LOSSY-TO-LOSSLESS BLOCK-BASED COMPRESSION OF HYPERSPECTRAL VOLUMETRIC DATA

Xiaoli Tang and William A. Pearlman[‡]

Center for Image Processing Research
Electrical, Computer, and Systems Engineering Department
Rensselaer Polytechnic Institute, Troy, NY 12180-3590

ABSTRACT

An embedded, block-based, wavelet transform coding algorithm of low complexity is proposed. Three-Dimensional Set Partitioned Embedded bloCK(3D-SPECK) efficiently encodes hyperspectral volumetric image data by exploiting the dependencies in all dimensions. Integer wavelet transform is applied to enable lossy and lossless decompression from the same bit stream. We demonstrate that 3D-SPECK, a wavelet domain algorithm, like other time domain algorithms, can preserve spectral profiles well. Airborne Visible Infrared Imaging Spectrometer (AVIRIS) imagery is used to test the proposed algorithm. Results show that 3D-SPECK, in addition to being very flexible, retains all the desirable features of compared state-of-the-art algorithms and is highly competitive to 3D-SPIHT and better than JPEG2000 multi-component in compression efficiency.

1. INTRODUCTION

Hyperspectral images produced by a new generation of sensors are finding many applications in detection and identification of the surface and atmospheric constituents present, analysis of soil type, environmental studies, and military surveillance. Hyperspectral images contain a wealth of data – they are generated by collecting hundreds of narrow and contiguous spectral bands of data such that a complete reflectance spectrum can be obtained for each point in the region being viewed by the instrument. As an example, the Airborne Visible InfraRed Imaging Spectrometer (AVIRIS) instrument, a typical hyperspectral imaging system, can yield about 16 Gigabytes of data per day. Efficient compression should be applied to these data sets before storage and transmission.

Many analysis hyperspectral image applications, such as common classification tools or feature extractions, can perform reliably on images compressed to very low bit rates. On the other hand, hyperspectral data carries rich information in the spectral domain. A ground sample point in a hyperspectral data set has a instinct spectral profile, which is the fingerprint information of the point. Some hyperspectral data users will rely on the spectrum of each point to create application products using their remote sensing algorithms. When compressing the hyperspectral images, we would like to preserve the important spectral profiles. Furthermore, given the extraordinary expense of acquiring hyperspectral imagery, it makes more sense to require lossless coding for archival applications. Therefore, in this study, we present integer wavelet filter solution allowing lossy-to-lossless compression for hyperspectral images.

Many Vector Quantization (VQ) based algorithms were proposed recently for hyperspectral image compression. Ryan and Arnold [5] proposed mean-normalized vector quantization (M-NVQ) for lossless AVIRIS compression. Each block of the image is converted into a vector with zero mean and unit standard variation. Motta [4] *et al.* proposed a VQ based algorithm that involved locally optimal design of a partitioned vector quantizer for the encoding of source vectors drawn from hyperspectral images. Pickering and Ryan jointly optimized spatial M-NVQ and spectral Discrete Cosine Transform (DCT) to produce compression ratios significantly better than those obtained by the optimized spatial M-NVQ technique alone. Other than VQ based methods, Harsanyi and Chang [1] applied Principle Component Analysis (PCA) on hyperspectral images to simultaneously reduce the data dimensionality, suppress undesired or interfering spectral signature, and classify the spectral signature of interest. A training sequence based entropy constrained predictive trellis coded quantization scheme was also proposed recently by Abousleman *et al.* for hyperspectral image compression. All these algorithms have promising performance on hyperspectral image compression. However, none of them generates embedded bit stream, and therefore cannot provide progressive lossy-to-lossless transmission.

To incorporate the embedded requirement and maintain other compression performances, many promising volumet-

[‡]Corresponding author. Voice: (518) 276-6082; fax: (518) 276-8715; e-mail: pearlw@ecse.rpi.edu

[†] This work was performed at Rensselaer Polytechnic Institute and was supported in part by National Science Foundation Grant No. EEC-981276. The government has certain rights in this material.

ric image compression algorithms based on wavelet transform were proposed recently. Several widely used ones are Three-Dimensional Context-Based Embedded Zerotrees of Wavelet coefficients (3D-CB-EZW) Three-Dimensional Set Partitioning In Hierarchical Trees (3D-SPIHT) [3], and Annex of Part II of JPEG2000 standard for multi-component imagery compression.

In this study, we propose a block-based volumetric image compression technique for hyperspectral images – Three-Dimensional Set Partitioned Embedded bloCK (3D-SPECK). It is an extended and modified version of SPECK [2]. Integer wavelet transform enables lossy-to-lossless reconstruction from the same embedded bit stream.

This paper is organized as following: We first present the 3D-SPECK algorithm in section II, followed by experimental results in section III. Section IV will conclude this study.

2. THE 3D-SPECK ALGORITHM

3D-SPECK is an extended and modified version of SPECK. Consider an image sequence which has been adequately transformed using the discrete wavelet transform in three dimensions. The transformed image sequence is said to exhibit a hierarchical pyramidal structure defined by the levels of decomposition, with the topmost level being the root. Pixels are grouped together in sets which comprise regions in the transformed images. As shown in Figure 1, each subband is treated as a code block, and the code block is called S set. The dimension of a set S depends on the dimension of the original images and the subband level of the pyramidal structure at which the set lies. Unlike SPECK, 3D-SPECK only has type S set. The algorithm only needs to check whether a set S is significant with respect to a certain bit plane.

To start the algorithm, all subbands are initialized as sets S and are put in the List of Insignificant Sets (LIS). In the first pass at the highest n, 3D-SPECK tests the significance of sets S in the LIS following the subband order of lowpass bands to highpass bands. As an example, for one-level decomposition, the scanning order is LLL, LHL, HLL, LLH, HHL, HLH, LHH, HHH. For higher level decomposition, the scanning path starts from the top of the pyramid down to its bottom by following the same order from lowpass bands to highpass bands.

If a set S is found significant, it will be split into eight approximately equal subsets. 3D-SPECK then treats each of these subsets as new type S sets, and in turn, tests their significance. This process will be executed recursively until reaching pixel level where the significant pixel in the original set S is located or reach the bit budget.

Integer wavelet filter is applied to enable lossy-to-lossless reconstruction from a same embedded bit stream. To achieve

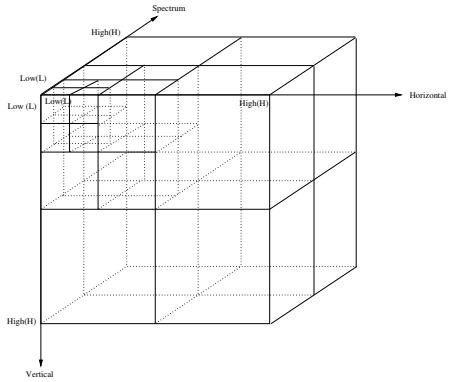

Fig. 1. Structure for 3D-SPECK

good lossy performance, it is important to have an unitary transform. If the transform is not unitary, the mean squared quantization error in the wavelet domain is, thus, not equal to the mean squared error (MSE) in the time domain. Therefore, the lossy coding performance will be compromised. Appropriate transform structure and scaling the integer wavelet coefficients can make the transform approximately unitary before quantization.

We adopt the transform structure mentioned in [7]. A 4-level 1D wavelet packet tree structure is applied on the spectral axis with appropriate scaling factors. As each scaling factor is some power of two, we can implement the scaling factor by bit shifting. For the spatial axes, we keep the same 2D dyadic wavelet transform to each slice.

The adaptive arithmetic coding algorithm is used to further improve the coding efficiency. As described, 3D-SPECK splits the significant set S into eight smaller subsets and tests the significance of the new subsets in turn. Instead of coding the significance test results of the eight subsets separately, 3D-SPECK codes them together first before further processing the subsets. Simple context is applied for conditional coding of the significance test result of this subset group.

3D-SPECK has low computational complexity. The algorithm is very simple, consisting mainly of comparisons, and does not require any complex computation. As 3D-SPECK checks sets from lowest subband to highest subband, it is natural multi-resolution analysis, while 3D-SPIHT does not have this advantage. Comparing to JPEG2000 multi-component, 3D-SPECK obviously has much lower complexity. 3D-SPECK also has low dynamic memory requirements. At any given time during the coding process, only one connected region is processed.

3. EXPERIMENT RESULTS

We performed coding experiments on three signed 16-bit reflectance AVIRIS image volumes. AVIRIS has 224 bands

Coding Methods	moffett scene 1	moffett scene 3	jasper scene 1
3D-SPECK	6.91	6.82	6.70
3D-SPIHT	6.94	6.74	6.72
JPEG2000 Mul	7.17	7.00	6.90
SPIHT	7.97	7.58	7.98
JPEG2000	7.79	7.73	8.59

Table 1. Comparison of methods for Lossless coding of test 16 bit image volumes. The data are given in bits per pixel per band (bpppb), averaged over the entire image volume. JPEG2000 Mul stands for JPEG2000 multi-component.

and 614 $\times$ 512 pixel resolution that corresponds to an area of approximately 11 km $\times$ 10 km on the ground. We have 1997 runs of Moffett Field scene 1 and 3 and Jasper Ridge scene 1. For our experiments, we cropped each scene to 512 $\times$ 512 $\times$ 224 pixels.

3.1. Comparison of Lossless Compression Performance

Table 1 presents the lossless performances of 3D-SPECK, 3D-SPIHT[1], JPEG2000 multi-component, 2D-SPIHT and JPEG2000. JPEG2000 multi-component is implemented by first applying S+P[2] filter on spectral dimension to decorrelate spectral correlation and followed by JPEG2000 on spatial dimensions. S+P integer filters are used for 3D-SPECK, 3D-SPIHT and 2D-SPIHT, while for JPEG2000, the integer filter (5,3) is used. For all 3D algorithms, including 3D-SPECK, 3D-SPIHT and JPEG2000 multi-component, the results of AVIRIS data are obtained by coding all 224 bands as a single unit, and for the two 2D algorithms, the results are obtained by first coding the AVIRIS data band by band and then averaging over the entire volume.

Overall, 3D algorithms perform better than 2D algorithms. Compared with 2D-SPIHT and JPEG2000, our proposed algorithm, 3D-SPECK yields, on average, 13.1% and 18.6% decreases in compressed file sizes for AVIRIS test image volumes. 3D-SPECK and 3D-SPIHT are fairly comparable as their results are quite close. They both out-perform the benchmark JPEG2000 multi-component, averaged over the three image volumes, by 3.0% and 3.2% decreases in file size, respectively. Surprisingly, considering its considerably higher complexity, JPEG2000 is not as efficient as 2D-SPIHT for hyperspectral images. As shown in the table, 2D-SPIHT always yields smaller bits per pixel per band (bpppb) than that of JPEG2000.

[1] We use symmetric tree 3D-SPIHT here.

[2] All S+P filters used in this study are B filters.

Coding Methods	SNR (dB) at bit rates (bpppb)			
	0.2	0.5	1.0	4.0
moffett scene 1				
3D-SPECK	20.78	29.20	37.28	54.07
3D-SPIHT	20.61	29.11	37.20	53.96
JPEG2000 Mul	19.66	27.99	36.31	53.69
moffett scene 3				
3D-SPECK	16.56	25.99	34.85	49.46
3D-SPIHT	16.74	26.10	34.95	49.55
JPEG2000 Mul	15.95	25.21	33.84	49.24
jasper scene 1				
3D-SPECK	22.68	30.40	36.70	51.76
3D-SPIHT	22.55	30.28	36.65	51.70
JPEG2000 Mul	21.87	29.04	36.04	51.12

Table 2. Comparative evaluation the rate distortions of 3D-SPECK, 3D-SPIHT and JPEG2000 multi-component.

3.2. Comparison of Lossy Compression Performance

We report lossy coding performances using rate-distortion results, by means of Signal-to-Noise Ratio (SNR) for the whole sequence: $\mathrm{SNR} = 10 \log_{10} \frac{\sigma_x^2}{\mathrm{MSE}}$ dB . Where σ_x^2 is the average squared value of the original AVIRIS sequence, and MSE is the mean squared error over the entire sequence.

The rate-distortion results for 3D-SPECK, 3D-SPIHT and JPEG2000 multi-component are plotted in Table 2 for our three test image volumes. Overall, both 3D-SPECK and 3D-SPIHT perform better than JPEG2000 multi-component, providing higher SNR all the time. For all three test image volumes, the results show that 3D-SPECK is comparable to 3D-SPIHT, being slightly worse for moffett scene 3, but slightly better for moffett scene 1 and jasper scene 1.

As the most import information for hyperspectral users is the spectral profile, we also illustrate the performance of 3D-SPECK by plotting the original spectral profiles of individual pixels, along with associated reconstructed and error profiles. Figure 2 shows the profiles for one asphalt pixel of Jasper scene 1. The spectral profiles are preserved excellently even at 1.0 bppp, with only several larger values of errors occur at the spectral valleys around bands 160 and 224. The largest error correspond to 2.4% of the maximum value. Increasing the bit rate, the error (difference) values drop quickly. The absolute values of errors are already within 25 at 2.0 bppp, corresponding to 0.7% of the maximum values. For bit rate at 4.0 bppp, as shown in figure 2, the differences between the original pixels and the reconstructed ones are barely distinguishable, and the errors are very small.

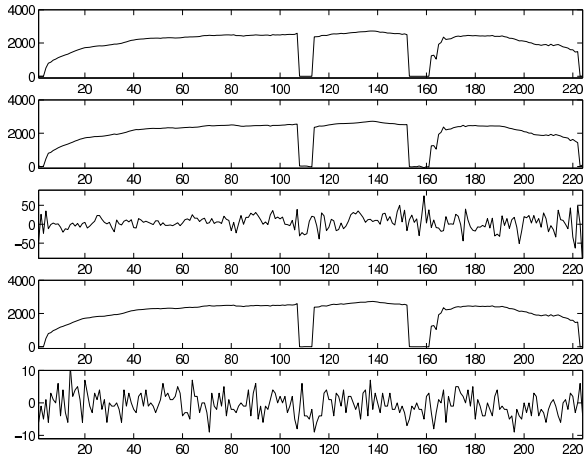

Fig. 2. The original, reconstructed and the difference values between the original and reconstructed pixels for an asphalt pixel for Jasper scene 1. The first graph is the original, the second one is the reconstructed pixel at 1.0 bppp, the third one is the difference values at 1.0 bppp, the fourth one is the reconstructed pixel at 4.0 bppp, and the last one is the difference values at 4.0 bppp.

3.3. Classification Performance

To address how our compression algorithm impacts remote sensing applications, we provide an experiment for a well-known remote sensing classification method, Spectral Angle Mapper (SAM), to test 3D-SPECK.

Table 3 lists the classification results for two classes (asphalt and vegetation) of Jasper scene 1. We can see that the classification tasks investigated are robust with respect to lossy compression of the source image. The percentage of correctly classified pixels converges to 100% at the rates higher than 1 bppp for all three algorithms, with JPEG2000 multi-component being slightly worse than that of 3D-SPECK and 3D-SPIHT. The distortions in the reconstructed data caused by the compression process result in only minor losses in classification accuracy even at low bit rate such as 1 bppp, with the classification accuracy higher than 99% almost all the time. For 3D-SPECK and 3D-SPIHT at very low bit rate such as 0.2 bppp, the percentages of classification accuracy are already higher than 97%. JPEG2000 multi-component provides much worse classification performances at 0.2 bppp. Overall, JPEG2000 multi-component performs not as well as the other two algorithms, rendering much poorer classification accuracy at very low bit rates and slightly worse performance at higher bit rate.

Coding Methods	**CA (%) at bit rates (bpppb)**			
	0.2	**1.0**	**2.0**	**4.0**
Asphalt				
3D-SPECK	97.91	99.87	99.97	99.98
3D-SPIHT	97.56	99.42	99.97	99.98
JPEG2000 Mul	75.57	99.31	99.88	99.96
Vegetation				
3D-SPECK	97.20	99.64	99.82	99.99
3D-SPIHT	97.83	99.84	99.90	99.99
JPEG2000 Mul	84.40	98.99	99.58	99.93

Table 3. Jasper scene 1 SAM classification. CA stands for Classification Accuracy.

4. CONCLUSION

The proposed 3D-SPECK provides excellent performance on hyperspectral image compression, while preserving important information of spectral profiles. It is a good candidate for many hyperspectral image applications.

5. REFERENCES

[1] J.C. Harsanyi, and C.I. Chang, *Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach*, IEEE Trans. Geoscience and Remote Sensing. Vol. 32, No. 4, July 1994.

[2] A. Islam and W.A. Pearlman, *An embedded and efficient low-complexity hierarchical image coder*, in Proc. SPIE Visual Comm. and Image Processing, vol. 3653, pp.294-305, 1999.

[3] B. Kim and W.A. Pearlman, *An embedded wavelet video coder using three-dimensional set partitioning in hierarchical tree*, IEEE Data Compression Conference, pp.251-260, March 1997.

[4] G. Motta, F. Rizzo, and J.A. Storer, *Compression of hyperspectral imagery*, Data Compression Conference. Proceedings. DCC 2003, pp. 25-27, March 2003.

[5] M.J. Ryan and J.F. Arnold, *The lossless compression of AVIRIS images by vector quantization*, IEEE Trans. Geoscience and Remote Sensing, Vol. 35, No. 3, May 1997.

[6] 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 6, pp.243-250, June 1996.

[7] Z Xiong, X. Wu, S. Cheng, and J. Hua, *Lossy-to-lossless compression of medical volumetric data using three-dimensional integer wavelet transforms*, IEEE Trans. on Medical Imaging, Vol. 22, No. 3, March 2003.