

文章编号: 0253-2239(2008)01-0067-07

基于三维集合分裂嵌入式零块编码算法的 超光谱图像压缩

侯 颖^{1,2} 刘贵忠¹

(¹ 西安交通大学电子与信息学院, 陕西 西安 710049)
(² 西安科技大学通信与信息学院, 陕西 西安 710054)

摘要 基于超光谱图像的特点,提出了一种三维集合分裂嵌入式零块编码(3D SPEZBC)的超光谱图像压缩算法。该算法首先采用三维小波包变换有效地去除超光谱图像的空间和谱间相关性,然后对于所生成的每个二维子带利用基于集合分裂的方法进行零块编码,最后再采用基于上下文的自适应算术编码来进一步提高编码性能。实验结果表明,3D SPEZBC 算法具有与三维嵌入式零块编码(3D EZBC)算法相同的压缩编码性能,在各比特率下编码性能均明显优于三维集合分裂嵌入式块编码(3D SPECK)、三维等级树集合分裂(3D SPIHT)和非对称三维等级树集合分裂(AT-3D SPIHT)算法,并且略好于多分量 JPEG2000 编码(JPEG2000-MC)算法。此外,3D SPEZBC 编码算法不但可以提供较好的率失真性能,而且相对于 3D EZBC 编码算法可以节省大量的存储空间。

关键词 信息光学;超光谱图像压缩;三维小波变换;零块编码

中图分类号 TN919.81 **文献标识码** A

Three-Dimensional Set Partitioned Embedded Zero Block Coding Algorithm for Hyperspectral Image Compression

Hou Ying^{1,2} Liu Guizhong¹

¹ School of Electronics and Information Engineering, Xian Jiaotong University, Xian, Shaanxi 710049, China
² School of Communication and Information Engineering, Xian University of Science and Technology,
Xian, Shaanxi 710054, China

Abstract Based on the characteristics of hyperspectral images, three-dimensional set partitioned embedded zero-block coding (3D SPEZBC) algorithm for hyperspectral image compression is proposed. This algorithm adopts three-dimensional wavelet-packet transform to decorrelate hyperspectral image spectrally and spatially, and uses the quadtree partitioning method based on zero-block coding to process each of the two-dimensional subbands, and the context-based adaptive arithmetic coding scheme to further improve the coding performance. Experimental results show that the 3D SPEZBC algorithm has the same rate-distortion performances of three-dimensional embedded zero block coding (3D EZBC) algorithm, obviously outperforms three-dimensional set partitioned embedded block (3D SPECK) algorithm, three-dimensional set partitioning in hierarchical trees (3D SPIHT) algorithm and (AT-3D SPIHT) algorithm, and is slightly better than JPEG2000 multi-component (JPEG2000-MC) algorithm at different bit rate in the compression performances. Moreover, the 3D SPEZBC algorithm not only provides excellent rate-distortion performance, but also can save considerable memory requirement in comparison with the 3D EZBC algorithm.

Keywords information optics; hyperspectral image compression; three-dimensional wavelet transform; zero-block coding

收稿日期:2007-04-04;收到修改稿日期:2007-09-11

基金项目:国家自然科学基金(60572045)和国家教育部高校博士学科点专项科研基金(20050698033)资助课题。

作者简介:侯颖(1976—),女,陕西人,博士研究生,主要从事图像处理、小波分析、图像视频压缩编码等方面的研究。

E-mail: houying@mailst.xjtu.edu.cn

导师简介:刘贵忠(1962—),男,陕西人,教授,博士生导师,主要从事非平稳信号的分析与处理、音视频数据压缩、模式识别、反演等方面的研究。E-mail: liugz@mail.xjtu.edu.cn

1 引 言

超光谱成像技术可以对同一地物用数百个连续的窄光谱波段来同时成像,它能够获得被测目标丰富的空间和光谱信息,因此在地质分析、环境监测、军事侦察以及农林地形勘测等领域得到了广泛应用。但是其海量的数据也给存储、传输和处理带来极大的困难,尤其是无法在带宽有限的卫星信道上实现实时传输,有效的图像压缩编码成为必不可少和急需解决的关键技术。

超光谱图像可以看作是由二维空间和一维光谱维构成的三维立体图像。近年来,基于三维小波变换的超光谱图像压缩编码被许多学者广泛研究,并取得了很大的成功。例如,基于零树编码的三维等级树集合分裂算法(Three-dimensional set partitioning in hierarchical trees, 3D SPIHT)^[1~4],基于零块编码的三维集合分裂嵌入式块编码算法(Three-dimensional set partitioned embedded block, 3D SPECK)^[5,6],以及多分量 JPEG2000 编码算法(JPEG2000 multi-component, JPEG2000-MC)^[7~9]。2000年, Hsiang 和 Woods 提出的嵌入式零块编码算法(Embedded zero-block coding, EZBC)^[10~12]是当今编码性能较好的一种图像/视频压缩算法,该算法采用了两种优秀的编码方法——四叉树分裂零块编码与基于上下文的自适应算术编码。EZBC 不仅具有低计算复杂度和高压缩性能的优点,而且具有质量、分辨率和时间可分级编码的特性。同时, Hsiang 和 Woods 还将二维 EZBC 推广到三维视频编码,成功提出了运动补偿三维 EZBC (Motion-compensated three-dimensional EZBC, MC 3D-EZBC)^[11,12]。

虽然 EZBC 具有优秀的编码性能和良好的编码特性,但是由于它在编码过程中要求建立四叉树结构,因此该算法需要开辟较多的内存空间,这一缺点成为实现 3D EZBC 压缩大数据量超光谱图像的障碍。所以,考虑到超光谱图像的特点、EZBC 的优秀编码性能以及 SPECK 的低存储空间和快速编解码特性,本文提出了一种三维集合分裂嵌入式零块编码(Three-dimensional set partitioned embedded zero block coding, 3D SPEZBC)的超光谱图像压缩算法。该算法在 EZBC 和 SPECK 的基础上进行了改进,即去除了 3D EZBC 的四叉树结构,同时对于所生成的每个二维子带采用集合分裂方法进行零块编码。实验结果表明,3D SPEZBC 不但可以提供较好的率失真性能,而且相对于 3D EZBC 编码算法可

以节省大量的存储空间。

2 三维小波变换

由于小波变换在时频域内具有良好的局部化特性,对图像进行多分辨率分解后仍可以保持其在各个分辨率下的精细结构,小波变换后系数在时频域内能量高度集中,从而能较好地去除像素间的相关性,这一系列的优秀特性使其在图像/视频压缩编码领域获得了广泛的应用,相关研究也取得了很大进展。超光谱图像可以看作是由二维空间域和一维光谱域构成的三维立体图像,它不仅存在空间的相关性,还具有很强的谱间相关性。经过许多学者研究^[1,8],超光谱图像在行方向、列方向和谱方向具有一种非对称的统计特性,其谱方向上的相关性比行、列方向上的相关性更强,因此三维小波包变换能更有效地去除超光谱图像的空间和谱间相关性。3D SPEZBC 在编码之初,对超光谱图像先进行三维小波包变换(The three-dimensional wavelet-packet transform, 3D WPT)。该三维小波包变换首先沿着超光谱图像的谱方向进行多级一维小波变换,然后再对每一幅变换后的图像进行多级二维小波变换,分解之后就产生了一个类似金字塔形状的三维非对称子带结构,如图 1 所示。若对包含 F 个谱带的超光谱图像进行 I_{spatial} 级空间方向和 L_{spatial} 级谱方向的三维小波包变换,则会产生 $B = (L_{\text{spatial}} \times 3 + 1) \times F$ 个大小不同的二维子带,图 1 给出了一个 8 谱带图像进行两级空间和谱间的 3D WPT 后,具有 56 个二维子带的结构图。

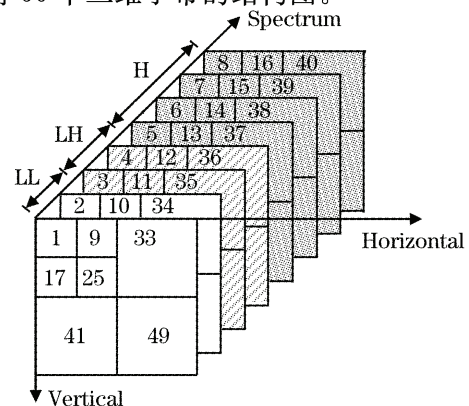


图 1 3D SPEZBC 算法两级空间和谱间的三维小波包变换结构(每个二维子带左上角的数字表示初始化不重要集合链表的顺序)。

Fig.1 Three-dimensional wavelet packet transform structure of two spatial and spectral levels for 3D SPEZBC (The numbers on the front upper left corners for the respective subbands indicate the initialization order of LIS)

3 3D SPEZBC 编码算法

3.1 3D SPEZBC 算法编码思想

EZBC^[12] 和 SPECK^[13] 作为两个成功的零块编码算法具有完全嵌入性、渐进传输性、高压缩性能和低计算复杂度等特点。零块编码算法充分利用了小波变换后子带间等级树结构及子带内的能量集中特性从而进行零块分裂编码。EZBC 和 SPECK 分别采用了两种不同的分裂机理：一种是 EZBC 的基于二叉树表示结构的二叉树分裂方法，另一种是 SPECK 的基于集合表示结构的二叉树分裂方法。在进行二叉树分裂编码之前，EZBC 编码器需要为三维小波变换后所生成的每一个二维子带建立二叉树表示结构。这种二叉树结构可以提供快速二叉树分裂，但为此却需要占用较多的内存空间。特别是对于三维图像（例如，超光谱图像，三维医学图像等）进行 3D EZBC 压缩时，内存使用量的增加尤为显著。因此，考虑到超光谱图像的特点、EZBC 的优秀编码性能以及 SPECK 的低存储空间和快速编解码特性，本文提出了一种改进的 3D EZBC 超光谱图像压缩编码算法——3D SPEZBC 算法。该算法在 EZBC 和 SPECK 的基础上进行了改进，即去除了 3D EZBC 编码算法的二叉树表示结构，对于所生成的每个二维子带采用基于集合

表示结构的二叉树分裂零块编码。

3.2 3D SPEZBC 编码算法

3D SPEZBC 算法的编码过程如图 2 所示。在编码之初，3D SPEZBC 算法先对超光谱图像进行三维小波包变换，从而会产生多个大小不同的二维子带（如图 1 所示），将每个独立的二维子带定义为一个初始的集合 S 。随后，3D SPEZBC 算法再采用基于集合表示结构的二叉树分裂零块编码方法对每一个集合 S 进行编码，如图 3(a) 所示。分裂过程中生成的所有集合依照其尺寸递增的顺序进行处理，例如尺寸为 1×1 的集合 S （即单像素点）先编码，其次是尺寸为 2×2 的集合 S 再编码，依此类推。当编码尺寸为 $M \times M$ 的集合 S 时，小波变换后所生成的所有子带按照其初始化顺序（如图 1 所示）依次进行处理。如果集合 S 中包含重要系数，则将其分裂为尺寸大小基本相等的四个子集合 $O(S)$ ，随后又将每个子集合 $O(S)$ 看作是新的集合 S 进一步测试其重要性，若重要则重复进行集合分裂处理，反之则不进行处理。整个分裂过程循环执行，直到将最初的集合 S 中每一个重要系数查找到为止。最后，3D SPEZBC 算法采用了与 3D EZBC 相同的熵编码方法来进一步提高编码性能，即基于上下文的自适应算术编码算法。

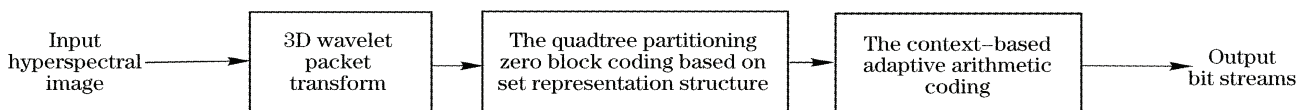


图 2 3D SPEZBC 编码系统框图

Fig. 2 Block diagram of the 3D SPEZBC coding system

3D SPEZBC 算法采用两类链表，即

1) LIS(List of insignificant sets)：不重要集合链表。

2) LSP(List of significant pixels)：重要像素链表。

超光谱图像进行三维小波包变换后，会产生 B 个大小不同的二维子带，3D SPEZBC 将每个独立的二维子带作为一个初始集合 S ，并为其分配一对 LIS_b （第 b 个子带的不重要集合链表）和 LSP_b （第 b 个子带的重要像素链表），因此这里共有 B 对 LIS_b 和 LSP_b 链表， $b=0, 1, \dots, B-1$ 。同时，根据图 1 每个子带上所标记的数字来初始化 LIS 和 LSP 链表的顺序。

一个大小为 $M \times M$ 的集合 S ，其集合分裂级数 L 被定义为

$$L = \text{lb } M,$$

这里，集合分裂级数 L 表示当前集合 S 被循环分裂为

像素级集合所需要的分裂深度，它与 EZBC 中的二叉树深度起着相同的作用。同时， L_b 表示初始化集合 S （即第 b 个二维子带）的集合分裂级数，这里 $b=0, 1, \dots, B-1$ 。此外，定义 L_{\max} 为编码过程中所有集合 S 的最大集合分裂级数。例如，对 $512 \times 512 \times 224$ 的超光谱图像进行 K 级（ $K \geq 1$ ）三维小波包变换后，编码过程中集合 S 的最大尺寸为 256×256 ，因此最大集合分裂级数 $L_{\max} = \log_2 256 = 8$ 。注意，当 $L=0$ 时表示当前集合为像素级集合，也就是说当前集合 S 的尺寸大小为 1×1 的单像素点。图 3(b) 给出了集合分裂级数与集合尺寸之间的关系示意图。

集合 S 关于阈值 2^n 的重要性测试函数定义为

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

式中 $c_{i,j,k}$ 表示 (i, j, k) 坐标位置处的小波系数。

3D SPEZBC 算法的编码步骤如下：

1) 初始化

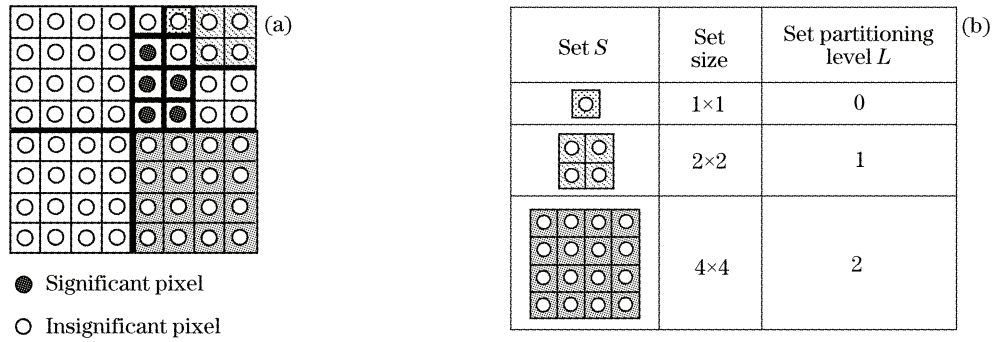


图 3 (a)集合 S 的分裂过程, (b)集合分裂级数与集合尺寸之间的关系示意图

Fig. 3 (a) Set partitioning procedure of set S, (b) the relationship between set partitioning level and set size

$$n = \lceil \lg \{ \max_{(i,j,k)} |c(i,j,k)| \} \rceil,$$

$$\text{for } b = 0; B-1$$

$$\text{LIS}_b[l] = \begin{cases} \{\text{集合 } S, \text{即第 } b \text{ 个二维子带}\}, & l = L_b \\ \Phi, & \text{other} \end{cases}$$

$$\text{LSP}_b = \Phi.$$

2) 分类通道

$$\text{for } l = 0; L_{\max}$$

$$\text{for } b = 0; B-1$$

$$\text{CodeLIS}(b, l);$$

$$\text{CodeLIS}(b, l)$$

{对于每一个集合 $S \in \text{LIS}_b[l]$

输出 $\Gamma_n(S)$, 并进行基于上下文的自适应算术编码;

如果 $\Gamma_n = 1$

如果 $l = 0$, 即集合 S 是一个单像素点

输出集合 S 的符号, 并进行基于上下文的自适应算术编码; 同时将集合 S 移到 LSP_b 链表;

否则, 进行 $\text{CodeSubSets}(b, l, S)$ 编码, 并将集合 S 移出 $\text{LIS}_b[l]$ 链表;

否则, 将集合 S 保留在 $\text{LIS}_b[l]$ 链表中。}

$$\text{CodeSubSets}(b, l, S)$$

{将集合 S 分裂为四个尺寸大小基本相等的子集合 $O(S)$ 。}

对每一个集合 $O(S)$

输出 $\Gamma_n(O(S))$, 并进行基于上下文的自适应算术编码;

如果 $\Gamma_n(O(S)) = 1$

如果 $l = 1$, 即集合 $O(S)$ 是一个单像素点

输出集合 $O(S)$ 的符号, 并进行基于上下文的自适应算术编码; 同时将集合 $O(S)$ 移到 LSP_b 链表;

否则, 进行 $\text{CodeSubSets}(b, l-1, O(S))$

编码。

否则, 将集合 $O(S)$ 加入到 $\text{LIS}_b[l-1]$ 链表中。}

3) 幅值细化通道

$$\text{for } b = 0; B-1$$

$$\text{CodeLSP}(b);$$

$$\text{CodeLSP}(b)$$

{对每一个像素 $(i, j, k) \in \text{LSP}_b$ (除过最后一次分类通道加入的像素) 输出 $|c(i, j, k)|$ 的第 n 个最高有效位, 并进行基于上下文的自适应算术编码。}

4) 量化步骤

n 减 1, 并转到步骤 2) 重新执行。

3.3 熵编码

为了进一步提高压缩性能, 3D SPEZBC 算法采用与 3D EZBC 相同的熵编码算法, 从而对重要性比特流、符号比特流和幅值比特流进行基于上下文的自适应算术编码。虽然 3D SPEZBC 算法去除了二叉树表示结构, 但是该算法依然可以利用集合分裂过程来建立起与 3D EZBC 类似的上下文模型。3D EZBC 为每一个不同子带内及二叉树深度上的节点建立了上下文模型。然而, 3D SPEZBC 算法却是为每个子带内的集合建立了一个等级树结构的上下文模型, 其中上下文模型的最大级数为该子带所对应的集合分裂级数。S 集合的上下文就记录于它在子带内其集合分裂级上上下文模型的一个节点处, 如图 4 所示。例如, 如果一个集合分裂等级为 L 的 S 集合被测试为重要集合, 则它会被分裂为四个大小相等的子集合 $O(S)$, 其子集合的集合分裂等级为 $L-1$ 。这里, 集合 S 的上下文记录在它所在子带内 L 级上下文模型的对应节点位置, 而子集合 $O(S)$ 的

上下文会记录在该子带内 $L-1$ 级上下文模型的对

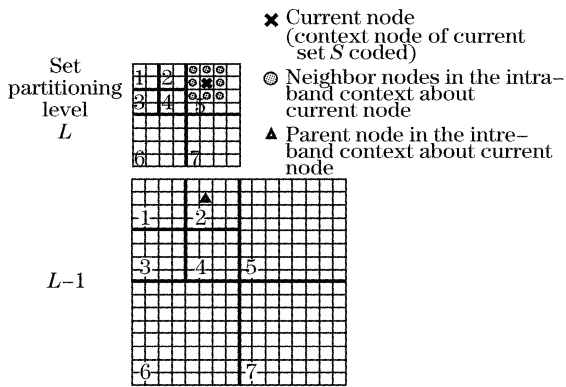


图 4 3D SPEZBC 算法中 S 集合重要性熵编码的上下文模型

Fig. 4 The context models of a set S for the 3D SPEZBC entropy coding of significance test

3D SPEZBC 算法重要性比特流熵编码的上下文如图 4 所示,它主要记录了所有 S 集合的重要性状态。同 3D EZBC 一样,3D SPEZBC 算法中集合重要性熵编码算法也利用了两种统计相关性,即(1)子带内各集合间的带内相关性;(2)各集合间跨尺度的带间相关性。一幅图像经过两级小波变换后所形成的各子带分布顺序如图 4 所示。其中,在相同的集合分裂等级 L 上,第 5 子带中当前编码集合 S 的上下文节点“ \times ”与它的 8 个相邻节点“ \circ ”构成了带内上下文关系;在集合分裂等级 L 上第 5 子带中的当前编码集合 S 的上下文节点“ \times ”与在集合分裂等级 $L-1$ 上第 2 子带中的父节点“ \blacktriangle ”构成了跨尺

应节点位置。

度的带间上下文关系。此外,基于上下文的符号比特流编码采用 EBCOT (Embedded block coding with optimized truncation)和 JPEG2000(联合图像专家组 2000 标准)的编码模式,即当前像素点的符号值是根据其子带内 8 个相邻节点的重要性和符号值预测获得。因此,3D SPEZBC 算法能够在没有二叉树结构的情况下,利用上述两种上下文关系采用与 3D EZBC 相同的基于上下文的自适应算术编码,其中上下文结构模型和查询列表的详细内容可参考文献[12]中 54~66 页以及附录 B。

当前几种优秀的编码算法,3D SPECK、3D SPIHT 和 AT-3D SPIHT 编码算法均采用的是基于简单上下文关系的自适应算术编码。然而,这种简单的上下文关系没有充分地利用编码过程中的上下文。由于 3D SPEZBC 算法采用的是基于 3D EZBC 精心设计的上下文模型的自适应算术编码,因此使得它在编码性能上要明显优于上述三种算法。此外,JPEG2000 也采用了基于上下文的自适应算术编码,但是由于本文算法采用了无需率失真优化的基于上下文自适应算术编码,所以它相对于 JPEG2000 算法具有较低的计算复杂度。

4 实验结果与分析

本文实验采用四幅 16 bit 的 AVIRIS(Airborne visible/infrared imaging spectrometer)超光谱反射

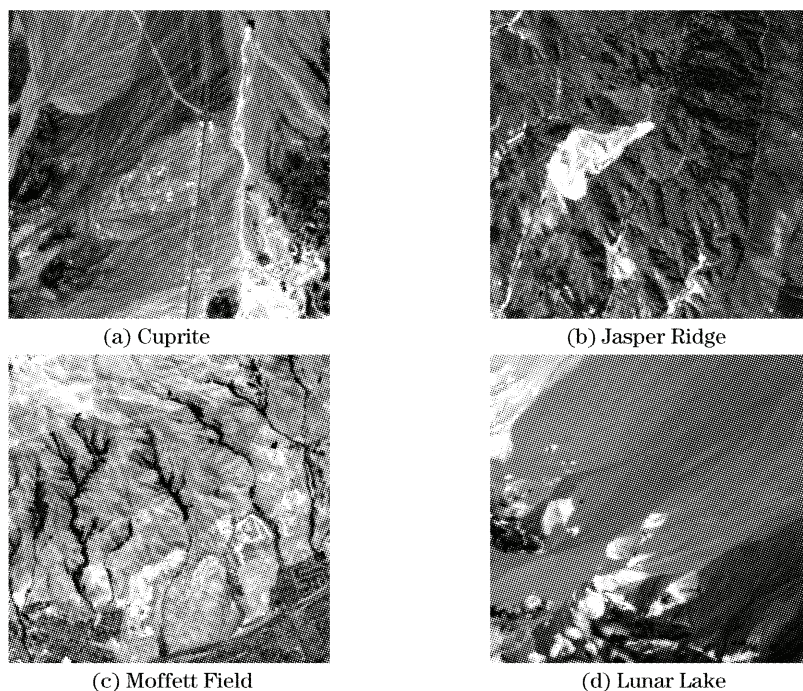


图 5 四幅超光谱图像在第 127 谱带的原始图像

Fig. 5 Original images of the 127th band of four experimental hyperspectral images

图像^[14]进行测试,即 Cuprite scene 1、Jasper Ridge scene 1、Moffett Field scene 1 以及 Lunar Lake scene 1。每幅图像由 224 个连续谱带构成,每个谱带图像的空间分辨率为 $614 \times 512 \times 16$ bit。在实验中,提取每一波段图像的 256×256 的左下角区域,最终形成 $256 \times 256 \times 224 \times 16$ bit 的四幅三维测试数据,如图 5 所示。

4.1 编码性能比较

表 1 给出了 3D SPEZBC、3D SPECK、3D SPIHT、AT-3D SPIHT 和 JPEG2000-MC 五种压缩算法在各比特率下 (Bits per pixel per band, bpppb) 的信噪比 (Signal-to-noise ratio, SNR) 比较结果,其中所有算法均采用双正交 9/7 滤波器进行 4 级空间和 4 级谱间的三维小波包变换。从实验结果可以看出,对于四幅超光谱图像 3D SPEZBC 算法在各比特率下均明显优于 3D SPECK、3D SPIHT 和 AT-3D SPIHT 算法。例如,对于四幅实验图像在 1 bpppb 比特率下,3D SPEZBC 算法分别比 3D SPECK 率失真性能平均提高 1.39 dB,比 3D SPIHT 算法率失真性能平均提高 1.72 dB,比 AT-3D SPIHT 算法率失真性能

表 1 信噪比性能比较 (dB)

Table 1 Signal-to-noise ratio performance comparison (in dB)

Coding methods	Bitr rate /bpppb				
	0.1	0.5	1.0	1.5	2.0
Cuprite					
3D SPEZBC	21.24	33.22	41.82	46.72	50.36
3D SPECK	20.22	31.64	40.10	45.48	49.10
3D SPIHT	20.16	31.58	39.85	45.38	48.85
AT-3D SPIHT	20.64	32.41	41.23	46.13	49.78
JPEG2000-MC	21.22	33.13	41.70	46.64	50.02
Jasper Ridge					
3D SPEZBC	22.82	35.10	41.43	45.25	48.14
3D SPECK	21.81	33.71	40.09	44.07	47.00
3D SPIHT	21.47	33.49	39.83	43.94	46.95
AT-3D SPIHT	22.45	34.32	40.67	44.60	47.47
JPEG2000-MC	22.78	34.92	41.15	44.99	47.95
Moffett Field					
3D SPEZBC	18.76	33.60	41.84	46.38	49.77
3D SPECK	17.14	31.94	40.55	45.20	48.62
3D SPIHT	17.77	31.66	40.15	44.98	48.22
AT-3D SPIHT	18.18	32.60	41.10	45.66	49.17
JPEG2000-MC	18.71	33.48	41.73	46.21	49.48
Lunar Lake					
3D SPEZBC	26.13	40.17	46.86	50.42	53.07
3D SPECK	23.97	38.73	45.64	49.33	51.98
3D SPIHT	24.30	38.41	45.21	49.08	51.83
AT-3D SPIHT	25.13	39.32	46.16	49.84	52.39
JPEG2000-MC	25.87	40.01	46.62	50.23	52.84

平均提高 0.70 dB。此外,除了“Lunar Lake”图像外,在较低比特率下(例如 0.5 bpppb 以下),3D SPEZBC 算法的编码性能与 JPEG2000-MC 算法的结果相当。但随着比特率的增长, JPEG2000-MC 算法的率失真性能逐渐略低于 3D SPEZBC 算法。

4.2 3D SPEZBC 算法与 3D EZBC 性能比较

3D EZBC 的基于四叉树表示结构的四叉树分裂方法与 3D SPEZBC 算法的基于集合表示结构的四叉树分裂方法的主要区别在于对零块区域结构的表示方式和分裂测试机制的不同,但是它们分裂和编码后的结果却是相同的。因此,3D SPEZBC 算法和 3D EZBC 算法虽然采用了不同的四叉树分裂方法,但是它们具有相同的压缩编码性能。这一结论我们通过实验也得到了充分的验证。

此外,由于 3D SPEZBC 算法去除了 3D EZBC 中的四叉树结构,所以 3D SPEZBC 算法相对 3D EZBC 可以节省大量的存储空间。在 3D EZBC 中,对于一个大小为 $M \times M$ 的二维子带,其四叉树深度 D 等于 $\lg M$,若小波变换系数采用 float 类型进行存储(即 4 字节),则建立四叉树表示结构需要分配 $\sum_{i=0}^D \left(\frac{1}{4}\right)^i \times M^2 \times 4$ 字节的内存空间。然而,由于本文的 3D SPEZBC 算法无需建立四叉树结构,因此将节省 $\sum_{i=1}^D \left(\frac{1}{4}\right)^i \times M^2 \times 4$ 字节的存储空间(其中 $i = 0$ 的零级四叉树用于存储小波变换后的系数,不能被去除)。故本文算法对于 $512 \times 512 \times 224$ 大小的超光谱图像进行压缩编码时,大致可以节省 75.52 兆字节。

4.3 分类性能比较

为了充分反映压缩编码算法对于超光谱重构图像造成应用性能上的影响,本文采用迭代自组织数据分析 (ISODATA) 的非监督分类方法进一步对重构图像进行分析,其中表 2 给出了上述五种压缩算法在各比特率下的分类精度比较结果。在较低比特率下(例如,0.4 bpppb 以下),3D SPEZBC 算法压缩后重构图像的分类精度要明显优于 3D SPECK,3D SPIHT 算法和 AT-3D SPIHT 算法。例如,对于四幅实验图像在 0.1 bpppb 比特率下,3D SPEZBC 算法分别比 3D SPECK 分类性能平均提高 4.77%,比 3D SPIHT 算法分类性能平均提高 7.56%,比 AT-3D SPIHT 算法分类性能平均提高 3.11%。此外,除了低比特率下的“Cuprite”图像外,3D SPEZBC 算法的分类性能与 JPEG2000-MC

算法的结果非常接近。随着比特率的增长,例如超过 0.7 bpppb 以上,上述五种算法的分类精度绝大部分高于 99%。

表 2 迭代自组织数据分析(ISODATA)分类算法的分类精度比较(%)

Table 2 Overall classification accuracy comparison (in %) based on ISODATA

Bit rate /bpppb	Coding methods	Hyperspectral images			
		Cuprite	Jasper Ridge	Moffett Field	Lunar Lake
0.1	3D SPEZBC	86.48	86.69	79.79	86.84
	3D SPECK	79.99	84.43	71.92	84.38
	3D SPIHT	79.65	83.38	64.16	82.39
	AT-3D SPIHT	80.71	85.44	75.54	85.68
	JPEG2000-MC	82.93	86.75	79.48	85.71
0.5	3D SPEZBC	98.93	98.58	98.18	98.71
	3D SPECK	98.61	98.41	95.63	98.15
	3D SPIHT	98.59	98.34	87.90	97.73
	AT-3D SPIHT	98.71	98.47	96.85	98.43
	JPEG2000-MC	98.88	98.83	97.67	97.78
1.0	3D SPEZBC	99.69	99.62	99.09	99.47
	3D SPECK	99.66	99.60	98.19	99.31
	3D SPIHT	99.64	99.55	97.82	99.28
	AT-3D SPIHT	99.67	99.60	98.57	99.40
	JPEG2000-MC	99.69	99.68	99.12	99.35

5 结 论

EZBC 算法是当今编码性能较好的一种图像/视频压缩算法,由于它在编码过程中需要建立二叉树结构,因此该算法要求开辟较高的内存空间。本文基于 EZBC 和 SPECK 的特点,提出了一种 3D SPEZBC 编码算法的超光谱图像压缩算法。3D SPEZBC 不但可以提供较好的率失真性能,而且相对于 3D EZBC 可以节省大量的存储空间。

参 考 文 献

- Xiaoli Tang, Sungdae Cho, William A. Pearlman. 3D set partitioning coding methods in hyperspectral image compression [C]. In *Proc. IEEE International Conference Image Processing*, 2003, **2**: 239~242
- Zhou Youxi, Li Yunsong, Wu Chengke. Environment satellite multispectral images compression algorithm [J]. *Acta Optica Sinica*, 2006, **26**(3): 336~340
周有喜,李云松,吴成柯. 环境卫星多光谱图像压缩算法[J]. *光学学报*, 2006, **26**(3): 336~340
- Ma Jing, Wu Chengke, Cheng Dong *et al.*. Compression of interference spectral image based on weighted rate-distortion optimization for SPIHT [J]. *Acta Optica Sinica*, 2007, **27**(3): 430~436
马 静,吴成柯,陈 东等. 基于加权率失真优化 SPIHT 算法的干涉多光谱图像压缩[J]. *光学学报*, 2007, **27**(3): 430~436
- Jing Huang, Rihong Zhu, Jianxin Li *et al.*. Hyperspectral image compression using three dimensional significance tree splitting [J]. *Chin. Opt. Lett.*, 2007, **5**(7): 393~396
- Xiaoli Tang, William A. Pearlman, James W. Modestino. Hyperspectral image compression using three-dimensional wavelet coding [J]. *SPIE/IS&T Electronic Imaging*, 2003, **5022**: 1037~1047
- Wu Jiaji, Wu Zhensen, Wu Chengke. Hyperspectral image compression using three-dimensional wavelet embedded zeroblock coding [J]. *J. Software*, 2007, **18**(2): 461~468
吴家骥,吴振森,吴成柯. 超光谱图像的三维小波嵌入零块压缩编码[J]. *软件学报*, 2007, **18**(2): 461~468
- ISO/IEC 15444-2, Information Technology—JPEG-2000 Image Coding System—Part 2: Extensions[S]. 2004. 1~332
- Barbara Penna, Tammam Tillo, Enrico Magli *et al.*. Progressive 3-D coding of hyperspectral images based on JPEG 2000 [J]. *IEEE Geoscience and Remote Sensing Letters*, 2006, **3**(1): 125~129
- Xiao Jiang, Wu Chengke, Li Yunsong *et al.*. Compression of the multispectral image by the three-dimensional EBCOT coding algorithm [J]. *J. Xidian University*, 2005, **32**(4): 550~554
肖 江,吴成柯,李云松等. 多光谱图像的 3D EBCOT 压缩编码算法[J]. *西安电子科技大学学报*, 2005, **32**(4): 550~554
- Shih-Ta Hsiang, J. W. Woods. Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling [C]. *Proc. IEEE International Symposium on Circuits and systems*, 2000, **3**: 662~665
- Shih-Ta Hsiang, J. W. Woods. Embedded video coding using motion compensated 3-D subband/wavelet filter bank [C]. *Signal Processing: Image Communications*, 2001, **16**: 705~724
- Shih-Ta Hsiang. *Highly Scalable Subband/Wavelet Image and Video Coding* [D]. Ph. D Dissertation, Department of Electrical, Computer, and Systems Engineering, Rensselaer Polytechnic Institute, Troy, 2002. 1~317
- A. Islam, W. A. Pearlman. An embedded and efficient low-complexity hierarchical image coder [C]. *Proc. SPIE*, 1999, **3653**: 294~305
- <http://aviris.jpl.nasa.gov/html/aviris.overview.html>