Hyperspectral Image Compression with Modified 3D SPECK

Ruzelita Ngadiran*, Said Boussakta*, Ahmed Bouridane** and Bayan Syarif*
* School of Electrical, Electronic and Computer Engineering, University of Newcastle upon Tyne, Newcastle upon Tyne, UK
** School of Computing, Engineering & Information Sciences, Northumbria University, Newcastle upon Tyne, UK

Abstract—Hyperspectral image consist of a set of contiguous images bands collected by a hyperspectral sensor. The large amount of data of hyperspectral images emphasizes the importance of efficient compression for storage and transmission. This paper proposes the simplified version of the three dimensional Set Partitioning Embedded block (3D SPECK) algorithm for lossy compression of hyperspectral image. A three dimensional discrete wavelet transform (3D DWT) can fully exploit the interband correlation in a volumetric block. This provides the obvious way to process data such as with hyperspectral images. The 3D structure of the SPECK algorithm will suitably extend the exploitation of interband dependence and correlation. The proposed algorithm maintains the 3D SPECK algorithm with modification of the implementation without using any list to store the significant information. This would reduce the memory requirement of the embedded coding algorithm and improve the coding time to compress the images.

I. INTRODUCTION

Hyperspectral data commonly possesses very good spatial resolution with super fine spectral details. This comes at the expense of a large dataset, for example with 224 subbands formed with 16 bit representation, the amount data for an image size of 512x614 pixels is about 134MB. Therefore, an efficient compression should be applied for efficient transmission and storage.

Discrete wavelet transform has been widely used in image compression because of its excellent de-correlation ability. Wavelet-based compression has managed to achieve rate scalability, high compression, embeddedness and progressive transmission. These excellent properties have been possible since the introduction of Shapiro's embedded zerotree wavelet (EZW) coder [1-2]. The EZW coder is based on significant tree quantization that exploits the similarities between the subband of the wavelet transform domain and the energy distribution of images through the subband. After that, Said and Pearlman proposed an improved version known as Set Partitioning in Hierarchical Trees (SPIHT) [3].

SPIHT is one of the most efficient wavelet-based compression algorithms and has become the benchmark for the current coder. The similarity of SPIHT and EZW is on the multipass significance testing of groups of pixels and bit plane progressive refinement. The difference is on its partitioning rule, where SPIHT encodes only direct descendants and maintains the tree until significant coefficients are found in the non-descendant tree, while EZW partitioned its tree if is significant [3]. The most important bits of the larger wavelet coefficients (magnitude) are then transmitted for progressive transmission while exploiting the inherent self similarities across different subbands. Another wavelet-based compression, Set partitioning embedded block coding (SPECK) was introduced later to provide a less complex alternative for an embedded coder. A SPECK coder has a comparable performance to SPIHT but with faster implementation. The partitioning rule in SPECK is block partitioning [4].

Hyperspectral/multispectral images require a 3D coder that will efficiently exploit the inter-band correlation between each frame. Within 3D wavelet transform domain, some of the popular existing 2D wavelet coders have been extended to 3D to suit these 3D image sources. For example, 3D SPIHT implemented are used to compress medical images [5] and multispectral images [7]. 3D SPECK also proved to be an excellent candidate in [8,9] and was less complex. A different approach for a 2D coder within a 3D wavelet domain also introduced in [10] with good a comparison to the original 3D coder.

In this paper, 3D SPECK algorithm is modified to provide faster implementation of large datasets. 3D SPECK is briefly explained in the next section, followed by the proposed algorithm. Section IV presents the implementation details and simulation results. This is followed by a conclusion in Section V.

II. 3D SPECK ALGORITHM: OVERVIEW

The extension of a SPECK algorithm to 3D is to make it suitable to 3D sources such as hyperspectral images. The 3D DWT transform domain can exploit the subsequent correlation in the hyperspectral images that contain tight statistical dependency along wavelength axis of its prism [7]. 3D SPECK maintains a block splitting algorithm to sort the significant pixels. If a code block contains significant coefficients it splits it into smaller sub-blocks. This technique can zoom to areas with high energy and code them first in which is suitable to exploit the presence of significant high frequency intra-band components.

3D SPECK is expected to be an excellent candidate for hyperspectral images due to the property of hyperspectral images having energy concentration in a high frequency band. The rate distortion curves prove that the SPECK algorithm out-performs SPIHT for images with higher
frequency content. This case is both true as proved in 2D and 3D cases in the initial SPECK development [4,7,8].

3D SPECK maintains two linked lists, List of Insignificant Sets (LIS) and List of Significant Pixels (LSP). It comprises of three stages, initialization, sorting and refinement. Sorting pass is based on a block splitting method after a significant test. Refinement pass is where some of the coefficient transmitted and the quantization process continued for the next decreased bit plane until a certain bit rate is achieved.

III. PROPOSED SPECK BASED CODER

The 3D DWT transformed coefficient decorrelates the hyperspectral image components spatially and spectrally. This produces some redundancy that can be exploited during the coding process. The proposed coder in this paper follows the basic sorting algorithm as in 3D SPECK. Therefore, inter-band dependence can be exploited automatically. The modification comes where the list used is removed and to compensate that, a state table marker is used to identify and keep track of the coefficients. Another modification relates to indexing a one dimensional array to offer some computational and organizational advantages. This technique can be similar to the algorithm applied for 2D images in [10-11].

The linear indexing system uses a single number to represent the index of coefficients, rather than three. To map the 3D coefficients to a 1D array, the mapping is based on a recursive Z curve or Morton ordering. This indexing is used instead of a raster scan because it performs better at preserving the locality. This order also suits the pyramid structure of the wavelet coefficients well. The indexing Z order scheme for 3D is shown in Figure 1. Linear indexing efficiently supports the organizational advantages. This technique can be similar to the algorithm applied for 2D images in [10-11].

The linear indexing system uses a single number to represent the index of coefficients, rather than three. To map the 3D coefficients to a 1D array, the mapping is based on a recursive Z curve or Morton ordering. This indexing is used instead of a raster scan because it performs better at preserving the locality. This order also suits the pyramid structure of the wavelet coefficients well. The indexing Z order scheme for 3D is shown in Figure 1. Linear indexing efficiently supports the operations carried out on the coefficients’ positions needed for the tree based algorithms with one operation instead of three, assuming the usual subband arrangement of wavelet coefficients.

![Figure 1. Z curve/ Morton order for 3D][12]

Following the linear indexing system, the number of coefficients is stored in a single array of length I, where I = Row x Column x Frame and is the magnitude array. The state table array, which is based on linear indexing, also has length I, with 4 bits per coefficient, which is the marked part. There is a one-to-one correspondence between magnitude and mark. In this proposed algorithm, the memory needed for it is fixed, given the size of the image. For example, the amount of memory required for an image sized 128x128x128, with an optional pre-computed array, 8 bits per pixel would require a total of 3 MB memory to use this algorithm. This actually gives a total saving of 67% compared to the original 3D SPECK algorithm. Unlike the original 3D SPECK, the dynamic memory is based on the list generated for images. With the use of two lists, the dynamic memory requirement can add up to 9 MB.

State table markers are placed in a four bit per coefficient state table mark, to keep track of the set partitions. Each element of the mark signifies a cube, indicating something about the corresponding element in the values image array. Each marker and its meaning are shown below:

- MIP: the pixel is significant
- MNP: the pixel is newly significant, so it will not be refined for this bitplane
- MSP: the pixel is significant and will be refined in this bitplane
- MS2: a block of size 2x2x2 (i.e., 8 elements) is to be skipped
- MS4: a block of size 4x4x4 (i.e., 64 elements) is to be skipped

An additional pointer is added to reduce the computational time required to calculate the number of MS2 markers.

In the sorting pass, the algorithm performs a significance test on the MIP, followed by all of the other blocks. All of the first elements of the initial block are marked by a marker and then it is partitioned by checking its significance. A block is significant if at least one of its coefficients is equal to or greater than the threshold. These are then partitioned into smaller sets for further significance testing.

The notation below is based on the original SPECK algorithm to describe the structure and terminology of the new proposed algorithm.

S is significant with respect to n, if

\[ \max_{i,j,k} |c_{i,j,k}| \geq 2^n \]  

Where \( c_{i,j,k} \) denotes the transformed coefficients at coordinates \((i, j, k)\), otherwise it is significant. The significance can be defined as the function below:

\[ S_n(T) = \begin{cases} 1: & 2^n \leq \max_{i,j,k} |c_{i,j,k}| \leq 2^{n+1} \\ 0: & \text{else} \end{cases} \]  

a) Initialization

- Output \( n = \left\lfloor \log_2 \left( \max_{i,j,k} |c_{i,j,k}| \right) \right\rfloor \)

- Map transformed coefficient to linear indexing (1D)

- State table initialization:
  - MIP for LLL (low-low-low) subband
  - MS* for each intial tree,
  - the marker defines the number of the coefficient to be skipped, MS2 is equal to 2x2x2 = 8 coefficient skip. (*refer to cube size 2,4,8,16 etc)

b) Sorting Pass

- Significance test for MIP
- Output \( S_n(S) \) whether the set is significant with respect to the current n or not
• If $S_i(S) = 1$
  If S is a pixel, output sign of S
Else check marker size perform 2
Partition S into eight equal subsets $O(S)$
• For each $O(S)$
  Output $S_i(O(S))$
If $S_i(O(S)) = 1$ output sign of $O(S)$
Else perform 2.
  
  c) Refinement Pass
  For each entry $(i, j, k) \in MSP$ except those included
  in the last sorting pass output $n^th$ MSB of $c_{i,j,k}$

  d) Quantization Step
  Decrement n by 1 and perform a sorting pass until a
  certain bit rate is achieved.
Significance tests are done using bitwise AND operations.
All operations can be implemented by bit shifting.

IV. NUMERICAL RESULT

Hyperspectral images from AVIRIS [13] and HYDICE [14] sensors used for the evaluation of the proposed algorithm. The tested images are listed below:

1. Terrain (307 pixels by 500 lines by 210 band)
2. Urban (307 pixels by 307 lines by 210 band),
3. Lunar Lake 1(512x 614 x224 band)
4. Lunar Lake 2(512x 614 x224 band)
5. Moffet field (512x 614 x224 band)
6. Cuprite (512x 614 x 224 bands)

Terrain and Urban images are from HYDICE sensors while the remaining are from AVIRIS. All images are represented by 16 bit data, so the original size of the images is very large. The simulation is carried out using MATLAB, so the images are cropped to an image of the same size at 256x256 pixels with 32 selected bands.

The 9/7 biorthogonal filters used to transform the hyperspectral images to wavelet with five per level decomposition at each dimension. The results shown are based on an average of 32 slides of randomly selected bands. The proposed algorithm has been compared with the original 3D SPECK. Full results obtained are reported in Table II.

A. Algorithm efficiency and memory allocation

Due to the nature of programming in Matlab, combined with the large size of the hyperspectral images, the time taken for both algorithms is quite long. However, the time consumption can be estimated by algorithm efficiency and its memory allocation.

The complexity of coding a bit plane can roughly be measured by the number of bit comparisons. Both algorithms only use the most basic operations like memory access, bit shift, additions and comparisons since the multiplication and division operations are not required.

The difference is only that 3D SPECK requires list management, while the proposed algorithm uses direct mapping to state table to track the coefficients.

3D SPECK algorithm maintains two linked lists. The list contains set of size and significant pixels. For volume size of 256x256x16 it would require a total of 20 bits per coefficients to store its coordinates (8-row, 8-column, 4-frame). So, for two list entries, the total memory required is about 256x256x16x20x2/(8x1024x1024) = 5 MB. However, for the proposed algorithm the memory required is 256x256x16x8+(4)/(8x1024x1024) = 1.5 MB The memory calculation does not take into account the memory required for the wavelet transform, because both coders employ the same transformed coefficients.

The proposed algorithm memory requirement is about 1/3 smaller than the original 3D SPECK. This would lead to a faster encoding and decoding process, up to three times faster than 3D SPECK.

B. Rate distortion performance

Table 1 shows the average rate distortion for the tested hyperspectral image with selected bands. The result shows that the rate distortion difference ranges from 0.1 dB to 1.8 dB, compared to the proposed algorithm on all images at low bit rate (0.1 to 1). This is predictable since 3D SPECK performs very well on images with high frequency content. However, the proposed algorithm shows better rate distortion at bit rate 2 in Cuprite and Moffet. This would suggest that the proposed algorithm would perform better at higher bit rate or maybe to lossless compression. But, further work needs to be carried out to further demonstrate the finding.

<table>
<thead>
<tr>
<th>Image</th>
<th>Bit rate</th>
<th>3D SPECK</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terrain</td>
<td>0.1</td>
<td>24.2724</td>
<td>23.7470</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>25.7430</td>
<td>25.6141</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>27.6367</td>
<td>27.0521</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>31.5106</td>
<td>29.7298</td>
</tr>
<tr>
<td>Urban</td>
<td>0.1</td>
<td>26.0493</td>
<td>25.4126</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>28.2505</td>
<td>27.5996</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>30.2143</td>
<td>29.4753</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>34.2288</td>
<td>32.3534</td>
</tr>
<tr>
<td>Lunar Lake</td>
<td>0.1</td>
<td>28.2940</td>
<td>27.2796</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>32.9460</td>
<td>32.3974</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>37.2054</td>
<td>36.4579</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>42.2519</td>
<td>41.1214</td>
</tr>
<tr>
<td>Lunar 2</td>
<td>0.1</td>
<td>28.8242</td>
<td>27.6569</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>33.0693</td>
<td>32.5395</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>37.4446</td>
<td>36.6421</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>42.3655</td>
<td>41.3401</td>
</tr>
<tr>
<td>Moffet</td>
<td>0.1</td>
<td>37.7965</td>
<td>35.9691</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>44.6717</td>
<td>44.1999</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>51.0352</td>
<td>51.4574</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>57.2607</td>
<td>56.3813</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>61.6714</td>
<td>61.7708</td>
</tr>
<tr>
<td>Cuprite</td>
<td>0.1</td>
<td>26.69807</td>
<td>25.5146</td>
</tr>
<tr>
<td></td>
<td>0.25</td>
<td>30.6765</td>
<td>30.0467</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>35.0285</td>
<td>34.1236</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>37.3060</td>
<td>38.7514</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>37.3060</td>
<td>45.0558</td>
</tr>
</tbody>
</table>

The values in Table 1 are plotted in Figure 2, 3 and 4. The figures combine a result of similar image properties. In figure 2, the HYDICE image consists of Terrain and Urban, plotted together. Figure 3 shows the rate distortion of the Lunar image. Both figures show that 3D SPECK performs better than the proposed algorithm at about 0.1 dB to 1.7 dB. In Figure 4 Moffet and Cuprite outperform
3D SPECK at bit rate 2. An example of the compressed image (Urban) at bit rate 0.1 using both algorithms can be found in Figure 5. The actual rate distortion value discrepancy is about 1.2 dB. Although 3D SPECK performs better than the proposed algorithm on all hyperspectral images at low bit rate, it still provides a good alternative for a low memory wavelet based coder.

V. CONCLUSION

In this paper we have proposed a modified 3D SPECK for lossy compression of a hyperspectral image. The proposed algorithm can reduce memory occupancy in the coding process and increase the coding speed effectively. The algorithm is applied on a 3D wavelet transform domain, and uses the 3D SPECK algorithm without a list to provide a low complexity coder. Although the simulation result shows that the performance of the proposed algorithm did not surpass the 3D SPECK at low bit rate, it still provides a good alternative for a low complexity wavelet based embedded coder. At higher bit rates however some promising results can be seen. This suggests that the proposed algorithm might be suitable for lossless compression. Future work will focus on lossless compression for a hyperspectral image using an adaptation of this algorithm.
ACKNOWLEDGMENT

R. Ngadiran would like to thank The Universiti Of Malaysia Perlis (UniMaP) and the Ministry of Higher Education Malaysia for the for scholarship and financial support of her PhD studies at Newcastle University, United Kingdom.

REFERENCES