'07*, Apr. 2007.
26. ——, "Adaptation of zerotrees using signed binary digit representations for 3 dimensional image coding," *EURASIP Journal on Image and Video Processing*, 2007.