

# A SCALABLE SPIHT-BASED MULTISPECTRAL IMAGE COMPRESSION TECHNIQUE

*Fouad Khelifi, Ahmed Bouridane, and Fatih Kurugollu*

School of Electronics, Electrical engineering and Computer Science  
Queen's University Belfast  
Northern Ireland, UK  
Email: {fkhelifi01,a.bouridane,f.kurugollu}@qub.ac.uk

## ABSTRACT

This paper addresses the compression of multispectral images which can be viewed, at the encoder side, as a three-dimensional (3D) data set characterized by a high correlation through the successive bands. Recently, the celebrated 3D-SPIHT (Sets Partitioning In Hierarchical Trees) algorithm has been widely adopted in the literature for the coding of multispectral images because of its proven state-of-the-art performance. In order to exploit the spectral redundancy in the 3D wavelet transform domain, a new scalable SPIHT based multispectral image compression technique is proposed. The rationale behind this approach is that image components in two consecutive transformed bands are significantly dependent in terms of zerotrees locations in the 3D-DWT domain. Therefore, by joining the trees with the same location into the List of Insignificant Sets (LIS), a considerable amount of bits can be reduced in the sorting pass in comparison with the separate encoding of the transformed bands. Numerical experiments on two sample multispectral images show a highly better performance of the proposed technique when compared to the conventional 3D-SPIHT.

## 1. INTRODUCTION

Multispectral images are normally encountered in a large number of applications, such as geology, meteorology, manufacturing, medicine, and agriculture. However, as the most sophisticated remote sensing systems are designed to provide higher spatial and spectral resolutions, higher radiometric precision, and larger ground coverage, storage and transmission of such a type of data become more challenging and may overwhelm the capacity of the available communication channels. Therefore, the use of efficient compression schemes is necessary.

Currently, with the huge advance in the use and distribution of digital multimedia data, the transmission and delivery of massively large images over heterogeneous data networks has turned out to be a real preoccupation for an increasing number of researchers in the field of image compression. Satisfying the requests in terms of image quality under the con-

straint of the diversity of the traffic on the network is a real issue. Since the server must satisfy the users having both high and low bandwidth simultaneously, a straightforward solution to the problem is to compress and store the encoded version of the image data at different bit-rates. However, this approach involves providing the server with a large disk space and a management system for the stored data. Therefore, scalability, the capability of decoding a compressed image at different bit-rates, provides an attractive solution, allowing only one encoded image version to be stored in the database thereby avoiding the overhead of maintaining several versions of the coded data at various bit rates. In rate scalable coding, all of the compressed data is embedded in a single bit stream which can be decoded at different bit-rates. The decompression algorithm receives and decodes the compressed data from the start of the bit stream until a desired data rate is achieved.

In practical applications on multi-component images, the simplest and direct extension is to encode the components of a volumetric image independently as a set of separate grayscale images. Besides, this strategy was adopted by the latest still image coding standard JPEG-2000 [1]. Indeed, the extension of JPEG2000 in Part II to 3D applications was confined to multi-component transformations [2]. Nevertheless, the separated coding of different bands would sacrifice the full embeddedness of the bit stream. Furthermore, an explicit rate allocation among the different planes is required, losing precise rate control in the bit-stream. In application on video data, Kim *et al.* [3] proposed an extension of SPIHT to 3D generating a fully embedded bit-stream. Further, it was modified and applied to compression of medical volumetric images [4]. An overview given in [5] showed that 3D-SPIHT yields the best performance with 9/7 biorthogonal wavelet filters in comparison with several related works. A variation of 3D-SPIHT was also adopted in [6] for multispectral image compression. The authors applied the KLT and the DWT in the spectral and spatial domains respectively.

More recently, it has been revealed that 3D-SPECK (Set Partitioning Embedded bloCK), used in the 3D-DWT domain, provides similar lossy and lossless compression performance to 3D-SPIHT and considerably outperforms the benchmark

JPEG-2000 multi components [7]. In this paper, our reference point will be drawn from the 3D-SPIHT coder which operates in the 3D-wavelet packet transform domain. Readers who wish to acquaint themselves with the 3D-tree structure related to the transform are referred to [3]. In order to exploit the spectral redundancy in the 3D-DWT domain, a new scalable SPIHT-based multispectral image compression technique is proposed. The rationale behind this approach is that, in the 3D-DWT domain, image components in two consecutive transformed bands are significantly dependent in terms of zerotrees locations. Therefore, by joining the trees with the same location into the List of Insignificant Sets (LIS), a considerable amount of bits can be reduced in the sorting pass in comparison with the separate encoding of the transformed bands. Numerical experiments on two sample multispectral images show a highly better performance of the proposed technique when compared to 3D-SPIHT.

## 2. JOINED SPECTRAL TREES

If one refers to a separate coding of the bands with SPIHT, at each value of the threshold, a portion of redundant information within LIS is sent to the decoder during the sorting pass since there is a similarity in terms of zero-trees locations between the consecutive bands. More specifically, if a set corresponding to a given band  $k$  is found insignificant with respect to the current threshold, a set with the same Spatial Orientation Tree (SOT) in band  $k + 1$  is more likely to be found insignificant. It can be shown that the probability to find  $\ell$  insignificant sets with the same SOT in  $\ell$  consecutive bands decreases as  $\ell$  increases [8]. Therefore, a judicious grouping of insignificant sets within spectral components would involve each two consecutive bands in the whole transformed image. While attempting to exploit the spectral redundancy, our attention is also focussed on the rate scalability.

In order to join two trees at the same spatial location in two consecutive bands  $k$  and  $k + 1$ , a virtual parent is used in the testing and partitioning processes related to LIS. As depicted in Fig. 1, the virtual parent-offspring relationship is determined in such a way that two children at a given spatial location  $(i, j)$  in bands  $k$  and  $k + 1$  share a virtual parent whose coordinates used in LIS are  $(i, j)$  as well. If one refers to the conventional SPIHT, the partitioning of a grouped set  $(i, j)$  produces four other grouped sets  $O(i, j)$  (see Fig. 2). Thus, from an implementation point of view, for a total number of bands<sup>1</sup>  $K$ , the number of initialized virtual parents in LIS is  $\lfloor \frac{K+1}{2} \rfloor$  times the number of parent coefficients in the LL subband of each band. Likewise, those which do not have children are initially added to LIP.

<sup>1</sup>In the case where  $K$  is odd, which is not considered here, the last band is separately processed.

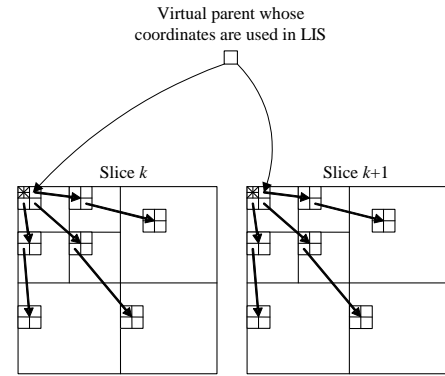


Fig. 1. Parent-children relationship in LL subband.

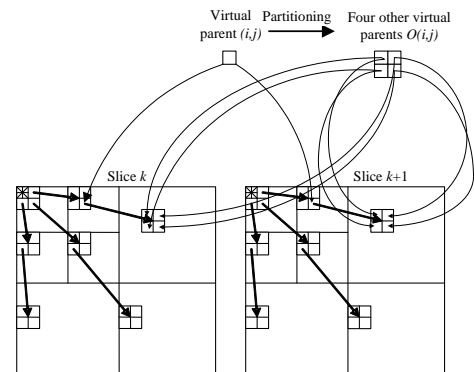


Fig. 2. Partitioning of a grouped set  $(i, j)$  into four other grouped sets.

## 3. PROPOSED JST-SPIHT ALGORITHM

Let us denote  $C_{k,k+1}$  be the maximum value in magnitude of the wavelet coefficients in two consecutive bands to be joined  $k$  and  $k + 1$ ; and

$$n_{k,k+1} = \lfloor \log_2(|C_{k,k+1}|) \rfloor \quad (1)$$

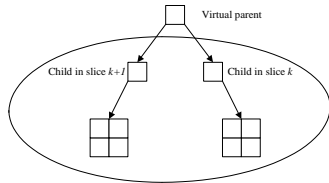
$$n = \max_{k \in \{0,2,4,\dots,K-1\}} \{n_{k,k+1}\} \quad (2)$$

Practically, in the first iterations, for most joined bands  $k$  and  $k + 1$  ( $k \in \{0, 2, 4, \dots, K - 1\}$ ),  $n > n_{k,k+1}$ . Thus, a comparison between  $n$  and  $n_{k,k+1}$  should be carried out at each sorting step as long as the above inequality is verified.

While the main contribution here lies in joining spectral trees in each group of two consecutive bands which involves a new tree-structure using the virtual parent-descendants relationship discussed above, the algorithmic part introduces an essential modification regarding the sets used in LIS. That is, four different types of sets are defined to take into account the worst cases where one of the direct descendants is not significant while the other is<sup>2</sup>. In such cases, once the significant

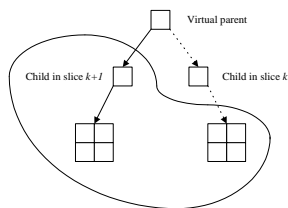
<sup>2</sup>Note that there are two direct descendants from different bands.

child is sorted, the remaining child is kept in LIS and tested within the corresponding set in the next iterations. The rationale behind this modification is motivated by the fact that the joined trees which do not exhibit high similarities with respect to the threshold provide a significance information related to the most significant tree. Sets of type A include all descendants of virtual parents within the two joined consecutive bands  $k$  and  $k + 1$  (see Fig. 3). As depicted in Fig. 4

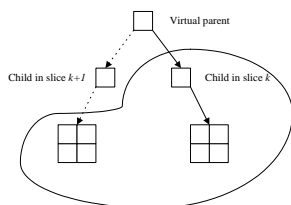


**Fig. 3.** Sets of type A.

and 5, sets of type B and C correspond to those which exclude a child in  $k$  and  $k + 1$  respectively. Finally, sets of type D are those excluding both the direct descendants in bands  $k$  and  $k + 1$  (Fig. 6). Let us define the following sets:

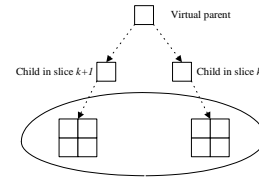


**Fig. 4.** Sets of type B.



**Fig. 5.** Sets of type C.

- $O(i, j)_k$  set of coordinates of all offspring of a parent coefficient  $(i, j)$  in band  $k$  as defined in the conventional SPIHT.
- $O_v(i, j)_{k, k+1} = \{(i, j)_k; (i, j)_{k+1}\}$  set of coordinates of offspring of a virtual parent  $(i, j)$  used to join two trees in consecutive bands  $k$  and  $k + 1$ .
- $D(i, j)_k$  set of coordinates of all descendants of a parent coefficient  $(i, j)$  in band  $k$  as defined in the conventional SPIHT.



**Fig. 6.** Sets of type D.

- $D_v(i, j)_{k, k+1}$  set of coordinates of all descendants of a virtual parent  $(i, j)$  used to join two trees in consecutive bands  $k$  and  $k + 1$ .
- $H(i, j)_k$  set of coordinates of all parent coefficients in LL subband in band  $k$ .
- $H_v(i, j)_{k, k+1}$  set of coordinates of all virtual parent coefficients in LL subband used to join initial trees in two consecutive bands  $k$  and  $k + 1$ .
- $L(i, j)_k = D(i, j)_k - O(i, j)_k$ .
- $L_v(i, j)_{k, k+1} = D_v(i, j)_{k, k+1} - O_v(i, j)_{k, k+1}$ .

In order to make clear the relationship between magnitude comparisons and message bits, the function  $S_n(\cdot)$  is used to identify significant coefficients or sets as given in the following equation

$$S_n(\Gamma) = \begin{cases} 1, & \text{if } \max_{(i, j) \in \Gamma} |C_{i, j}| > 2^n \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

To effectively describe the proposed Joined Spectral Trees SPIHT algorithm, which is referred here to as JST-SPIHT, let us consider that the  $K$  bands to be encoded are of size  $M \times N$ . Therefore, under the assumption that  $p$  levels of spatial wavelet decomposition are performed, LL subbands of each transformed band is of size  $(\frac{M}{2^p} \times \frac{N}{2^p})$ . Hence, the number of initial virtual parents in LIS is  $\frac{3}{4} \lfloor \frac{K+1}{2} \rfloor (\frac{M}{2^p} \times \frac{N}{2^p})$ . In the following,  $K$  is assumed even. The algorithm is detailed below.

## 1 Initialization

- for  $k' \in \{0, 1, \dots, \frac{K}{2} - 1\}$ , Output  $n_{2k', 2k'+1}$ . Set LSP as an empty list. Add all coordinates  $(i, j)$  in LL subband to LIP with band index  $2k'$ . Add all coordinates  $(i, j)$  in LL subband to LIP with band index  $2k' + 1$ . Add the coordinates  $(i, j) \in H_v|_{2k', 2k'+1}$  to LIS with band index  $k'$  as type A entries.

## 2 Sorting pass

- for each entry  $(i, j)$  in LIP, output  $S_n(i, j)$  in the corresponding band.
- if  $S_n(i, j) = 1$  then move  $(i, j)$  with the corresponding band index to LSP and output the sign of the coefficient  $C_{i, j}$ .

c) for each entry  $(i, j)$  in LIS do

- c.1) if  $(i, j)$  is a set of type A
  - Encode\_sets\_A  $(i, j)$
- c.2) if  $(i, j)$  is a set of type B
  - Encode\_sets\_B  $(i, j)$
- c.3) if  $(i, j)$  is a set of type C
  - Encode\_sets\_C  $(i, j)$
- c.4) if  $(i, j)$  is a set of type D
  - Encode\_sets\_D  $(i, j)$

### 3 Refinement pass

a) for each entry  $(i, j)$  in LSP, except those included in the last sorting pass, output the  $n^{th}$  most significant bit of  $|C_{i,j}|$  corresponding to the band index.

### 4 Quantization step

a) decrement  $n$  by 1 and go to step 2.

In the sorting pass, part c) calls four functions, Encode\_sets\_A  $(i, j)$ , Encode\_sets\_B  $(i, j)$ , Encode\_sets\_C  $(i, j)$ , and Encode\_sets\_D  $(i, j)$ , which are described as follows:

#### Encode\_sets\_A $(i, j)$

- output  $S_n(D_v(i, j)|_{2k', 2k'+1})$
- if  $S_n(D_v(i, j)|_{2k', 2k'+1}) = 1$  then
  - output  $S_n(i, j)|_{2k'}$
  - if  $S_n(i, j)|_{2k'} = 1$  then add  $(i, j)$  to LSP with band index  $2k'$  and output the sign of  $C_{i,j}$  in band  $2k'$ .
  - output  $S_n(i, j)|_{2k'+1}$
  - if  $S_n(i, j)|_{2k'+1} = 1$  then add  $(i, j)$  to LSP with band index  $2k'+1$  and output the sign of  $C_{i,j}$  in band  $2k'+1$ .
  - if  $S_n(i, j)|_{2k'} = 1$  and  $S_n(i, j)|_{2k'+1} = 1$  then move  $(i, j)$  to the end of LIS with band index  $k'$  as an entry of type D.
  - if  $S_n(i, j)|_{2k'} = 1$  and  $S_n(i, j)|_{2k'+1} = 0$  then move  $(i, j)$  to the end of LIS with band index  $k'$  as an entry of type B.
  - if  $S_n(i, j)|_{2k'} = 0$  and  $S_n(i, j)|_{2k'+1} = 1$  then move  $(i, j)$  to the end of LIS with band index  $k'$  as an entry of type C.
  - if  $S_n(i, j)|_{2k'} = 0$  and  $S_n(i, j)|_{2k'+1} = 0$  then add each  $(p, q) \in O(i, j)$  to the end of LIS with band index  $k'$  as an entry of type A and remove  $(i, j)$  from LIS. Add  $(i, j)$  to LIP with band index  $2k'$ . Add  $(i, j)$  to LIP with band index  $2k'+1$ .

#### Encode\_sets\_B $(i, j)$

- output  $S_n((i, j)|_{2k'+1}, L_v(i, j)|_{2k', 2k'+1})$
- if  $S_n((i, j)|_{2k'+1}, L_v(i, j)|_{2k', 2k'+1}) = 1$  then
  - output  $S_n(i, j)|_{2k'+1}$
  - if  $S_n(i, j)|_{2k'+1} = 1$  then add  $(i, j)$  to LSP with band index  $2k'+1$  and output the sign of  $C_{i,j}$  in band  $2k'+1$ . Move  $(i, j)$  to the end of LIS with band index  $k'$  as an entry of type D.

- if  $S_n(i, j)|_{2k'+1} = 0$  then add  $(i, j)$  to LIP with band index  $2k'+1$ . Add each  $(p, q) \in O(i, j)$  to the end of LIS with band index  $k'$  as an entry of type A and remove  $(i, j)$  from LIS.

#### Encode\_sets\_C $(i, j)$

- output  $S_n((i, j)|_{2k'}, L_v(i, j)|_{2k', 2k'+1})$
- if  $S_n((i, j)|_{2k'}, L_v(i, j)|_{2k', 2k'+1}) = 1$  then
  - output  $S_n(i, j)|_{2k'}$
  - if  $S_n(i, j)|_{2k'} = 1$  then add  $(i, j)$  to LSP with band index  $2k'$  and output the sign of  $C_{i,j}$  in band  $2k'$ . Move  $(i, j)$  to the end of LIS with band index  $k'$  as an entry of type D.
  - if  $S_n(i, j)|_{2k'} = 0$  then add  $(i, j)$  to LIP with band index  $2k'$ . Add each  $(p, q) \in O(i, j)$  to the end of LIS with band index  $k'$  as an entry of type A and remove  $(i, j)$  from LIS.

#### Encode\_sets\_D $(i, j)$

- output  $S_n(L_v(i, j)|_{2k', 2k'+1})$
- if  $S_n(L_v(i, j)|_{2k', 2k'+1}) = 1$  then
  - Add each  $(p, q) \in O(i, j)$  to the end of LIS with band index  $k'$  as an entry of type A and remove  $(i, j)$  from LIS.

As depicted by steps a) and b), to encode an entry in LIP during the sorting pass, the JST-SPIHT algorithm follows closely the methodology used in the conventional SPIHT [9]. The difference lies in step c) where four types of sets in LIS are considered. Indeed, since they are differently processed by the JST-SPIHT algorithm, an index is used to identify each set. A set is partitioned if it is of type D and one of its non direct descendants is found significant as given by function **Encode\_sets\_D**  $(i, j)$ . Otherwise, the remaining offsprings of the corresponding virtual parent are sorted jointly with its non direct descendants using the functions **Encode\_sets\_A**  $(i, j)$ , **Encode\_sets\_B**  $(i, j)$ , and **Encode\_sets\_C**  $(i, j)$ . This depends on the type of entry in LIS to be sorted as described by the steps c.1), c.2), and c.3). At the decoder side, the same steps are followed and, as the scalability involves, the reconstructed image is instantaneously updated subject to the received bits during the sorting and refinement passes.

## 4. IMPLEMENTATION AND NUMERICAL RESULTS

Our experiments were performed on two sample images 'Terrain' (307 pixels by 500 lines by 210 bands) and 'Urban' (307 pixels by 307 lines by 210 bands) quantized at non signed 16 bits [10]. For simplicity, we use 32 bands (from 60 to 91) of a square region of  $288 \times 288$  pixels. Also, the wavelet packet transform has been chosen. For the spectral dimension, three levels of decomposition were carried out by using first the 9/7 biorthogonal filters then Haar filters in the last decomposition<sup>3</sup>. In the spatial domain, four decomposition

<sup>3</sup>This is to ensure an accurate wavelet reconstruction since in the third decomposition level we only have 8-length signals to be decomposed.

levels were also conducted by using the 9/7 biorthogonal filters. The quality assessment of the decoded images is based on rate-distortion results measured by means of the overall Signal-to-Noise ratio ( $SNR$ ) given by

$$SNR = 10 \log_{10} \left( \frac{P}{MSE} \right) (dB) \quad (4)$$

where  $P$  and  $MSE$  denote the power of the original image and the mean squared error respectively. For the sake of comparison, we also report the results of 3D-SPECK [7] and SPIHT and JPEG-2000 [11], carried out separately on each band. This is referred to as separated SPIHT. Fig. 7 and 8 show that the proposed JST-SPIHT coder provides a significant improvement over 3D-SPIHT, which in turn considerably outperforms 3D-SPECK at all the bit-rates. In addition to the loss of the scalability property, the separated SPIHT and JPEG-2000 coders, which perform closely in terms of SNR, provide the poorest results for both images. The obtained results indicate that at all the bit-rates (between 0.1 and 0.5 bpp) the quality of the reconstructed images with JST-SPIHT is considerably higher than those obtained with 3D-SPIHT with a difference increasing up to 4.17 dB for 'Terrain' at 0.3 bpp and 5.05 dB for 'Urban' at 0.4 bpp (Fig. 9).

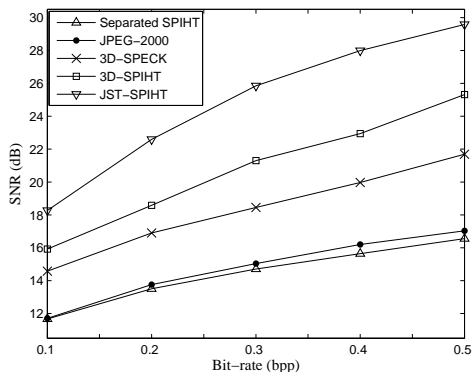


Fig. 7. Results for -Urban-.

## 5. CONCLUSION

In this paper, multispectral image compression based on the celebrated SPIHT algorithm has been addressed. In order to exploit the spectral redundancy within multi-band images, an efficient technique has been proposed. The key idea consists in joining each two consecutive bands according to a new tree-structure using a virtual parent-descendant relationship. Furthermore, to overcome the worst cases where one of the direct descendants is significant while the other is not, the proposed algorithm, JST-SPIHT, uses four types of insignificant sets during the sorting pass. Simulation results, conducted at different bit-rates and carried out on two sample multispectral

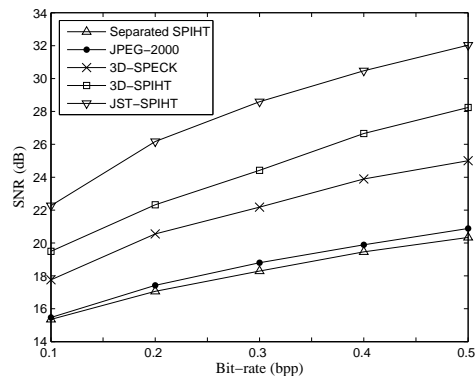


Fig. 8. Results for -Terrain-.

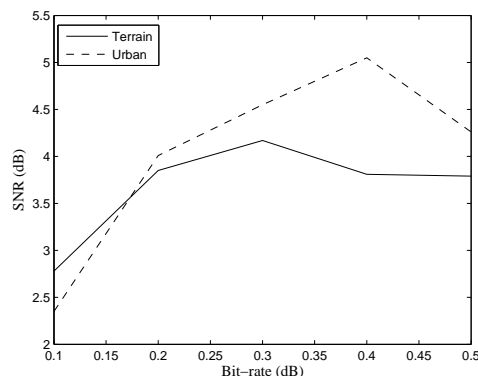


Fig. 9. Improvement over 3D-SPIHT.

images, have obviously shown that JST-SPIHT significantly outperforms the state-of-the art 3D-SPIHT.

## 6. REFERENCES

- [1] M.W. Marcellin, M.J. Gormish, A. Bilgin, and M.P. Boliek, "Overview of JPEG2000," *In Proc. IEEE Data Compression Conference*, pp. 523–541, June 2000.
- [2] A. Tzannes, "The new JPEG 2000 part 2 multi-component transfer syntaxes," in *Proc. International Conference On Digital Imaging and Communications in Medicine DICOM*, Hungary, Sep. 2005.
- [3] B-J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3D set partitioning in hierarchical trees (3D SPIHT)," *IEEE Trans. Circuits and Systems for video technology*, vol. 10, pp. 1374–1387, Dec. 2000.
- [4] G. Ginesu D. D. Giusto and W. A. Pearlman, "Lossy to lossless SPIHT-based volumetric image compression,"

in *Proc. IEEE Int. Conf. On Acoustic, Speech and Signal Processing*, Canada, Apr. 2004, pp. 693–696.

- [5] P. Schelkens, A. Munteanu, J. Barbarien, M. Galca, X. Giro-Nieto, and J. Cornelis, “Wavelet coding of volumetric medical datasets,” *IEEE Trans. On Medical Imaging*, vol. 22, pp. 441–458, Mar. 2003.
- [6] P. L. Dragotti, P. Giovanni, and A. R. P. Ragozini, “Compression of multispectral images by three-dimensional SPIHT algorithm,” *IEEE Trans. On Geoscience and Remote Sensing*, vol. 38, pp. 416–428, Jan. 2000.
- [7] X. Tang and W. A. Pearlman, *Three-Dimensional Wavelet-Based Compression of Hyperspectral Images*, chapter in *Hyperspectral Data Compression*, Kluwer Academic Publishers. Available on <http://www.cipr.rpi.edu/pearlman/>, 2005.
- [8] F. Khelifi, A. Bouridane, and F. Kurugollu, “Joined spectral trees for a scalable SPIHT-based multispectral image compression,” *To be submitted to IEEE Trans. On Multimedia*.
- [9] A. Said and W. A. Pearlman, “A new, fast and efficient image codec based on set-partitioning in hierarchical trees,” *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June. 1996.
- [10] ,” Available on: <http://www.tec.army.mil/Hypercube>.
- [11] D. Taubman, “Kakadu software, Ver. 5.0,” <http://www.kakadusoftware.com>.