

## **Aviris Hyperspectral Images Compression Using 3d Spiht Algorithm**

Ouahioune mohand<sup>1</sup>, Akroul leila<sup>2</sup>, Lahdir mourad<sup>3</sup>, Ameer soltane<sup>4</sup>  
<sup>1, 2, 3, 4</sup>(Laboratoire d'Analyse et Modélisation des Phénomènes Aléatoires (LAMP A)),  
Department of Electronics, Mouloud Mammeri University (UMMTO),  
Po. Box 17 RP 15000, Tizi-Ouzou, Algeria)

---

**Abstract**—Hyperspectral Sequence image is compressed using a 3D discrete wavelet transform (3D DWT) to decorrelate the three dimensions of hyperspectral image. Then the sequence is encoded efficiently by the 3D SPIHT algorithm, which operates on the pyramid structure of wavelet coefficients obtained. After decompression, the image evaluated by the PSNR, are faithfully reproduced and the memory space required for storage and transmission has been reduced. The Building of GUI in MATLAB has facilitated the use of algorithms developed and the tests of the compression for different parameters.

**Keywords** — 3D DWT, 3D SPIHT, AVIRIS, Image compression, Hyperspectral.

---

### **I. INTRODUCTION**

Sensors which they are used to observe the Earth from space or to explore deep space, have always sought to acquire data of better quality, to improve the scientific information provided. The improvement of the performances of these sensors requires the improvement of space resolution, radiometric precision and the number of spectral bands acquired. Instruments with high spectral resolution fall within this global evolution. These sensors are named hyperspectral sensors [1].

Hyperspectral sensors are capable of recording information in multitude spectral bands, much narrower and contiguous, in the visible and infrared electromagnetic spectrum portions. Hyperspectral data provide more detailed information of the spectral properties of a scene (fine spectral signature). Typically, each pixel in the image is represented by hundred values, corresponding to different wavelengths. These values correspond to a sampling of the continuous spectrum emitted by the pixel. This sampling of the spectrum with very high resolution allows identification and a more accurate discrimination of the objects than multispectral data [1]. The availability of the spectral information of each pixel leads to new applications in all areas using remote sensing; we can name: geology (identification of the mineral), the forestry (health state, identification of kinds) or the management of aquatic environment (quality of waters...).

Because of the huge amount of information and their specific properties, compression step becomes a crucial part for transmission and storage. Based on the techniques available in the literature, various algorithms and standards have been developed to deal with this type of data [2]. For instance, wavelet transform has been efficiently used for 2D image coding [3] [4]. Besides, it is considered as the kernel of the standard JPEG2000 [5]. Extended to volumetric images, JPEG2000 standard has been widely applied to 3D hyperspectral images encoding [6]. Afterwards, the 3D wavelet transformation has been efficiently employed for various types of data through [2] [7] [8] [9]. Recently, a 3D wavelet decomposition which includes an adaptation of the zero tree structure [1] highlighted the potential of using such scheme to compress 3D hyperspectral images.

This document proposes to adapt the SPIHT to a 3D wavelet coding and to compare the performance of 2D compression vs. 3D compression in terms of PSNR.

### **II. COMPRESSION METHOD**

The idea of our scheme of compression adapted to hyperspectral images is represented in Figure1.

The 1st step of this compression scheme is the wavelet transform. Its ability to compact the energy on a small number of coefficients provides an efficient encoding of the image. The three-dimensional nature (multitude of spectral bands) and the high spectral correlation (narrow bands and contiguous) of hyperspectral imaging has led us to expand this wavelet transform to three dimension (3D DWT). This transformation is based on the multiresolution decomposition of Mallat [10] [11]; this decomposition is based on the high pass and low pass filters deduced from the wavelet and the associated scale function.

The 2nd step of this compression scheme is the application of the 3D SPIHT encoder, which exploits the pyramidal structure of wavelet coefficients where there are significant correlations between the different sub-bands. The 3D SPIHT uses three-dimensional trees, thus, isolated large areas of insignificant coefficients; it is

what achieving good performance in the compression. The implementation of an entropic encoder does not improve the results, so we decided to not implement it in our chain to not increase the complexity of the developed algorithm.

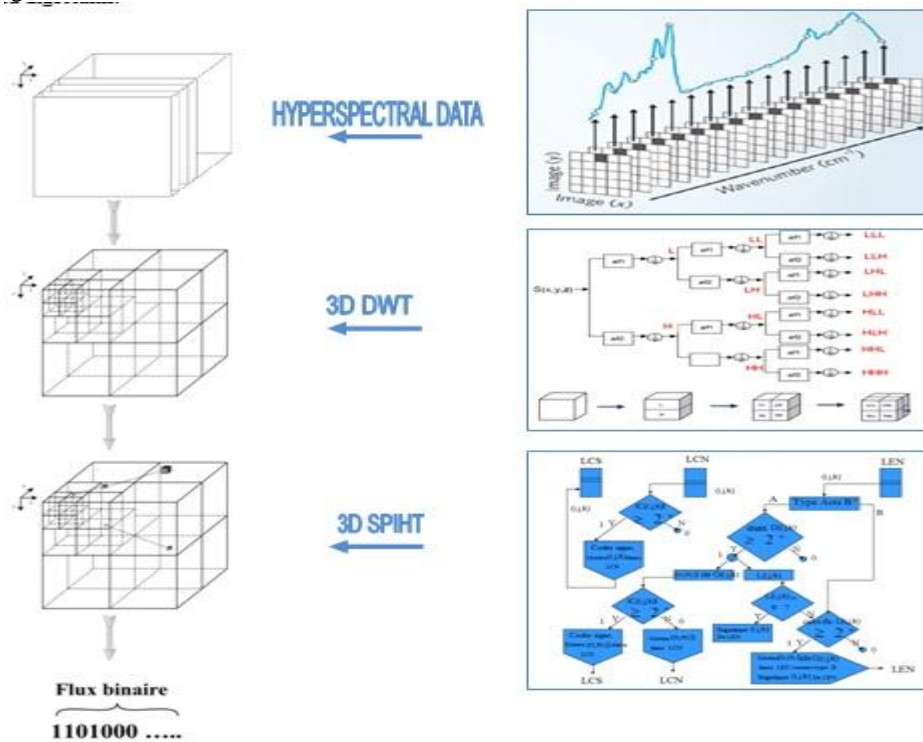


Figure 1. 3D Scheme of Compression Adapted to Hyperspectral Images.

### III. ADAPTING SPIHT TO 3D WAVELET TRANSFORM

#### III.1 Method:

The main idea of zero trees encoding, is to use the position of the significant coefficients at the lowest resolution to predict the position and value of the significant coefficients at the highest resolution. A tree structure is then used to represent the wavelets coefficients.

SPIHT (Set Partitioning in Hierarchical Tree) [1] [12] [13] is the most popular variant of this type of encoding. It was accomplished in 1996 by A. Said and W. Pearlman. We will do in the following the extension of the SPIHT to three-dimensional, to be used with the 3D DWT.

The extension to the 3D images of the SPIHT relies on three-dimensional trees [1] [14]. In the 3D case, the root node of the tree (corresponding to the coefficients of the sub-band LLL<sub>j</sub> for 3D DWT with j levels of decomposition) has only seven children, while all other nodes except for ends have eight, in other words, except for the root node and the ends of the tree, the link parent/children is the following:

$$O(i, j, k) = \left\{ \begin{array}{l} (2i, 2j, 2k), (2i, 2j+1, 2k), (2i+1, 2j, 2k), \\ (2i+1, 2j+1, 2k), (2i, 2j, 2k+1), (2i+1, 2j, 2k+1), \\ (2i, 2j+1, 2k+1), (2i+1, 2j+1, 2k+1) \end{array} \right\} \quad (1)$$

Where  $O(i, j, k)$  represents the set of coordinates of all children of the node  $(i, j, k)$ .

SPIHT uses two types of zero trees. The first (type A) consists of a simple root with all its descendants to zero for a given plan of bits. Although the zero tree is specified by the coordinates of the root, but the root is not included in the tree. The second type of zero trees (type B) is similar but excludes the eight children of the root. Type B contains only grandchildren, great grandchildren...ext... of the root.

To characterize these parent/children relationships in the sub-bands. The following sets are used:

$O(i, j, k)$ : All coordinates of all children of node  $(i, j, k)$ .

$D(i, j, k)$ : All coordinates of all descendants of node  $(i, j, k)$  (type A of zero tree).

$L(i, j, k) = D(i, j, k) - O(i, j, k)$ : All coordinates of all descendants except the children (type B of zero trees).

SPIHT stores the significant information in three ordered lists. These are lists of coordinates  $(i, j, k)$  in LIP (List of Insignificant Pixels) and LSP (List of Significant Pixels) and they represent coefficients. In the LIS (List of Insignificant Sets), they represent the sets  $D(i, j, k)$  (type A) or the sets  $L(i, j, k)$  (type B).

The LIP contains the coordinates of coefficients that were insignificant in the previous pass. In the current pass, they are tested, and those that are significant are moved to the LSP. Similarly, the sets of LIS are evaluated in order of entry, and when a set is found significant, it is removed from the list and then partitioned. The new sets with more than one item are added to the end of LIS with the type (A or B), while the simple coefficients are added to the end of LSP or LIP according to their significance. The list LSP contains the coordinates of the coefficients to be visited in the next pass of refinement. The refinement pass transmits the  $n^{th}$  most significant bit of entries of LSP.

**III.2 Algorithm:**

Step 1: Calculation of the starting threshold  $T_n$ , given by:

$$T_n = 2^{\lfloor \log_2 \max(im(x,y,\lambda)) \rfloor} \quad (2)$$

Step 2: Define the lists LIP, LSP and LIS of coordinates (i, j, k).

Step 3: Compare each pixel of LIP to threshold  $T_n$ :

- If positive, then it will be coded by '11'.
- If negative, then it will be coded by '10'.
- Otherwise it will be code '0'.

Step 4: For each set of LIS, compare the descendant with the largest absolute value to threshold  $T_n$ .

If type A:

- If there are descendants greater than  $T_n$ , output '1' and test the eight children of the node (i, j, k).  
For each of them:  
If positive, then it will be coded by '11'.  
If negative, then it will be coded by '10'.  
Otherwise it will be code '0'.  
If children have descendants, put the node (i, j, k) at the end of LIS as type B.

- If there are no descendants greater than  $T_n$ , output '0'.

If type B:

- If there are descendants greater than  $T_n$ , output '1' and put each one of the eight children of the node (i, j, k) as type A
- If not, output '0'.

Step 5: Refinement pass.

Step 6: Decrement n and return to step 3.

**IV. RESULTS AND DISCUSSION**

When using lossy compression techniques, the use of quality measurement is essential for performance evaluation. To do this, we will use the PSNR to allow the evaluation of the compression quality of our compression algorithm.

$$PSNR_{dB} = 10 \log_{10} \frac{(2^{16}-1)^2}{MSE} \quad (3)$$

Where: Peak Signal =  $2^q$ , q is the number of bits used to encode values. (q=16 for hyperspectral images). This equation of PSNR treats the three dimensions of the image.

A way to give an order of magnitude of the entropy (or at least an upper bound) of hyperspectral images is to test the lossless compression methods. We used the following tools on our AVIRIS image:

- gzip: LZ77 algorithm;
- bzip2: Burrows-Wheeler algorithm, and Huffman.

Original		gzip		bzip2	
size	bpp	size	bpp	size	bpp
134.4Mb	16	82.9Mb	9.86	69.6Mb	8.23
		CR =1.61		CR =1.92	

TABLE I. Entropy Estimation of Hyperspectral Image.

This allows us to understand that the entropy of hyperspectral images is less than 8.23 bits. For our tests, we used a sequence of 32 channels of the Yellowstone scene, acquired in 2006. It is an AVIRIS image that can be downloaded from: <http://aviris.jpl.nasa.gov/html>.

The behavior of different wavelet functions was studied. Three types of wavelet families are used: Daubechies (DbN), Symlettes (SymN) and biorthogonal (BioN). In each family of wavelets, we can find a wavelet function which gives the optimal solution associated with the order, but this depends on the image.

Table.II shows the performance of our algorithm for the decomposition level  $J = 3$  and for different families of wavelets.

Wavelet CR	Sym 6	Sym 8	Db 6	Db 8	CDF 9/7	LeGall 5/3
20	47.66	47.97	47.66	48.03	47.93	48.17
30	44.19	45.35	44.19	45.35	45.41	45.30
40	42.41	43.31	42.41	43.31	43.36	42.84
50	40.56	41.43	40.56	41.43	41.40	41.17
60	39.03	40.27	39.03	40.27	40.09	39.39
70	37.43	38.89	37.43	38.89	38.56	38.04
80	36.38	37.88	36.38	37.88	37.29	36.51
90	34.99	36.12	34.99	36.12	36.40	35.55
100	34.77	34.90	34.77	35.36	35.38	34.78

TABLE II. PSNR (dB) for Different Compression Ratio.

PSNR changes with the different families of wavelets. How each wavelet compact energy of the image depends very much on the image itself (spectral content of the image) and each wavelet differs in way that characterize this spectral content.

Fig.2 shows the performance of our algorithm for the decomposition level  $J = 3$  and for two biorthogonal wavelets (CDF9/7 and LeGall5/3). The PSNR varies with the different wavelets; as we see, the PSNR obtained exceed 35 dB for high compression ratio (CR=100), for low compression ratio, we get a good PSNR, exceeding 50 dB (CR=10).

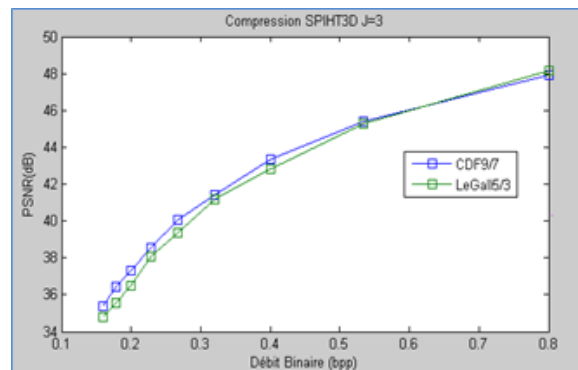


Figure 2. PSNR/CompRatio for CDF9/7 and LeGall5/3.

We can also see that the CDF9/7 is better than LeGall5/3 in high compression ratio but when compression ratio become smaller LeGall5/3 is better.

One advantage of SPIHT is a progressive coding by starting with the most significant bits to least significant. This allows the algorithm to do a progressive transmission of images, since the decoder can stop anywhere in the sequence of bits transmitted and produce the best reconstructed image. At the receiver, you can choose the quality, when we increase the flow, we obtain more detailed information and of course the image quality is better.

Figure 3 shows the compression quality obtained for various compression ratios. You can clearly observe an artifact effect in the high compression ratios, this is due to the limited amount of information contained in the image (high compression ratio), then when adding information, the quality of the image increases (low compression ratio).

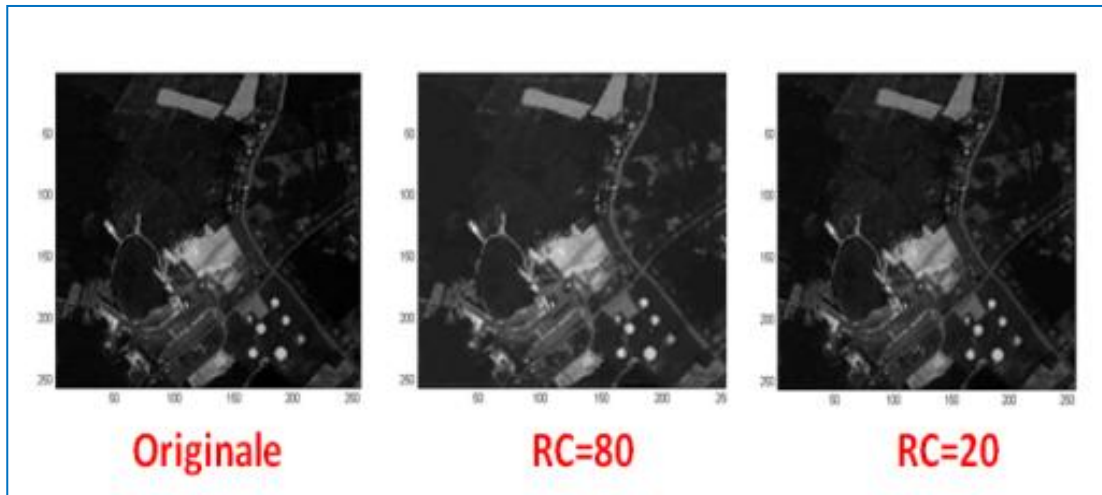


Figure 3. A) : Original Image ; B) : CR=80; C) : CR=20.

The algorithm can be implemented for any image size, but when the image size increases the time required for compression and reconstruction of the image also increases. Table III shows the effect of image size on quality of compression using the wavelet CDF9 / 7 for a CR = 100.

We can see that as the size of the sequence increases either that of the space domain (x, y) or / and of the spectral domain (z) the execution time of the algorithm increases. When the size of image increases, we get better results, especially when spectral domain increases, the results are very good because of the strong spectral correlation (the trees of zeros include more coefficients over a single tree, which allows better coding).

We can deduce from these results, that it is more interesting to design our algorithm to process image compression by small block, reducing the size of the space domain and favoring that of the spectral domain. This will fully benefit from the strong spectral correlation and reducing the cost of the computational algorithm (by reducing the size of the space domain, we reduce the execution time without sacrificing quality).

↓	→ <b>Spatial Domain</b>
	Size:128x128x32
	Size:256x256x32
	J : 3 PSNR : <b>35.4339</b>
	J : 3 PSNR : <b>35.3807</b>
	Size:128x128x64
	Size:256x256x64
	J : 3 PSNR : <b>32.5385</b>
	J : 3 PSNR : <b>32.8579</b>
	J : 4 PSNR : <b>36.9398</b>
	J : 4 PSNR : <b>36.0967</b>
	Size:128x128x128
	Size:256x256x128
	J : 3 PSNR : <b>34.9114</b>
	J : 3 PSNR : <b>33.1755</b>
	J : 4 PSNR : <b>39.5542</b>
	J : 4 PSNR : <b>39.3307</b>
	J : 5 PSNR : <b>39.3759</b>
↓	
<b>Spectral Domain</b>	

TABLE III. PSNR (dB) Obtained for Different Image Sizes.

To perform this comparison, we compressed each of the 32 channels of our image sequence separately, we calculated the PSNR for this sequence, i.e. once all channels are compressed separately (2D DWT and 2D SPIHT), we have grouped them to form the compressed 3D cube than we calculated his PSNR. We compared the results to those found with the 3D compression (3D DWT and 3D SPIHT). Figure 4 shows the PSNR for both types of compression. We can see that the 3D compression, offers more than **10dB** of improvement at all compression ratios. This implies that the hyperspectral images are highly correlated.

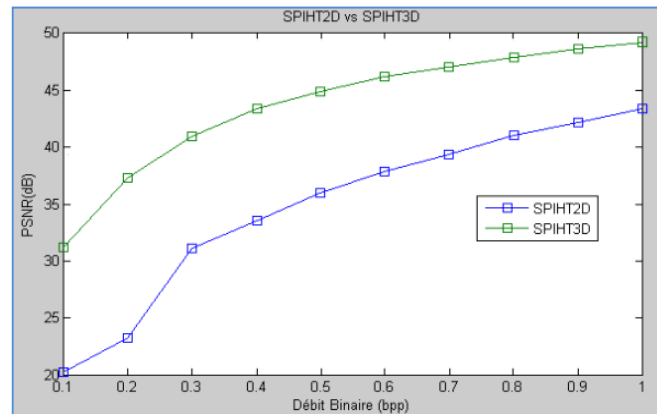


Figure 4. 2D Compression vs. 3D Compression.

### V. CONCLUSION

The algorithm that we have implemented can be used for any image size. When the size of the hyperspectral image increases, the time required for compression and reconstruction of the image also increases. It is more interesting to design the algorithm to process on images in small block, reducing the size of the space scene and focusing the spectrum scene. This will fully allow to benefit from the high spectrum correlation without sacrificing the execution time of the algorithm (by reducing the size of the space scene, we reduce the execution time without sacrificing quality).

Good performances are obtained by our algorithm because of the superiority of the SPIHT coding; the algorithm was tested using hyperspectral image sequence of 32 channels. The results show that the extension to the third dimension (spectral dimension) improves the compression quality, the 3D compression takes advantage of the high spectral correlation and provides a coding gain much higher than that offered by the 2D compression. Our results show a gain up to 10 dB for all compression ratios (by increasing the sequence, we increase the results).

### REFERENCES

- [1] E. Christophe, C. Mailhes and P. Duhamel, Hyperspectral Image Compression: Adapting SPIHT and EZW to Anisotropic 3D Wavelet Coding, *IEEE Transactions on Image Processing*, Vol. 17, No. 12, 2008, 2334-2346.
- [2] M. Lahdir, A. Nait-ali and S. Ameur, Fast Encoding-Decoding of 3D Hyperspectral Images Using a Non-Supervised Multimodal Compression Scheme, *Journal of Signal and Information Processing*, Vol. 2 No. 4, 2011, 316-321.
- [3] B. Penna, T. Tillo, E. Magli and G. Olmo, Transform Coding Techniques for Lossy Hyperspectral Data Compression, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 45, No. 5, 2007, 1408-1421.
- [4] G. Motta, F. Rizzo and J.A. Storer, Hyperspectral Data Compression, (*Springer 2006*).
- [5] D. S. Taubman and M. W. Marcellin, JPEG2000, Image Compression Fundamentals, Standards and Practice, (*Kluwer Academic Publishers, Boston, 2002*).
- [6] Q. Du and J. E. Fowler, Hyperspectral Image Compression Using JPEG2000 and Principal Component Analysis, *IEEE geoscience and Remote Sensing Letters*, vol. 4. 2007, 201-205.
- [7] J. E. Fowler and J. T. Rucker, 3D Wavelet-Based Compression of Hyperspectral Imagery, in C.I. Chang, Ed., *Hyperspectral Data Exploitation: Theory and Applications*, (John Wiley & Sons, Inc., 2007).
- [8] X. Tang, W. A. Pearlman and J. W. Modestino, Hyperspectral Image Compression Using Three-Dimensional Wavelet Coding: A Lossy-to-Lossless Solution, *Rensselaer Polytechnic Institute Troy, NY, 2004*, 22.
- [9] B. Penna, T. Tillo, E. Magli and G. Olmo, "Transform Coding Techniques for Lossy Hyperspectral Data Compression," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 45, No. 5, 2007, pp. 1408-1421.
- [10] S.G.Mallat, Theory for Multiresolution Signal Decomposition: The Wavelet Representation, *IEEE PAMI*, Vol.11. 1989, 674-693.
- [11] S.G Mallat, *Wavelet Tour of Signal Processing* (The Sparse Way. Academic Press).
- [12] J.Shapiro, Embedded image coding using Zerotree of Wavelet coefficients, *IEEE TSP*, vol. 41. 1993, 3445-3462.
- [13] A. Said and W. A. Pearlman, A new and efficient image codec based on set partitioning in hierarchical trees, *Proc. IEEE TCSVT*, Vol. 6. 1996, 243-250.
- [14] Y.Gaudeau, *Contributions en compression d'images médicales 3D et d'images naturelles 2D*, doctoral diss., Université Henri Poincaré, France. 2006.