Hyperspectral image compression using three-dimensional significance tree splitting

Jing Huang (黄 菁), Rihong Zhu (朱日宏), Jianxin Li (李建欣), and Yong He (何 勇)

School of Electronic Engineering and Photoelectric Technology, Nanjing University of Science and Technology, Nanjing 210094

Received November 28, 2006

A three-dimensional (3D) wavelet coder based on 3D significance tree splitting is proposed for hyperspectral image compression. 3D discrete wavelet transform (DWT) is applied to explore the spatial and spectral correlations. Then the 3D significance tree structure is constructed in 3D wavelet domain, and wavelet coefficients are encoded via 3D significance tree splitting. This proposed algorithm does not need to use ordered lists, moreover it has less complexity and requires lower fixed memory than 3D set partitioning in hierarchical trees (SPIHT) algorithm and 3D set partitioned embedded block (SPECK) algorithm. The numerical experiments on AVIRIS images show that the proposed algorithm outperforms 3D SPECK, and has a minor loss of performance compared with 3D SPIHT. This algorithm is suitable for simple hardware implementation and can be applied to progressive transmission.

OCIS codes: 100.0100, 100.2000, 100.6890.

Hyperspectral images that have been widely used in military and civilian applications are massively large sized three-dimensional (3D) data sets. Efficient compression needs to be applied to these data sets in order to reduce the storage and bandwidth costs. In recent years, 3D wavelet image compression algorithms based on 3D set partitioning in hierarchical trees (SPIHT)[1−3] and 3D set partitioned embedded block (SPECK)[4−6] have been proposed for progressive hyperspectral image compression. The ordered lists are used in these algorithms to store the coordinates of significance coefficients and subsets in the sorting order. The use of lists poses some drawbacks for hardware implementation in that a large amount of unfixed memory is needed to maintain these lists and the operations of the list nodes increase the complexity of algorithms.

In this paper, we extend our recently proposed two-dimensional (2D) embedded wavelet coder based on significance tree splitting[7] to three dimensions for hyperspectral image compression. We call this new coding technique 3D significance tree splitting. First, 3D discrete wavelet transform (DWT) is used to exploit the spatial and spectral correlations. Next, the 3D significance tree structure is constructed from the 3D orientation tree and the wavelet coefficients are encoded via 3D significance tree splitting. This algorithm does not require lists and it is suitable for hardware implementation.

As hyperspectral images have a tight statistical dependency along both spatial and spectral directions, 3D DWT can exploit spatial and spectral correlations. Here, we first apply 2D dyadic wavelet decomposition on each spectrum of image, and then apply one-dimensional (1D) dyadic wavelet decomposition on the spectral dimension. In the resulting 3D wavelet domain, most of the energy is concentrated in the low frequency subbands. The spatial relationship on the hierarchical pyramid of the coefficients can be represented as the 3D orientation tree structure. Figure 1 shows the 3D orientation tree structure after two-level decomposition. All the coefficients are organized by 3D orientation trees with roots located at the low-low-low (LLL) subband. Each root node in LLL subband has seven offspring which correspond to the pixels of the seven different spatial orientations at the next finer scale of the pyramid. Except of the highest frequency subbands and LLL subband, each node has eight offspring corresponding to pixels of the same spatial orientation at the next finer level of the pyramid, and these eight offspring form a group of $2 \times 2 \times 2$ adjacent pixels. Let us define $O(i, j, k)$ as the offspring of the pixel (i, j, k). So for the LLL subband we have

$$O(i,j,k) = \{(i, j + h_{\mathrm{LLL}}, k), (i + w_{\mathrm{LLL}}, j, k), (i, j, k + l_{\mathrm{LLL}}), (i + w_{\mathrm{LLL}}, j + h_{\mathrm{LLL}}, k), (i + w_{\mathrm{LLL}}, j, k + l_{\mathrm{LLL}}), (i, j + h_{\mathrm{LLL}}, k + l_{\mathrm{LLL}}), (i + w_{\mathrm{LLL}}, j + h_{\mathrm{LLL}}, k + l_{\mathrm{LLL}})\}, \quad (1)$$

and for the subbands with the exception of the highest frequency subbands and LLL subband we have

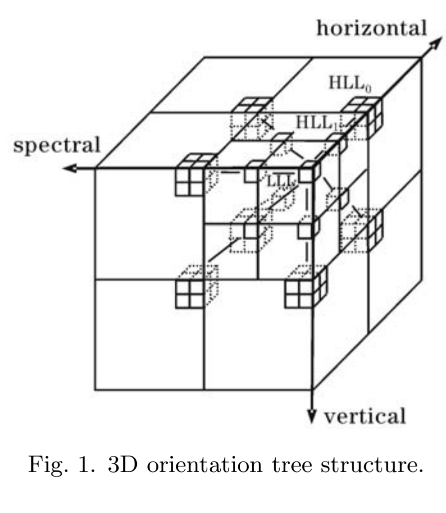

Fig. 1. 3D orientation tree structure.

$$O(i,j,k)=\{(2i,2j,2k),(2i+1,2j,2k),(2i,2j+1,2k),$$
$$(2i,2j,2k+1),\ (2i+1,2j+1,2k),$$
$$(2i,2j+1,2k+1),(2i+1,2j,2k+1),$$
$$(2i+1,2j+1,2k+1)\}. \quad (2)$$

In Eq. (1) $h_{\rm LLL}$, $w_{\rm LLL}$ and $l_{\rm LLL}$ are the height, width, and length of LLL subband, respectively.

During the sorting pass for each bitplane as in 3D SPIHT and 3D SPECK, we need to scan all the coefficients in the set of tree or block to judge the significance of the set, which leads to repeated operations in every bitplane. We are trying to find a way to make this significance test convenient. In addition, the coefficients of the different spatial orientations on the same level of the pyramid also have correlations. For the above two reasons, we construct another tree structure, called 3D significance tree, from 3D orientation tree structure. In order to represent the significance tree structure, we call the nodes at the same spatial location of different spatial orientation subbands at the same scale of pyramid the neighbouring nodes. Let $N(i,j,k)$ denote as the neighbouring nodes of the node (i,j,k). Here we only use the neighbouring nodes of the node in high-low-low (HLL) subbands as shown in Fig. 1. $N(i,j,k)$ is defined as

$$N(i,j,k)=\{(i,j+h_m,k),(i,j,k+l_m),$$
$$(i,j+h_m,k+l_m),(i-w_m,j+h_m,k),$$
$$(i-w_m,j+h_m,k+l_m),(i-w_m,j,k+l_m)\},$$
$$(i,j,k)\in {\rm HLL}_m,\quad m=0,\cdots,D_{\max}-1 \quad (3)$$

where h_m, w_m, and l_m are respectively the height, width, and length of the HLL subband of the mth level in the pyramid, $D_{\max}$ is the total wavelet decomposition level. Then we define $P(i,j,k)$ as the parent node of the node (i,j,k) in HLL subbands, and we have

$$P(i,j,k)$$
$$=\left\{\begin{array}{l}(i-w_{D_{\max}-1},j,k),\\ \quad (i,j,k)\in {\rm HLL}_{D_{\max}-1}\\ (\lfloor i/2\rfloor,\lfloor j/2\rfloor,\lfloor k/2\rfloor),\\ \quad (i,j,k)\in({\rm HLL}_0\bigcup{\rm HLL}_1\cdots\bigcup{\rm HLL}_{D_{\max}-2})\end{array}\right., \quad (4)$$

where $w_{D_{\max}-1}$ is the width of the ${\rm HLL}_{D_{\max-1}}$ subband. The relationship between parent node and its offspring in ${\rm HLL}_{D_{\max-1}}$ subband is different from that in other HLL subbands.

After we defined the neighbouring nodes and parent nodes of the node in HLL subbands, let us define ${\rm ST}(i,j,k,d)$ as the value of the significance tree node at depth d with the pixel (i,j,k). These pixels are only in LLL subband and HLL subbands. We use $C_{i,j,k}$ to represent the value of the pixel (i,j,k) in the wavelet domain. For the lowest depth $d=0$, ${\rm ST}(i,j,k,d)$ is set to the maximum absolute value of $C_{i,j,k}$ and the value of all pixels in $N(i,j,k)$. For the higher depths, ${\rm ST}(i,j,k,d)$ is set to the maximum absolute value of $C_{i,j,k}$, the value of all pixels in $N(i,j,k)$ and the value of the offspring of the

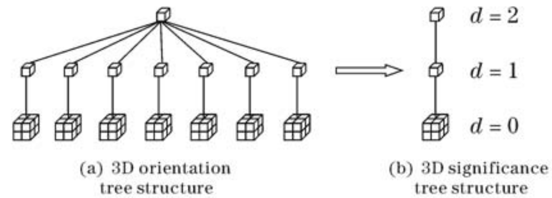

Fig. 2. Construction of 3D significance tree from 3D orientation tree.

node (i,j,k) at depth $d-1$ of the significance tree. For the highest depth $d=D_{\max}$, ${\rm ST}(i,j,k,d)$ is set to the maximum absolute value of $C_{i,j,k}$ and the value of the offspring of the node (i,j,k) at depth $D_{\max}-1$ of the significance tree. Figure 2 shows the construction of the significance tree. Steps for constructing the significance tree are listed as follows.

1) Set the nodes at depth $d=0$,

$${\rm ST}(i,j,k,d)$$
$$=\max\{{\rm abs}(C_{i,j,k},C_{u,v,w},\ (u,v,w)\in N(i,j,k))\},$$
$$(i,j,k)\in {\rm HLL}_0. \quad (5)$$

2) Set the nodes at depth $d=1,2,\cdots,D_{\max}-1$,

$${\rm ST}(i,j,k,d)$$
$$=\max\{{\rm abs}(C_{i,j,k},C_{u,v,w},(u,v,w)\in N(i,j,k)),$$
$${\rm ST}(x,y,z,d-1),(x,y,z)\in O(i,j,k)\},$$
$$(i,j,k)\in {\rm HLL}_d \quad (6)$$

3) Set the nodes at depth $d=D_{\max}$,

$${\rm ST}(i,j,k,d)=\max\{{\rm abs}(C_{i,j,k}),{\rm ST}(x,y,z,d-1),$$
$$(x,y,z)\in O(i,j,k)\},$$
$$(i,j,k)\in {\rm LLL}. \quad (7)$$

From the idea of constructing the 3D significance tree, we can observe that the value of each node is the largest magnitude of the same spatial-location node, its neighbouring nodes and all of their descendants in 3D orientation tree. If the node in significance tree is significant, it indicates that the set including the corresponding tree coefficients is significant. The significance tree provides significance information of the coefficients, so we only need to test the significance of the node in significance tree to judge the significance of its corresponding set of tree. This avoids scanning all the coefficients in the set repeatedly for each bitplane.

After we build up the significance tree, the coefficients are encoded via 3D significance tree splitting. Since 3D significance tree can provide the significance information of the coefficients, the encoding of the coefficients is guided by the significance test of the significance tree node. We define the significance function of the node in 3D significance tree or the coefficient in wavelet domain at bitplane n as

$$S_n(T)=\left\{\begin{array}{ll}1, & \text{if } |T|\geq 2^n\\ 0, & \text{otherwise}\end{array}\right.. \quad (8)$$

Below we present the encoding algorithm using 3D significance tree splitting:

1) Initialization
  Output $n = \lfloor \log_2(\max_{(i,j,k)\in \text{LLL}} \text{ST}(i,j,k,D_{\max})) \rfloor$.
2) Encoding Step
  For $d = D_{\max}$ to 0,
   Encode (d, n).
3) Quantization Step
  Decrement n by 1 and go to step 2).

```
Encode (d, n)
  If d = D_max
    For each (i, j, k) in LLL subband
      If S_{n+1}(T(i, j, k, d)) = 0
        Output S_n(T(i, j, k, d))
      If S_n(T(i, j, k, d)) = 1
        Encode coefficient (i, j, k, n)
  Else
    For each (i, j, k) in HLL_d subband
      If S_n(T(P(i, j, k), d + 1)) = 1
        If S_{n+1}(T(i, j, k, d)) = 0
          Output S_n(T(i, j, k, d))
        If S_n(T(i, j, k, d)) = 1
          For each (x, y, z) ∈ {(i, j, k), N(i, j, k)}
            Encode coefficient (x, y, z, n)
Encode coefficient (i, j, k, n)
  If S_{n+1}(C_{i,j,k}) = 1
    Output the nth MSB of |C_{i,j,k}|
  Else
    Output S_n(C_{i,j,k})
    If S_n(C_{i,j,k}) = 1
      Output the sign of the C_{i,j,k}.
```

The coding scheme for each bitplane is performed from the lowest resolution subband to higher resolution subbands, and subbands of the same resolution are encoded together. Unlike other hierarchical coders, we utilize the significance tree to test the significance of the group of coefficients. In the encoding of the significance tree node corresponding to the pixel in LLL subband, we first test the significance of the node in significance tree that has never been significant and output its significance. If it is significant, the coefficient of the same spatial location in wavelet domain is encoded. While encoding the significance tree node corresponding to the pixel in HLL subbands, we first test the significance of its parent node in the significance tree. If parent node has been significant, it indicates that there is the possibility that its children nodes are significant. Then we test the significance of the node that has never been significant in significance tree and output its significance. If it is significant, the coefficient of the same spatial location and its neighbouring nodes are encoded together. While encoding the coefficient, if it has been significant before the last encoding step, we encode its bit of the current bitplane; otherwise, we test its significance, once it is significant, output its sign. The last step of this algorithm decreases the bitplane by 1 and returns to the encoding step. This step makes the whole encoding process progressive.

The bitplane decoding scheme basically follows the same procedure as the encoding scheme. The significance tree build-up stage is not needed at the decoder because the significance information is already contained in the encoded bit stream. Given the encoded sequences of the significance tree test, the decoder can duplicate the steps taken by the encoder. The decoding has less computational complexity than encoding.

This algorithm does not use ordered lists that is always used in 3D SPIHT and 3D SPECK. Although the utilization of ordered lists enables the encoding process to be decomposed into multipass to follow the decreasing order of the perceived R-D significance levels, it also has some drawbacks. Firstly, the memory required by these lists is not fixed for it not only depends on the size of image but also on the characteristics of image. It will increase along with the embedded encoding process. Secondly, the inserting and deleting operations of these lists increase the complexity of the algorithm. The complexity of the algorithm will increase along with the encoding process, because at high bit rate there will be a lot of nodes in lists. The unfixed memory requirement and complexity make the algorithm using lists not an effective compression algorithm for hardware implementation. In our algorithm, we use significance tree instead of ordered lists to track the significance status of the coefficients. In addition, the utilization of significance tree simplifies the significance test of the set of tree because the significance of one node indicates the significance of its corresponding set of tree, and as a result we need not scan all of the coefficients in the set of tree. During the encoding and decoding process the sorting pass and refinement pass are merged into just one pass that results in a simple control flow implementation and the memory required for the significance tree is fixed only to the size of LLL subband

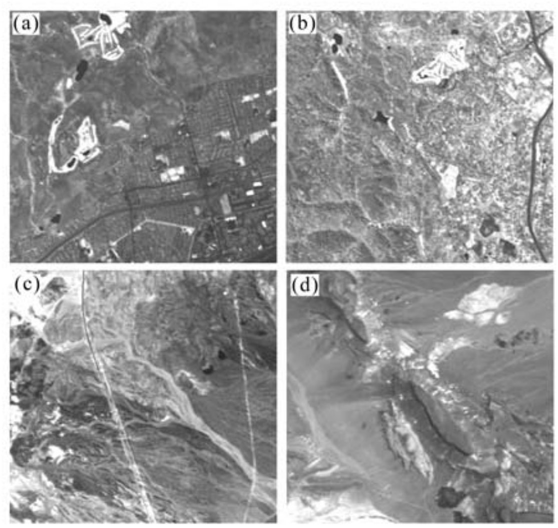

Fig. 3. Hyperspectral images. (a) Band 70 of "Moffet" test image; (b) band 60 of "Jasper" test image; (c) band 65 of "Cuprite" test image; (d) band 55 of "Lunar" test image.

Table 1. Lossless Compression Ratio

Hyperspecral Image	Compression Ratio		
	3D SPECK	3D SPIHT	Proposed
Moffet	1.540	1.562	1.544
Jasper	1.475	1.501	1.482
Cuprite	1.505	1.535	1.512
Lunar	1.586	1.613	1.595

Table 2. Rate Distortion Performance Comparison

Bit Rate (bpppb)	SNR (dB)											
	Moffet			Jasper			Cuprite			Lunar		
	3D SPECK	3D SPIHT	Proposed	3D SPECK	3D SPIHT	Proposed	3D SPECK	3D SPIHT	Proposed	3D SPECK	3D SPIHT	Proposed
0.5	28.912	30.164	29.401	26.442	26.953	26.426	37.233	38.589	37.712	36.977	37.892	37.022
1.0	36.347	37.497	36.696	32.053	32.900	32.260	43.488	44.866	43.975	43.482	44.093	43.560
1.5	42.032	43.041	42.348	37.910	39.141	38.510	47.421	48.488	47.856	47.768	48.780	48.014
2.0	46.070	46.972	46.369	44.047	44.840	44.253	50.847	51.812	51.152	50.908	51.807	51.116
2.5	48.830	49.550	49.114	48.179	48.894	48.259	53.411	54.092	53.759	53.220	54.075	53.629
3.0	51.784	52.138	52.056	50.142	51.141	50.618	56.003	56.910	56.469	55.661	56.713	56.320

and HLL subbands, which is related to the image size and decomposition level. Thus the proposed coder can achieve lower complexity and fixed predetermined memory requirement.

The performance of the proposed algorithm is evaluated on four AVIRIS images. These hyperspectral images are 512 lines by 614 columns and 224 spectral bands. We select the data set of size $256 \times 256 \times 32$ and 16 bits per pixel as the test set in the experiments. Figure 3 shows examples of one band of these four hyperspectral images. We use four-level dyadic decomposition with the 9/7 integer lifting scheme[8] along three directions of the data set. The lossless compression performances of 3D SPIHT, 3D SPECK, and our algorithm are illustrated in Table 1. Our algorithm exhibits a loss of average 1.2% compared with 3D SPIHT and a better ratio of average 0.4% compared with 3D SPECK. We also compare the lossy compression performances at different bit rates of these three algorithms by means of signal-to-noise ratio (SNR)

$$\mathrm{SNR} = 10\log_{10}\frac{\sigma_x^2}{\mathrm{MSE}}\ (\mathrm{dB}), \qquad (9)$$

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. Table 2 shows the lossy compression performances of three algorithms. The experimental results reveal that the proposed algorithm exhibits better performance of about average 0.3 dB compared with 3D SPECK and a performance loss of about average 0.6 dB compared with 3D SPIHT.

This paper proposes a 3D wavelet based coder via significance tree splitting for hyperspectral image compression. The 3D wavelet transform is applied to exploit both the spectral and spatial correlations. 3D significance tree structure is established in 3D wavelet domain, and then every bitplane is encoded via significance tree splitting. The proposed algorithm effectively exploits the intraband correlation and interband correlation in wavelet domain, and it has low complexity and fixed low memory requirement. It is suitable for hardware implementation. Besides, the proposed algorithm is embedded and can be applied to progressive transmission.

In our future work, we will focus on the wavelet packet decomposition scheme for 3D wavelet transform to exploit the spectral and spatial correlations of hyperspectral images more effectively, and then modify the establishment of the 3D significance tree and the corresponding coding algorithm according to the 3D wavelet packet transform.

This work was supported by the China Academy of Space Technology (CAST200641) and Jiangsu Key Laboratory of Modern Optic Technology (No. T8108030). J. Huang's e-mail address is jingjing@vip.sina.com.

References

1. P. L. Dragotti, G. Poggi, and A. R. P. Ragozini, IEEE Trans. Geoscience and Remote Sensing **38,** 416 (2000).
2. Y. Zhou, Y. Li, and C. Wu, Acta Opt. Sin. (in Chinese) **26,** 336 (2006).
3. Y. Li, C. Wu, J. Chen, and L. Xiang, Acta Opt. Sin. (in Chinese) **21,** 691 (2001).
4. X. Tang, W. A. Pearlman, and J. W, Modestino, Proc. SPIE **5022,** 1037 (2003).
5. J. Wu and C. Wu, Chin. Opt. Lett. **2,** 325 (2004).
6. J. Wu, Z. Wu, and C. Wu, Proc. SPIE **5637,** 484 (2005).
7. J. Huang, R. Zhu, and J. Li, in *Proceedings of 6th International Progress on Wavelet Analysis and Active Media Technology* **1,** 128 (2005).
8. Z. Xiong, X. Wu, S. Cheng, and J. Hua, IEEE Trans. Medical Imaging **22,** 459 (2003).