Multispectral image compression using three-dimensional transform zeroblock coding

Jiaji Wu (吴家骥) and Chengke Wu (吴成柯)

National Key Laboratory on ISN, Xidian University, Xi'an 710071

Received December 9, 2003

This paper proposes a new multispectral image data compression algorithm (KLT/WT-3DEZB). The proposed coding strategy consists of three main steps. Firstly, a wavelet transform (WT) is applied to reduce the spatial redundancies. Then, a Karhunen-Loeve transform (KLT) is used to reduce the redundancies in the spectral domain. Finally, a modified SPECK algorithm—three-dimensional embedded zeroblock (3DEZB) algorithm is proposed and used to encode the transformed coefficients. Numerical experiments show that the reconstructed images using the proposed algorithm exhibite a better quality and a higher compression ratio than those obtained by traditional KLT/WT-3DSPIHT, 3DSPIHT, and 3DSPECK algorithms.

OCIS codes: 100.0100, 280.0280.

Multispectral images require large amounts of storage space, therefore a lot of attention has recently been focused to compress these images. Multispectral images include both spatial and spectral redundancies. Usually, we can use vector quantization, prediction, and transform coding to reduce the redundancies. For example, a hybrid transform/vector quantization (VQ) coding scheme is proposed[1]. Instead, Karhunen-Loeve transform (KLT) is used to reduce the spectral redundancies, followed by a two-dimensional (2D) discrete cosine transform (DCT) to reduce the spatial redundancies[2]. A quadtree technique for determining the transform block size and the quantizer for encoding the transform coefficients was applied across KLT-DCT method[3]. In Refs. [4] and [5], the researchers used a wavelet transform (WT) to reduce the spatial redundancies and KLT to reduce the spectral redundancies, and then encoded using the three-dimensional SPIHT (3D SPIHT) algorithm. Recently, Tang *et al.*[6] take 3D WT on hyperspectral images and apply 3DWT-3DSPECK to encode the transformed coefficient. However, 3DWT-3DSPECK does not work efficiently for multispectral images having low spectral resolution, such as the Landsat TM images.

In this paper, we extend and improve SPECK[7], EZBC[8], SBHP[9], and the algorithm in Ref. [3], and propose a new Karhunen-Loeve transform/wavelet transform-3D embedded zeroblock (KLT/WT-3DEZB) algorithm. It exhibits the following properties: high compression ratio, low complexity, and progressive transmission.

The best compression method exploits redundancies in both the spatial and spectral dimensions for multispectral image. In our method, a spatial transform (wavelet transform) is firstly applied[10]. Then the spectral components of each spatial frequency band are decorrelated using a KLT.

The multispectral image data is given as

$$X = \{X_1, X_2\ , X_3, \cdots , X_n\}^T\ , \tag{1}$$

where the subscript n denotes the number of bands, which equals seven in the case of Landsat TM; X_n represents the sequence of the different spectral images. The multispectral image data W after a WT is represented as

$$X \xrightarrow{\text{WT}} W = \{W_1, W_2, W_3, \cdots , W_n\}^T\ . \tag{2}$$

The covariance matrix C_w is defined as

$$\begin{aligned} C_w &= E\left\{(W - m_w)(W - m_w)^T\right\} \\ &= \frac{1}{M}\sum_{i=0}^{M-1}(W_i - m_w)(W_i - m_w)^T\ , \\ m_w &= E\{W\} = \frac{1}{M}\sum_{i=0}^{M-1}W_i, \end{aligned} \tag{3}$$

where M represents the number of the wavelet coefficients vectors. Let λ_i be real and the positive eigenvalues of matrix C_w, and e_i be the corresponding eigenvectors. When the eigenvalues λ_i are arranged in descending order so that $\lambda_1 \ge \lambda_2 \ge \lambda_3 \ge \cdots \ge \lambda_i$, the auto-correlations of transformed signal vector are arranged in descending order. The transformation matrix T is represented as $T = \{e_1, e_2\ , e_3, \cdots , e_n\}^T$. The matrix Y, a result of the KLT, is represented as $Y = TW$.

The WT and the KLT concentrate much of the energy into a single band, improving overall coding efficiency.

The zeroblock algorithm is based on the quadtree decomposition applied to wavelet bitplane image coding in the literature, for example, SPECK, EZBC, SBHP, and EBCOT[11]. Quadtree decomposition employed by zeroblock coders is conceptually simple, efficient, and straightforward. Whenever a given set (block) tests significant, it is partitioned into four equal subsets (subblocks). If not significant, large areas of zero coefficients are called as the zeroblock and can be indicated as a bit "0". If a parent set tests significant, at least one of its four children will test significant in the subsequent descendant test. In other words, if a set is significant and its first tested three subsets are insignificant, then obviously the fourth subset must be significant for sure and no testing and no coding. Using this approach, we improved quadtree algorithm, reduced some bits outputs, and enhanced the coding efficiency. We also discovered in the experiment, if only one is important in four subsets, then the important probability of the right under foot subset is the biggest.

We use the function

$$\Gamma_n(\tau) = \begin{cases} 1, & 2^n \leq \max_{(i,j,k)\in\tau} |c_{i,j,k}| < 2^{n+1} \\ 0, & \text{otherwise} \end{cases}$$

to indicate the significance of a set τ of pixels, $c_{i,j,k}$ is the coefficients at position (i, j, k). In the transformed image at least one $c_{i,j,k}$ is significant against the threshold $n_{\max}$.

The algorithm starts by partitioning the transformed multispectral images into two parts: part ν which is the topmost spectral band where the energy is the most centralized, and part H which is everything that is left of the spectrum bands. According to the wavelet subband size, we divide part ν into $3L + 1$ sets of type S, where L is the number of the wavelet decomposition levels. Part H alone is treated as one set of type I. To start the algorithm, set S and set I are added to the list of insignificant sets (LIS).

If the tested set S is significant against the threshold n, it is partitioned into four subsets. If not significant, it stays in the LIS (If S is one pixel, set S is added to the list of insignificant pixels (LIP)). The set I is processed by yet another partitioning scheme. If set I is found to be significant, it is partitioned into two sets—one of type S and one of type I. Unceasingly iterating these sets of type S and type I to test whether they are significant, until LIS is empty. The concrete algorithm is as follows.

Definition:

LIP: list of insignificant pixels.

LIS: list of insignificant sets.

LSP: list of significant pixels.

$c_{i,j,k}$: transformed coefficients at position (i, j, k).

1) Initialization

• output $n_{\max} = \lfloor \log_2(\max\{|c_{i,j,k}|\}) \rfloor$

• add $3L + 1$ sets of type S and one set of type I to LIS; set the LIP and LSP as empty lists

2) Sorting Pass

• in increasing order of size of sets of type S

— for each set $S \in$ LIP do Process S (S)

— for each set $S \in$ LIS do Process S (S)

— for each set $I \in$ LIS do Process I (I)

3) Refinement Pass

• for each pixel in LSP, except those included in the last sorting pass, output the n-th most significant bit of $|c_{i,j,k}|$

4) Quantization Step

• decrement n by 1, and go to step 2

Procedure Process S (S)

1) output $\Gamma_n(S)$

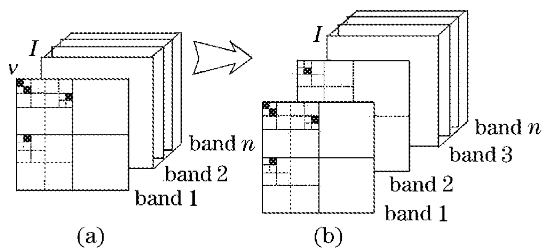

Fig. 1. Illustration of 3DEZB algorithm. (a): Partitioning of the transformed image into ν and I; (b): partitioning of set I.

2) if $\Gamma_n(S)$=1

• if S is a pixel, output its sign, add S to LSP

• else Quadtrees S (S)

3) else

• if S is a pixel, add S to LIP

• else add S to LIS

4) return

Procedure Process I (I)

1) output $\Gamma_n(I)$

2) if $\Gamma_n(I)$=1

• partition I into two sets: one of type S and one of type I (see Fig. 1)

• for the set S do Process S (S)

• for the set I do Process S (I)

3) return

Procedure Quadtrees S (S)

1) partition S into four equal subsets S_i, $i = 0, 1, 2, 3$

2) for each set S_i do

• output $\Gamma_n(S_i)$

• if ($\Gamma_n(S_1)$=0 and $\Gamma_n(S_2)$=0 and $\Gamma_n(S_3)$=0), do not output $\Gamma_n(S_4)$; else output $\Gamma_n(S_4)$

• if $\Gamma_n(S_i)$=1

—if S_i is a pixel, output its sign and add S_i to LSP

—else Quadtrees S (S_i)

• else

—if S_i is a pixel, add S_i to LIP

—else add S_i to LIS

3) return

To optimize the rate-distortion properties of the embedded bit stream, one important step of the algorithm is that the sets of type S are processed in increasing order of their size in step 2. We sort the elements in the LIS, LIP, and LSP[7].

LIS and LIP: pixels added first are coded first (FIFO).

LIS: sets are processed in increasing order of their size. When sets have the same size, sets split first are processed first.

Our simulations were carried on a TM image of Basin (see Fig. 2(a)) which is distributed with the ENVI software and a 128-band hyperspectral image of Village (see Fig. 2(b)). Each image consists of 512 × 512 pixels at 8 bit per pixel. The thermal band 6 of TM image had different spatial resolution, therefore, it was not considered in the experiment. WT is applied with the 4-level 9/7 tap biorthogonal filters[10] firstly, followed by KLT. Finally, the coefficients are encoded using the 3DEZB algorithm. Also, the proposed KLT/WT-3DEZB is compared with KLT/WT-3DSPIHT[5], 3DWT-3DSPECK[6], 3DWT-3DSPIHT[12], 2DSPECK[7] and 2DSPIHT[13], for test images Basin and Village. In our experiments, 3DWT technique works for hyperspectral but not for multispectral data. We select 16 bands as a test set from the Village image. All simulations are measured on a PC running P4 2.4-GHz processor.

Figures 3 and 4 show the rate-distortion curves for coding the Basin image and the Village image over a wild bit rate range using KLT/WT-3DEZB, KLT/WT-3DSPIHT, 3DWT-3DSPECK, 3DWT-3DSPIHT, 2DSPECK, and 2DSPIHT, respectively. Compared with the 2DSPIHT (band-by-band SPIHT) and 2DSPECK (band-by-band SPECK), obviously, 3D version obtains better performance than band-by-band

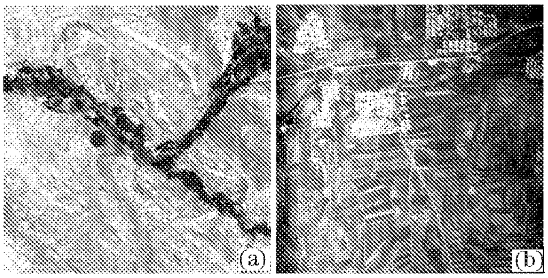

Fig. 2. (a): Band 1 of Basin image; (b) band 18 of Village image.

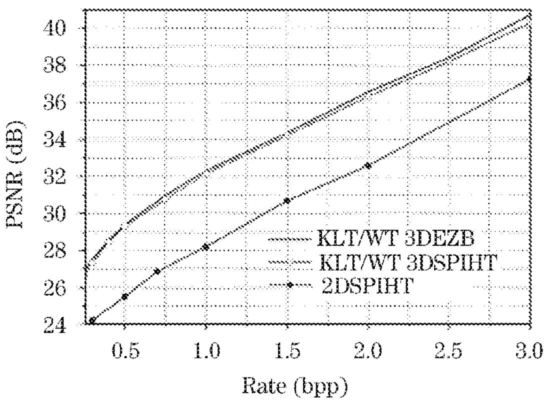

Fig. 3. Rate-distortion performance of KLT/WT-3DEZB, KLT/WT-3DSPIHT and 2DSPIHT for Basin image.

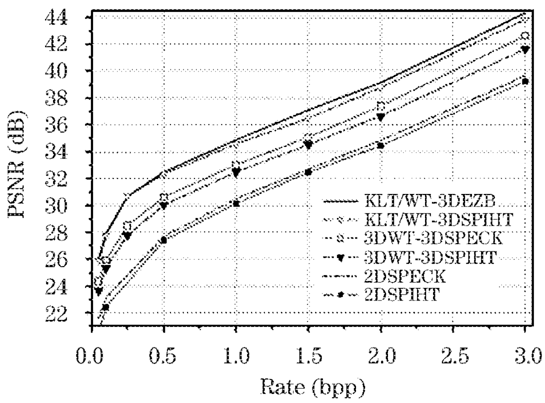

Fig. 4. Rate-distortion performance of various methods for Village image.

2D version. As we can see, the proposed algorithm KLT/WT-3DEZB beats 2DSPIHT and 2DSPECK by 4 dB at all rates. Besides, the curves shows that our KLT/WT-3DEZB PSNR is about 0.2 – 0.9 dB above KLT/WT-3DSPIHT, and performs better at medium to high bit rates. The rate-distortion curves in Fig. 4 show that KLT/WT-3DEZB outperforms 3DWT-3DSPECK and 3DWT-3DSPIHT in PSNR by 1.5 – 2 dB. Clearly, the winner in these results is KLT/WT-3DEZB.

In this paper, the two powerful transform techniques, KLT and WT, are efficiently combined in the proposed algorithm. Also, a new 3D image encoding technique 3DEZB is presented. Compared with the conventional techniques, KLT/WT-3DEZB exhibits a better performance and a lower complexity. Moreover, KLT/WT-3DEZB supports progressive transmission.

This work was supported by the Advanced Program of Military Communications Technology under Grant No. 410010302. J. Wu's e-mail address is luckydog73@sina.com, C. Wu's e-mail address is ckwu@ns2.xidian.edu.cn.

References

1. J. Vaisey, M. Barlaud, and M. Antonini, in *Proc. IEEE Int. Conf. Image Processing* 307 (1998).
2. J. A. Saghri, A. G. Tescher, and J. T. Reagan, in *Proc. Geoscience and Remote Sensing Symposium 1994* 313 (1994).
3. J. Lee, IEEE Trans. Image Processing **8,** 453 (1999).
4. T.-S. Kim, S.-J. Kim, B.-J. Kim, J.-W. Lee, S.-G. Kwon, and K.-I. Lee, in *Proc. ITC-CSCC2002* 204 (2002).
5. P. L. Dragotti, G. Poggi, and A. R. P. Ragozini, IEEE Trans. Geosci. Remote Sens. **38,** 416 (2000).
6. X. Tang, W. A. Pearlman, and J. W. Modestino, Proc. SPIE **5022,** 1037 (2003).
7. A. Islam and W. A. Pearlman, Proc. SPIE **3653,** 284 (1999).
8. S.-T. Hsiang and J. W. Woods, in *Proc. IEEE Int. Conf. on Circuits and Systems* **3,** 662 (2000).
9. C. Chrysafis, A. Said, A. Drukarev, A. Islam, and W. A. Pearlman, in *Proc. IEEE Int. Conf. on Acoust., Speech, and Sig. Proc.* **4,** 2035 (2000).
10. M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, IEEE Trans. Image Processing **1,** 205 (1992).
11. D. Taubman, IEEE Trans. Image Processing **9,** 1158 (2000).
12. B.-J. Kim and W. A. Pearlman, in *Proc. IEEE Data Compression Conf.* 251 (1997).
13. A. Said and W. A. Pearlman, IEEE Trans. Circuits Syst. Video Technol. **6,** 243 (1996).