SCALABLE HYPERSPECTRAL IMAGE CODING

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

Here we propose scalable Three-Dimensional Set Partitioned Embedded bloCK (3D-SPECK)–an embedded, block-based, wavelet transform coding algorithm of low complexity for hyperspectral image compression. Scalable 3D-SPECK supports both SNR and resolution progressive coding. After wavelet transform, 3D-SPECK treats each subband as a coding block. To generate SNR scalable bitstream, the stream is organized so that the same indexed bit planes are put together across coding blocks and subbands, so that the higher bit planes precede the lower ones. To generate resolution scalable bitstreams, each subband is encoded separately to generate a sub-bitstream. Rate is allocated amongst the sub-bitstreams produced for each block. To decode the image sequence to a particular level at a given rate, we need to encode each subband at a higher rate so that the algorithm can truncate the sub-bitstream to the assigned rate. Resolution scalable 3D-SPECK is efficient for the application of an image server. Results show that scalable 3D-SPECK provides excellent performance on hyperspectral image compression.

1. INTRODUCTION

Hyperspectral imagery, with its ability to detect, monitor and recognize different surface and atmospheric constituents, could effectively be applied to many applications such as positive identification of unknown objects and general 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 some applications become truly practical.

[‡]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.

SNR scalability is a very attractive feature of wavelet based embedded algorithms. With successive approximate quantization implicitly executed by the algorithm, a single embedded code stream can generate images at a variety of precision levels without repeating coding.

Resolution scalability is also a useful functionality in a case that the display devices have a wide range of resolution levels. As an example, to display a high resolution image sequence on a monitor which can only show relatively low resolution images, one only needs to decode portion of the bit stream that eventually contribute to coding of the related low frequency components. Resolution scalability is important also in the sense that it is connected to complexity scalability as the consumptions of memory and computational resource is commonly exponentially increased or reduced from one resolution level to another. Therefore, this feature can be used for the application of an image server.

Hyperspectral imagery has an important property that it has numerous high frequency content. Therefore, hyperspectral image compression algorithm should also have excellent performance on images with numerous high frequency content.

Vector Quantization (VQ) based algorithms were proposed for hyperspectral image compression. Ryan and Arnold [6] 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 [5] *et al.* proposed a VQ based algorithm that involved locally optimal design of partitioned vector quantizer for the encoding of source vectors drawn from hyperspectal image. 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. All these algorithm have promising performance on hyperspectral image compression. however, none of them generates embedded bitstream.

To incorporate the embedded coding requirement and maintain other compression performances, many promising volumetric 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) [4], 3D-SPECK [7], and Annex of Part II of JPEG2000 standard for multi-component imagery compression.

Among these wavelet based embedded image compression algorithms, 3D-SPECK has very good performance on hyperspectral image compression; it performs excellently on image sequences with numerous high frequency content. It's simple, efficient, and with low computational complexity.

Most embedded algorithms in the literature are unable to efficiently provide resolution scalable codestreams due to the entanglement in coding, modeling, and data structure across different resolution. In particular, the classical zerotree coders with individual zerotrees spanning several subband scales are not efficient for resolution scalable coding. Scalable 3D-SPECK, however, is designed to have the block based structure which is easy to support resolution scalable coding. The original 3D-SPECK supports SNR progressive coding. It is easy to implement 3D-SPECK to support resolution progressive coding as well.

This paper is organized as following: We first present SNR and resolution scalable 3D-SPECK in section 2, followed by experimental results in section 3. Section 4 will conclude this study.

2. SCALABLE 3D-SPECK

3D-SPECK is an extended and modified version of SPECK [3]. 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. Figure 1 illustrates such a structure with three-level decomposition. The finest pixels lie at the bottom level of the pyramid while the coarsest pixels lie at the top level.

Pixels are grouped together in sets which comprise regions in the transformed images. Each subband is treated as a code block, and a 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.

The original 3D-SPECK [7] supports only SNR scalable coding. It tests the significance for all sets S against the highest bit plane first, followed by lower bit planes. 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.

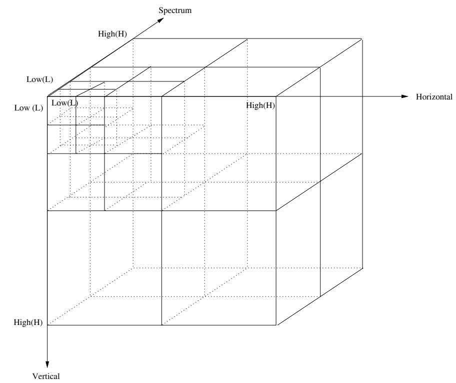

Fig. 1. Structure for scalable 3D-SPECK

Although the subband transform structure is inherently scalable of resolution, most embedded coders in the literature are unable to efficiently provide resolution scalable code streams. 3D-SPECK, however, is efficient for multi-resolution coding in a subband coding system. There are two reasons. First, the adaptive arithmetic coding models are accumulated from samples within the same resolution scale; Second, the modeling contexts do not include any neighbor from the finer scales. These two conditions guarantee the decodability of the truncated code stream.

The idea of implementing resolution scalable 3D-SPECK is just to run the 3D-SPECK algorithm on each subband separately. Instead of maintaining the same significance threshold across subbands until we exhaust them, we maintain separate LIS and LSP lists for each subband and proceed through the lower thresholds in every subband before moving to the next one.

To obtain resolution progressive bit stream, the bits in blocks belonging to subbands of coarser scales are encoded before those of finer scales. To start the resolution scalable 3D-SPECK, each subband is initialized as a set S, and is put in a List of Insignificant Sets (LIS). For total of K subbands, there are K LIS at the initialization. With the same cube splitting algorithm, resolution scalable 3D-SPECK tests the significance of each set S separately from the highest bit plane to the lowest bit plane to generate an embedded sub-bitstream. The overall bitstream is organized that the sub-bitstreams for coarser scales precede those of finer scales.

For the SNR progressive coding mode, the bits are allocated optimally across subbands, according to the significance threshold of the coefficients. But, for the resolution progressive mode, rate is allocated amongst the sub-bitstreams produced for each block. For a given target bit rate, we need now to apply an explicit bit allocation algorithm to assign different bit rates to different subbands to minimize mean squared error. We adopt the algorithm proposed in [2] to do rate allocation. To decode the image sequence to a particular level at a given rate, we need to encode each subband at a higher rate so that the algorithm can truncate the sub-bitstream to the assigned rate. The overall bitstream can serve lossy-to-lossless hyperspectral image compression.

3. EXPERIMENT RESULTS

We implemented both floating point filter and integer filter for scalable 3D-SPECK including both SNR and resolution scalable 3D-SPECK. A wavelet packet structure with appropriate scaling factors is used for the integer filter implementation to make the transform approximate unitary. SNR scalable 3D-SPECK results could be found in [7, 8] for several sets of AVIRIS image sequences comparing with other state-of-the-art algorithms. In this study, we present resolution scalable results. For comparison, we also provide results of the original SNR scalable 3D-SPECK at the partial and full scales.

We performed coding experiments on a signed 16-bit reflectance AVIRIS image volume. AVIRIS has 224 bands and 614×512 pixel resolution that corresponds to an area of approximately 11 km $\times$ 10 km on the ground. We have 1997 run of Jasper Ridge scene 1. For our experiments, we cropped the scene to $512 \times 512 \times 224$ pixels.

To quantify fidelity, the coding performances are reported using rate-distortion results, using root mean square error (RMSE) calculated over the whole sequence as the distortion measure. Note that the 16-bit value range is -32,768 to 32,767.

For the floating point filter resolution scalable 3D-SPECK, the RMSE values for a variety of bit rates in bits per pixel per band (bpppb) for the sequence are listed in Table 1 by comparing to the results of SNR scalable 3D-SPECK. The RMS error values listed in Table 1 for low resolution image sequences are calculated with respect to the reference image generated by the same analysis filter bank and synthesized to the same scale. Since our bit allocation algorithm is not optimal, SNR scalable version performs slightly better at the full scale. As there is significant high frequency content in hyperspectral images, resolution scalable 3D-SPECK yields lower RMSE values at partial scales.

Figure 2 demonstrates the reconstructed band 20, one band from the reconstructed sequence, decoded from a single resolution scalable code stream at 0.5 bpppb to a variety of resolutions. Even at low resolutions, we can get fine images. When the reconstructed sequences are presented at same display resolution, the perceived distortion for viewing a sample image at half resolution is equivalent to that at full resolution but from twice a distance. The low resolution sequences can thus be allowed to be coded relatively coarsely.

Bit Rate (Full) (bpppb)	**RMSE**			
	1/8	**1/4**	**1/2**	**Full**
Resolution scalable				
0.1	6.9	9.9	20.1	60.7
0.5	6.4	9.5	16.5	21.2
1.0	3.8	7.1	9.3	10.0
2.0	2.6	3.5	4.0	4.4
SNR scalable				
0.1	7.1	10.5	20.7	59.3
0.5	6.7	9.9	16.8	20.9
1.0	4.0	7.4	9.6	9.8
2.0	2.7	3.6	4.1	4.3

Table 1. RMS error at a variety of resolution and coding bit rates using floating point resolution scalable and SNR scalable 3D-SPECK.

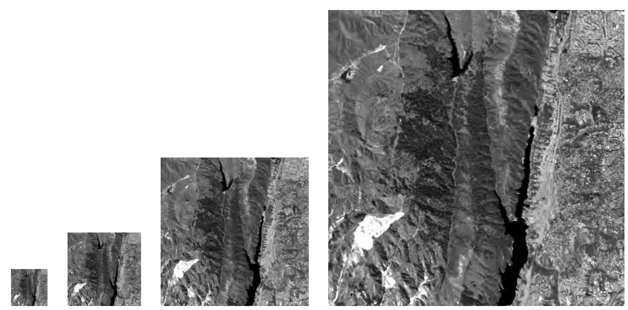

Fig. 2. A visual example of resolution progressive 3D-SPECK using floating point wavelet transform. From left to right: 1/8, 1/4, 1/2, and full resolution at 0.5 bpppb.

The basic function of integer filter implementation is the same as the floating point implementation. The difference is that the integer filter version supports lossy-to-lossless coding, and thus lossy and lossless reconstructions can be generated from the same embedded bitstream. The lossless decoding of resolution progressive 3D-SPECK is demonstrated in Figure 2 for the reconstructed band 20 of our test AVIRIS sequence at different resolutions.

With bit allocation, lossy reconstructions at different resolutions can be generated from the same embedded bitstream. Table 2 lists the RMSE results of the jasper scene 1 sequence reconstructed to different resolution levels at different bit rates by comparing both resolution and SNR scalable 3D-SPECK. Comparing to the values listed in Table 1, we could see that the floating point implementation performs better than the integer filter implementation on lossy im-

Bit Rate (Full) (bpppb)	RMSE			
	1/8	1/4	1/2	Full
Resolution scalable				
0.1	7.3	10.5	22.6	65.3
0.5	6.9	10.1	18.1	23.3
1.0	4.1	7.8	9.7	11.2
2.0	2.7	3.8	4.3	5.0
SNR scalable				
0.1	7.7	11.3	23.2	64.2
0.5	7.4	10.5	18.6	22.9
1.0	4.4	8.3	10.2	10.8
2.0	2.8	4.0	4.4	4.9

Table 2. RMS error at a variety of resolution and coding bit rates using integer filter resolution scalable and SNR scalable 3D-SPECK.

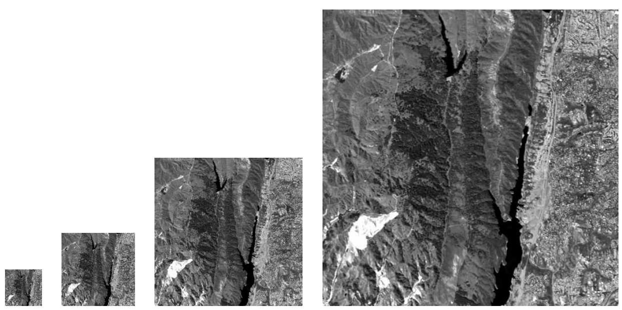

Fig. 3. A visual example of lossless resolution progressive 3D-SPECK. From left to right: 1/8, 1/4, 1/2, and full resolution (original).

age compression, demonstrating lower RMSE values at the same bit rates and resolutions. Again, SNR scalable 3D-SPECK yields slightly better performance at the full scale and worse performance at partial scales.

The corresponding byte budgets for the individual resolutions of the resolution scalable 3D-SPECK for Table 2 are provided in Table 3. We can see that the computational cost of decoding reduces from one resolution level to the next lower one. The total bit cost decreases rapidly with successive reductions in resolution. However, for SNR scalable 3D-SPECK, the decoder needs to visit the whole full bitstream in order to decode to a certain resolution level.

4. CONCLUSION

An embedded, block based, image wavelet transform coding algorithm of low complexity has been proposed. The algorithm has excellent performance for hyperspectral image compression. It supports both SNR and resolution scalable coding.

Bit Rate (bpppb)	Bit budget (accumulated Kbytes)			
	1/8	1/4	1/2	Full
0.1	61	274	734	734
0.5	68	290	963	3670
1.0	77	357	1505	7340
2.0	91	471	2422	14680

Table 3. Corresponding bit budgets for resolution scalable 3D-SPECK results for Table 2

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] M. Balakrishnan and W.A. Pearlman, "Hexagonal subband image coding with perceptual weighting," Optical Engineering, Vol. 32, No. 7, pp.1430-1437, July, 1993.

[3] 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.

[4] 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.

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

[6] 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.

[7] X. Tang, W.A. Pearlman and J.W. Modestino, "Hyperspectral image compression using three-dimensional wavelet coding," Visual Communications and Image Processing, SPIE/IS&T Electronic Imaging 2003, Proceedings of SPIE, Vol. 5022, pp.1037-1047, Jan. 2003.

[8] X. Tang and W.A. Pearlman, "Lossy-to-lossless block-based compression of hyperspectral volumetric data," IEEE International Conference on Image Processing (ICIP), Oct. 2004. To appear.