

一种基于低复杂度嵌入零块编码的遥感图像压缩编码方法

吴家骥 吴成柯

(西安电子科技大学 ISN 国家重点实验室 西安 710071)

摘要:该文针对遥感图像的数据特点,提出了一种新的遥感图像编码方法。它基于一种改进的小波变换嵌入零块编码算法。新算法中改进了零块编码中二叉树分裂算法(quadtree),并设计了新的链表生成和不重要集合排序策略。通过这些方法的改进,不仅提高了图像编码性能,同时还大大提高了运算效率。实验表明该文阐述的算法具有很低的复杂度和高的压缩率,PSNR 和计算速度均超过 SPIHT 和 SPECK。在 1bpp 下,该文方法的 PSNR 比 SPIHT 提高了 0.3dB 以上,计算速度比 SPIHT 提高了 35%。

关键词:遥感图像编码,嵌入编码,零块编码,二叉树,小波变换

中图分类号: TP751 **文献标识码:** A **文章编号:** 1009-5896(2005)03-0445-04

A Remote Sensing Image Compression Algorithm Based on Low-Complexity Embedded Zeroblock Coder

Wu Jia-ji Wu Cheng-ke

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

Abstract In this paper, a new coding approach to remote sensing images is proposed according to the statistical characteristic of remote sensing images data. The approach is based on an improved embedded, zeroblock-based, image wavelet transform coding algorithm. A modified quadtree algorithm and a new ordering's strategy of List of Insignificant Sets (LIS) are proposed in new algorithm. By improving in this proposal, the performance of the embedded wavelet image coder is increased and a higher computing efficiency is obtained. The proposed algorithm exhibits a lower complexity, a more efficient compression ratio and a higher PSNR than SPIHT and SPECK. Experimental results show that the proposed method increases the PSNR by about 0.3dB over SPIHT, and beats SPIHT by 35% at the CPU time.

Key words Remote sensing image coding, Embedded coding, Zeroblock coding, Quadtree, Wavelet transform

1 引言

随着遥感技术的迅速发展,遥感数据量日益庞大,有限的信道容量与传输大量遥感数据的需求之间的矛盾日益突出。数据编码压缩技术作为解决这一问题的有效途径,在遥感领域越来越受到重视。与一般静止图像相比,遥感图像具有局部相关性较弱、纹理复杂、高熵值和低冗余度的特点。目前普遍采用基于小波嵌入图像编码方法对遥感图像进行高效地压缩编码。如嵌入式零树小波编码(EZW)算法^[1]和多级树集合分裂(SPIHT)算法^[2]能够大幅提高图像的压缩率。嵌入编码(embedded coding)就是编码器将待编码的比特流按重要性的不同进行排序,根据目标码率或失真度大小要求随时结束编码。同样,对于给定码流解码器也能够随时结束解码,并可以得到相应码流截断处的目标码率的恢复图像。嵌入编码中首先传输的是最重要的信息,也就是幅值最大的变换系数的信息。依次以二进制的码流输出,从而实现

图像的渐进传输。这种以重要性排序和分级量化的思想被许多编码算法所采用。

在 SPIHT 算法中,采用的是零树编码,但它的复杂度较高,而且鲁棒性较差,因此人们提出了基于零块的编码方法。不像 SPIHT 编码依靠树中元素的相关性,零块编码依靠小波分解中子带系数的相关性。如集合分裂嵌入块编码(SPECK)算法^[3]利用零块编码的思想不仅降低了编码复杂度,而且在同码率下的 PSNR 与 SPIHT 的结果相近,同时抗误码的能力也比基于零树的 SPIHT 算法有所提高。本文针对遥感图像的特点,改进了零块编码中的二叉树分裂算法并重新设计了零块的存储链表结构和排序策略,不仅大大缩短了计算时间。而且压缩比有了进一步的提高。

2 零块编码(Zeroblock coding)的改进

零块编码具有非常好的图像编码性能和较低的复杂度。

它采用简单直观和有效的块分裂方法：只要一个给定的集合被测试是重要的，那么它就会分裂成 4 个相等的子块，这就是四叉树分裂方法。四叉树算法被广泛运用在图像视频编码领域^[4,5]。近来，四叉树技术也被运用到一些比特面编码的算法中，如简单小波嵌入集合分裂 (SWEET) 算法^[6]、SPECK^[3] 和嵌入零块 (EZBC) 算法^[7]等。四叉树分裂的过程实际上就是把一个正方形的块分裂成为 4 块，形成 4 个较小的正方形块。在运用到内嵌零块比特面编码中时，对测试重要的块输出 1，并进行四叉树分解，对分裂的四个子块再分别测试，重要的输出 1，不重要的输出 0。

四叉树分裂是个反复迭代的过程，利用四叉树分裂能够快速获得子块中重要元素的位置。也就是说在嵌入零块比特面编码中对不重要的块或元素只要用一个“0”比特就可以表示，对于那些重要的用“1”表示的块再进行四叉树分裂，直到分裂到只有 4 个元素为止。

我们可以发现，在四叉树算法中，对于一个重要的块，它的四叉树子孙中的 4 个小块中，至少有 1 个是重要的，即每个子块有 0.25 的几率是重要的。换句话说，如果有 3 个子块不重要，那么显然剩下的那个子块一定重要。如果我们输出 3 个“0”比特表示不重要的子块，则剩下的那个重要的子块不用输出“1”，在解码时我们也知道它是重要的。利用这个思想，我们改进了四叉树算法，减少了一些比特的输出，以提高编码效率。通过对大量的图像测试发现，如果 4 个子块中仅有 1 个子块是重要时，那么右下脚的子块是重要的概率最大，这样通常采用“Z”形顺序对 4 个子块进行编码的效果会更好些。图 1 中显示了四叉树的分裂步骤及改进后的四叉树分裂编码的输出。

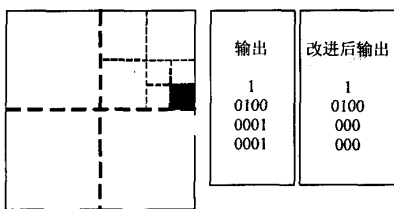


图 1 一个 8×8 块的四叉树分裂和改进后编码输出
黑色的块为重要，输出比特“1”表示块将进一步分裂

3 不重要系数和零块优化排序的改进

原始图像经过多级小波变换后，生成相关性很低小波系数。在零块编码中，零块大小可能包含整个子带也可能只包含一个小波系数，这些不重要的零块被存储在不重要集合链表 (LIS) 中。LIS 中的零块的排列顺序对下一比特面的测试极为重要，极大地影响编码的性能。对零块的优化排序实际是对比特面内的率失真的优化。通常来说，如果一个零块周围的系数已经是重要的，那么下一次测试中，这个零块可能

是重要的概率就很高。我们试验了一种基于上下文模型的 LIS 排序方法^[8]，把那些周围有重要系数的零块排在 LIS 的前面，先进行测试，但发现对基于零块的编码算法效果不如零树的效果好。我们这里采用 SPECK 中的排序算法。在 SPECK 中，只需每次对零块按块的大小升序排列，即块小的排在前面（只包含一个系数的块排在 LIS 的最前面），块大的排在后面，也就是说，零块越小，则此块周围的重要系数就越多，那么下次测试时它就越有可能是重要的。试验发现对于尺寸相同的块按 FIFO 的原则排序，编码效果最优。

但在 SPECK 算法中，随着比特面 n 的减少，LIS 中包含了大量的只含有一个系数的零块，这些不重要的系数在参与排序过程中使计算量加大。我们参考 SPIHT 算法，把这些只含有一个系数的零块按照 FIFO 另存到不重要元素链表 (LIP) 中，这样每次参加排序的零块数目就大大减少，其中 LIP 中的每一个元素都对应着一个测试后不重要的小波系数的坐标。

另外在对 LIS (或 LIP) 链表中的集合 S 进行测试其是否重要时，如果 S 重要，且 $S \in LIS$ ，则从 LIS 中移出 S ；如果 S 不重要，且 $S \notin LIS$ ，则把 S 加入到 LIS 中。这两个过程需要把 LIS (或 LIP) 遍历搜索一遍。我们设计了新的非常简单的方法，可以省略这两个过程：在每次测试 LIS 时，我们先生成一个空的链表 LIS_C，从链表中取出要测试的 S ，无论它是否重要都从 LIS 中删除，如果 S 不重要，则把它加入到 LIS_C 中，对重要的 S 则进行四叉树迭代分裂，把四叉树生成的不重要的 S 也加入到 LIS_C 中。全部测试完后，LIS 链表则为空，再把 LIS_C 中的内容指向 LIS，把 LIS_C 清空。LIP 的处理也类似。通过增加 LIP 链表和取消对链表的遍历搜索，使计算时间大大减少的同时，没有增加任何内存开销。

在实验中我们还发现，对于细小目标较多的遥感图像来说，经过小波变换后的能量簇变化很快，这样在编码的过程中，零块分裂很快，对 PSNR 贡献较大的绝大多数未被测试的有用信息都集中在从 2 到 64 大小的块中，而再大些的块分裂后生成的重要元素很少，对 PSNR 的贡献也非常少。因此为了进一步优化比特面内的失真率，我们只排序 LIS 中大小为 2 至 64 的块，然后就进行重要元素链表 (LSP) 的精细量化 (refinement)，在最后才对 LIS 中剩下的零块判断其是否重要。这样不仅使排序的零块减少，提高了计算速度，也使比特面内的率失真更为优化。

4 具体算法实现

本文算法没有采用 SPECK 中复杂的集合分裂方法，而采用类似 EZBC 的方法，以各小波子带作为初始零块，详细算法如下：

重要性测试函数定义为

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

式中 $c_{i,j}$ 为小波系数。

(1) 初始化

(a) 输出 $n_{\max} = \left\lfloor \log_2 \left(\max_{(i,j)} \{ |c_{i,j}| \} \right) \right\rfloor$;

(b) 把各小波子带加到 LIS 中, 并设 LIP, LSP 为空。

(2) 排序测试不重要的元素和元素集合

(a) 把 LIS 中的 S 按照 S 集合的大小, 升序排列, 注意: 只排序大小从 2 到 64 的块;

(b) 设 LIP_C 和 LIS_C 为空;

(c) 每次移出 LIP 中的第一个元素 S , 执行过程 ProcessS(S), 直到 LIP 为空;

(d) 每次移出 LIS 中的第一个集合 S , 执行过程 ProcessS(S), 直到 LIS 中没有任何块的尺寸小于等于 64。

(3) 细化过程

对 LSP 中的元素, 除了那些在最近一次排序过程中包含进来的元素 (即具有相同的 n 值), 输出 $|c_{i,j}|$ 的第 n 层重要位的值。

(4) 进一步测试不重要的元素集合

(a) 每次移出 LIS 中的第一个集合 S , 执行 ProcessS(S), 直到 LIS 为空;

(b) 把 LIP_C 的内容指向给 LIP, 把 LIS_C 的内容指向给 LIS。

(5) 量化步长更新: n 减 1, 转 (2) 执行。

Procedure ProcessS (S)

(1) 输出 $\Gamma_n(S)$

(2) 假如 $\Gamma_n(S) = 1$

(a) 假如 S 是一个元素, 则输出 S 的符号并增加 S 到 LSP;

(b) 否则 QuadtreeS (S) 。

(3) 否则, 假如 $\Gamma_n(S) \neq 1$

(a) 假如 S 的尺寸为 1, 增加 S 到 LIP_C 中;

(b) 否则, 增加 S 到 LIS_C 中。

(4) S_i 返回

Procedure QuadtreeS (S)

(1) 分裂 S 为 4 个相等的子集 $O(S)$

(2) 对 $S_i \in O(S), (i=0,1,2,3)$

(a) 输出 $\Gamma_n(S_i)$;

(b) 假如 $\Gamma_n(S_1)$ 、 $\Gamma_n(S_2)$ 和 $\Gamma_n(S_3)$ 都等于 0, 则不输出 $\Gamma_n(S_4)$, 否则输出 $\Gamma_n(S_4)$;

(c) 假如 $\Gamma_n(S_i) = 1$;

• 假如 S_i 是一个元素, 输出 S_i 的符号, 并增加到 LSP 中;

• 否则执行 QuadtreeS(S)。

(d) 否则, 假如 S_i 的尺寸为 1, 增加 S_i 到 LIP_C 中; 假如 S_i 的尺寸大于 1, 增加 S_i 到 LIS_C 中。

(3) 返回

5 实验结果与分析

为了测试本文方法, 我们对 5 幅图像进行了测试 (图 2), 前 2 幅分别是 8bpp, 512×512 的 Lena 和 Barbara 各 1 幅; 后 3 幅分别是 8bpp, 2048×2048 的 Downtown 1 幅; 8bpp, 1024×1024 的 Pentagon 1 幅和 8bpp, 512×512 的 Aerial 1 幅的遥感图像。这里 bpp 表示图像中平均每个像素占用比特数 (bits per pixel)。图 3 为各算法对 Pentagon 图像在 0.5bpp 下恢复时局部放大的效果图。所有测试图像, 均用双正交 9/7 小波^[9]作 4 级小波变换, 图像边缘作对称扩展。以下所有测试均在 CPU 1.4GHz 的 PC 下完成。

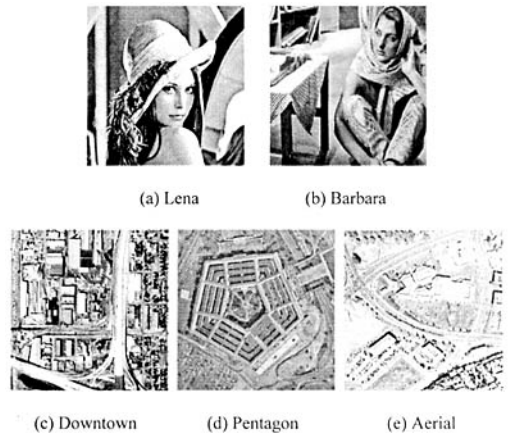


图 2 测试图像



图 3 Pentagon 在 0.5bpp 码率下各算法

计算的 PSNR 结果 (局部放大)

表 1 结果显示, 在所有码率下, 本文方法的 PSNR 均优于 SPIHT 和 SPECK, 其中比 SPIHT 平均提高 0.3dB 以上。如表 2 所示, 本文方法压缩率不仅超过 SPIHT 和 SPECK, 而且复杂度也比 SPIHT 和 SPECK 算法低, 本文方法的编码速度平均比 SPIHT 快 49%, 这是由于 SPIHT 算法的零树分裂过程中, 对零树重要性搜索非常耗时, 且重要系数的定位

表 1 本文方法与其他方法的 PSNR 性能比较 (dB)

图像	方法	bpp		
		0.25	0.5	1
Lena 512×512 8bpp	SPIHT	33.19	36.30	39.40
	SPECK	33.37	36.47	39.63
	本文	33.48	36.55	39.73
Barbara 512×512 8bpp	SPIHT	26.88	30.43	35.13
	SPECK	26.96	30.79	35.53
	本文	27.48	30.96	35.64
Downtown 2048×2048 8bpp	SPIHT	15.73	19.62	22.18
	SPECK	16.20	19.88	22.41
	本文	16.30	20.02	22.48
Pentagon 1024×1024 8bpp	SPIHT	22.13	24.40	27.43
	SPECK	22.38	24.61	27.45
	本文	22.45	24.70	27.53
Aerial 512×512 8bpp	SPIHT	24.80	27.47	31.44
	SPECK	24.96	27.78	31.74
	本文	25.05	27.87	31.86

表 2 各方法在计算 Pentagon 图像时所用的 CPU 时间 (s)

bpp	SPIHT		SPECK		本文	
	编码	解码	编码	解码	编码	解码
0.25	1.88	0.73	11.39	9.02	0.81	0.73
0.5	2.25	0.90	95.43	78.79	1.21	1.15
1	2.89	1.42	377.18	302.65	1.59	1.56

较慢,而本文基于块的搜索却很简单,对重要系数的定位非常快。SPIHT 的解码速度最快,这是因为解码时不需要对零树进行重要性搜索,而且不像零块算法中需要对 LIS 中的集合排序,但由于本文编码时间大大优于 SPIHT,使得总的计算速度仍比 SPIHT 高 35% 以上。SPECK 算法速度最慢,这主要由于对 LIS 中所有不重要的零块排序非常耗时所导致的;此外本文算法对内存占用较小,如在 1bpp 下对 Pentagon 编码时,本文算法存储链表需 3MB 内存,而 SPIHT 需 3.7MB 内存存储链表。实验还发现,在 SPECK 中所采用的复杂的集合分裂方法相对于本文简单的四叉树算法,实际上改善 PSNR 不到 0.01dB。SPIHT 算法利用子带系数的相关性进行树型编码,当产生误码时,容易导致整个图像解码失败,本文方法采用了基于块的编码方式,误码不易扩散,具有一定的鲁棒性能。另外从实验还发现,对于相关性较弱且细小目标较多的遥感图像,其小波域的能量谱变化较大,零块编码比零树编码对重要元素的定位更快,能够输出比零树编码更

少的比特。

6 结论

本文方法具有很低的复杂度和较高的图像压缩性能。与 SPIHT 和 SPECK 算法比较,本文方法只对各小波变换后的子带进行改进后的四叉树编码,降低了编解码的复杂度,通过调整 LIS 链表中零块排序策略,大幅度减少了运算时间,同时压缩性能有了进一步的提高,是一种值得推广的图像压缩算法。由于比 SPIHT 和 SPECK 更低的复杂度,所以在硬件实现上更为容易。为了在处理遥感数据时,实时性更强,我们还可以更进一步减少参与排序的零块的尺寸,使计算速度更快。在多光谱图像编码方面,本文方法与 K-L 变换去谱段相关技术相结合,也收到了良好的效果,将在另文讨论。

参考文献

- [1] Shapiro J M. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. on Signal Processing*, 1993, 41(12): 3445 - 3462.
- [2] Said A, Pearlman W A. A new, fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. on Circuits and Systems for Video Technology*, 1996,6(3): 243 - 250.
- [3] Asad Islam, Pearlman W A. An embedded and efficient low-complexity hierarchical image coder. in Proc. of SPIE Vol.3653, Visual Communications and Image Processing' 99, San Jose, CA, January 1999: 294 - 305.
- [4] Strobach P. Quadtree-structured recursive plane decomposition coding of images. *IEEE Trans. on Signal Processing*, 1991, 39(6): 1380 - 1397.
- [5] Sullivan G, Baker A. Efficient quadtree coding of images and video. *IEEE Trans. on Image Processing*, 1994, 3(3): 327 - 331.
- [6] Andrew J. A simple and efficient hierarchical image coder. *IEEE Int. Conf. on Image Proc. (ICIP-97)*, Washington, DC, 1997, 3: 658 - 661.
- [7] Hsiang S T, Woods J W. Embedded image coding using zeroblock of subband/wavelet coefficients and context modeling. *IEEE Int. Conf. on Circuits and Systems (ISCAS2000)*, Geneva, Switzerland, 2000, 3: 662 - 665.
- [8] Ordentlich E, Weinberger M, Seroussi G. A low-complexity modeling approach for embedded coding of wavelet coefficients. in Proc. 1998 IEEE Data Compression Conference, Snowbird, Utah, 1998: 408 - 417.
- [9] Antonini M, Barlaud M, Mathieu P, Daubechies I. Image coding using wavelet transform. *IEEE Trans. on Image Processing*, 1992, 1(2): 205 - 220.

吴家骥: 男, 1973 年生, 博士, 主要从事静态图像与多光谱图像的压缩编码方面的研究。

吴成柯: 男, 1938 年生, 教授, 博士生导师, 从事图像通信, 图形学、计算机视觉和模式识别等方面的研究。