

一种改进的无表零树编码的图像压缩方法

王娜, 贾传荧

(大连海事大学交通信息工程与控制实验室, 辽宁大连 116026)

摘要: 本文对 SPIHT 算法进行了改进, 得到一种在编码和解码过程中都不用链表的小波零树编码方法, 称为改进的无表零树编码. 这种编码方法大大地减少了内存消耗, 降低了编码复杂度, 提高了编码速度. 实验表明, 在相同压缩比下本文算法比 LZC 算法提高了 0.66dB, 因此为硬件实现图像压缩编码提供了一种有效的方法.

关键词: 小波变换; 零树编码; SPIHT; 无表

中图分类号: TN919.81 **文献标识码:** A **文章编号:** 0372-2112(2003)06-0879-03

An Improved Image Coding Method Based on Listless Zerotree Wavelet

WANG Na, JIA Chuanying

(The lab of Transportation information engineer and control DMU, Dalian, Liaoning 116026, China)

Abstract: It presents a zerotree coding method for gray images that uses no lists during encoding and decoding, permitting the omission of the lists requirement in Said and Pearlman's Set Partitioning In Hierarchical Trees (SPIHT) algorithm. Without lists, the memory requirement is reduced significantly, and the improved listless zerotree coding method enhances the coding speed by improving the zerotree coding process, and reduces the complexity in the encoding and decoding procedure. Experimental results show that the improved LZC codec increases the ratio of compression ratio at the same PSNR, so improved LZC method is an efficient method for image coding.

Key words: wavelet transform; zerotree coding; SPIHT; listless

1 引言

基于小波变换的图像编码是一种有效的图像压缩方法^[1]. 对于多媒体业务和网上图像通信, 它能提供更多的编码灵活性, 如码流嵌入性、图像传输渐进性等. 其中最著名的是 Shapiro 提出的小波零树编码 (EZW) 方法^[2]. 从它被提出以来, 又出现了几种基于零树理论的编码方法^[3~5], 其中 SPIHT 算法通过改变 EZW 算法的搜索策略并加入集合分裂方法, 重构图像质量有了明显提高, 成为最有效的编码小波图像方法. 但 SPIHT 算法在编码过程中需要 3 个链表来存储图像小波系数的零树结构和重要系数信息, 在编码过程中维护 3 个链表需要占用很大内存, 不利于硬件实现. 例如, 对于一幅 512 × 512 的灰度图像来说, 链表的每一项需要 18bit 的内存来存储它的行和列. 这样链表需要占用的内存是所有小波系数占用内存的两倍, 共需要内存为 1.125Mbytes. 随后 Wen 提出了无表零树编码方法 LZC^[6,7], 采用两张状态位图替代 SPIHT 算法中的三个链表, 内存占用量能减少 1085kbytes, 大大减少了内存的消耗. 本文提出的一种改进的无表零树编码算法, 采用的就是 LZC 编码的状态图结构, 但改进了树编码过程, 从而降低了树编码过程的复杂度, 实验表明对于 Lena 图像信噪比最多能提

高 0.66dB, 在高压缩比的情况下, 编码速度能提高 0.3 秒.

2 改进的无表零树编码算法

基于零树编码算法的关键是解决两个问题: 小波变换系数的组织和编码搜索策略. 在 LZC 编码算法中采用 Shapiro 的零树结构组织小波变换系数, 如图 1 所示. 树结构特征符号如下:

- $C(i, j)$: (i, j) 像素点的小波系数;
- $O(i, j)$: (i, j) 像素点的四个儿子节点;
- $D(i, j)$: (i, j) 像素点的所有后代节点;
- $F_c(i, j)$: (i, j) 像素点的重要位标志;
- $F_d(i, j)$: $D(i, j)$ 像素集的重要位标志;
- $R(i, j)$: LL 子带的所有系数.

在图 2 中, F_d 和 F_c 分别用来标记树节点本身和树节点的子孙是否有重要像素位. F_c 图和原图象的大小相同, 而 F_d 图只有原图象的四分之一. 因为位于第一层的小波系数没有子孙, 因此对于一幅 512 × 512 的灰度图像, 用来存储零树结构占用的内存只有 40kbytes. 然而在 SPIHT 算法中占用内存 1.125Mbytes, 因此大大减少了内存占有量.

LZC 零树编码可以概括为两个编码过程: 主编码过程和

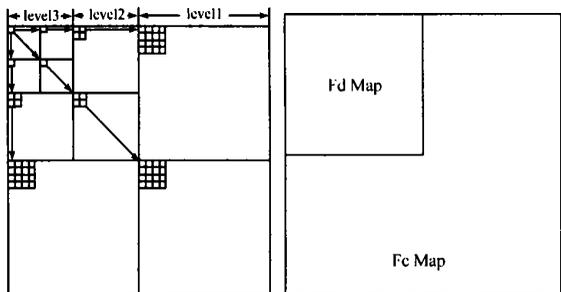


图 1 LZC 树结构图

图 2 F_c 和 F_d 状态图的大小

树编码过程. 在这两个过程中, 函数 $S_n()$ 用来判定重要象素和重要象素集合 T . $S_n()$ 函数定义如下:

$$S_n(T(i, j)) = \begin{cases} 1, & (|i, j| \in T) \quad 2^n \\ 0, & \text{其他} \end{cases}$$

LZC 主编码算法过程主要完成编码单个小波系数和编码零树集合, 树编码过程主要完成编码根节点的子孙中含有重要系数的树^[3].

在 LZC 算法中, 用 F_d 和 F_c 两张位图替代 SPIHT 算法中的三个链表 LIP, LIS, LSP 存储编码搜索顺序, 从而决定了 LZC 零树编码算法的搜索策略. 从 LZC 的树编码过程可以看出, 编码这棵树采用的是精确的深度搜索策略. 然而图像变换的能量大部分集中在低频带, 在以零树结构的编码算法中, 宽度搜索策略优于深度搜索策略, 本文提出的改进的树编码算法正是基于这种思想.

改进的树编码算法过程: 在编码根节点的子孙中含有重要系数的树时, 首先检测根节点的儿子节点是否标记为重要象素集合的根节点. 如果是, 则编码方法同 LZC. 如果不是, 则改变它的精确深度搜索策略. 即处理完儿子节点后, 不直接进行深度搜索, 而是判断儿子节点是否有子孙, 如果没有则直接返回, 节省编码时间, 缩短码流. 如果有子孙, 则判断儿子的子孙是否有重要象素, 如果有, 进行深度搜索找出重要象素, 否则不必进行深度搜索, 直接返回到主编码, 进行宽度搜索.

由上面改进树编码过程可以看出, 在编码树时加入了两个判断: 当前节点是否有子孙; 子孙中是否有重要系数. 从而改变了它的精确深度搜索策略. 其好处是: 在低比特率时, 由于高频子带的大部分系数不重要, 采用宽度搜索策略就可以把整个不重要系数的

子孙集合编码为 1 个比特, 缩短码流并节省编码时间.

3 实验结果及分析

用一个简单的例子来说明改进算法的操作步骤和优越性. 实验数据如表 1 所示, LZC 算法和本文算法的输出结果见表 2 和表 3.

表 1 实验数据

| | | | | | | | |
|-----|-----|-----|-----|---|----|-----|----|
| 63 | -34 | 49 | 10 | 7 | 13 | -12 | 7 |
| -31 | 23 | 14 | -13 | 3 | 4 | 6 | -1 |
| 15 | 14 | 3 | -12 | 5 | -7 | 3 | 9 |
| -9 | -7 | -14 | 8 | 4 | -2 | 3 | 2 |
| -5 | 9 | -1 | 47 | 4 | 6 | -2 | 2 |
| 3 | 0 | -3 | 2 | 3 | -2 | 0 | 4 |
| 2 | -3 | 6 | -4 | 3 | 6 | 3 | 6 |
| 5 | 11 | 5 | 6 | 0 | 3 | -4 | 4 |

表 2 LZC 的输出结果

| LZC | 敏检测系数和系数集 | 输出比特 | 状态位图 F_c, F_d |
|------------------|---|----------------|--|
| $N=5$ 样序 | | | $F_c = F_d = 0$ |
| | (0,0) | 1 ⁺ | $F_c(0,0) = 1$ |
| | (0,1) | 1 ⁺ | $F_c(0,1) = 1$ |
| | (1,0) | 0 | |
| | (1,1) | 0 | |
| | $S(D(0,1))$ | 1 | $F_d(0,1) = 1$ |
| | (0,2) | 1 ⁺ | $F_c(0,2) = 1$ |
| | (0,4) (0,5) (1,4) (1,5) (0,3) (0,6) (0,7) (1,6) (1,7) (1,2) (2,4) (2,5) (3,4) (3,5) (1,3) (2,6) (2,7) (3,6) (3,7) | | 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 |
| | $S(D(1,0))$ | 1 | $F_d(1,0) = 1$ |
| | (2,0) (4,0) (4,1) (5,0) (5,1) (2,1) (4,2) | | 0, 0, 0, 0, 0, 0, 0 |
| | (4,3) | 1 ⁺ | $F_c(4,3) = 1$ |
| | (5,2) (5,3) (3,0) (6,0) (6,1) (7,0) (7,1) (3,1) (6,2) (6,3) (7,2) (7,3) $S(D(1,1))$ | | 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 |
| End $N=5$ Coding | | | |
| $N=4$ 样序 | (0,0) | 1 | $F_c(0,0) = 1$ |
| | (0,1) | 1 | $F_c(0,1) = 1$ |
| | (1,0) | 1 ⁻ | $F_c(1,0) = 1$ |
| | (1,1) | 1 ⁺ | $F_c(1,1) = 1$ |
| | (0,2) | 1 | $F_c(0,2) = 1$ |
| | $S(D(0,2))$ (0,3) (0,6) (0,7) (1,6) (1,7) (1,2) (2,4) (2,5) (3,4) (3,5) (1,3) (2,6) (2,7) (3,6) (3,7) (2,0) (4,0) (4,1) (5,0) (5,1) (2,1) (4,2) | | 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 |
| | (4,3) | 1 | $F_c(4,3) = 1$ |
| | (5,2) (5,3) (3,0) (6,0) (6,1) (7,0) (7,1) (3,1) (6,2) (6,3) (7,2) (7,3) $S(D(1,1))$ | | |
| End $N=4$ Coding | | | |

由表 2 可以看出, LZC 算法输出码流共包含 95 个比特, 平均每个象素用 1.48 个比特. 而本文算法输出码流 (表 3) 共有 57 个比特, 平均每个象素用 0.89 个比特, 平均每个象素节省了 0.59 个比特. 因此同样的信噪比提高了压缩比; 同时简

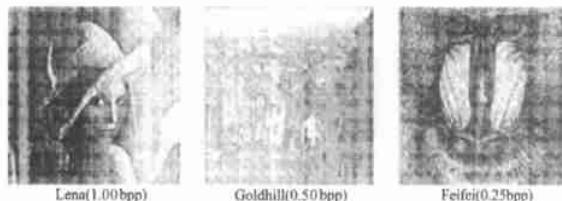


图 3 实验结果图象

化了树编码过程,提高了压缩速度.

表 3 改进 LZC 的输出结果

| 改进的 LZC | 敏检测系数和系数集 | 输出比特 | 状态位图 F_c, F_d |
|------------------|---|---------------------------|-----------------|
| $N=5$ 样序 | | | $F_c = F_d = 0$ |
| | (0,0) | 1 ⁺ | $F_c(0,0) = 1$ |
| | (0,1) | 1 ⁺ | $F_c(0,1) = 1$ |
| | (1,0) | 0 | |
| | (1,1) | 0 | |
| | $S(D(0,1))$ | 1 | $F_d(0,1) = 1$ |
| | (0,2) | 1 ⁺ | $F_c(0,2) = 1$ |
| | $S(D(0,1)) (0,3) S(D(0,3)) (1,2) S(D(1,2)) (1,3) S(D(1,3))$ | 0, 0, 0, 0, 0, 0, 0 | |
| | $S(D(1,0))$ | 1 | $F_d(1,0) = 1$ |
| | (2,0) $S(D(2,0)) (2,1) S(D(2,1)) (4,2)$ | 0,0,0,0,0 | |
| | (4,3) | 1 ⁺ | $F_c(4,3) = 1$ |
| | (5,2) (5,3) (3,0) $S(D(3,0)) (3,1) S(D(3,1)) S(D(1,1))$ | 0, 0, 0, 0, 0, 0, 0 | |
| End $N=5$ Coding | | | |
| $N=4$ 样序 | (0,0) | 1 | $F_c(0,0) = 1$ |
| | (0,1) | 1 | $F_c(0,1) = 1$ |
| | (1,0) | 1 ⁻ | $F_c(1,0) = 1$ |
| | (1,1) | 1 ⁺ | $F_c(1,1) = 1$ |
| | (0,2) | 1 | $F_c(0,2) = 1$ |
| | $S(D(0,2)) (0,3) S(D(0,3)) (1,2) S(D(1,2)) (1,3) S(D(1,3)) (2,0) S(D(2,0)) (2,1) (4,2)$ | 0, 0, 0, 0, 0, 0, 0, 0, 0 | |
| | (4,3) | 1 | $F_c(4,3) = 1$ |
| | (5,2) (5,3) (3,0) $S(D(3,0)) (3,1) S(D(3,1)) S(D(1,1))$ | | |

End $N=4$ Coding

使用的实验图象为 512 x512 x8bit 的灰度图象,如图 3 所示.小波滤波器采用 Antonini 9/7 双正交滤波器,小波变换级数为 8 级,在图象边界处采用了对折式扩展.采用峰值信噪比 (PSNR) 来比较不同的算法,实验的硬件平台为 Pentium 3,软件平台为 Visual C++ 6.0,结果如表 4 所示.

表 4 不同编码方法间的 PSNR (dB) 的比较

| bpp | Lena | | Goldhill | | Feifei | |
|------|--------|-------|----------|-------|--------|-------|
| | LZC 算法 | 本文算法 | LZC 算法 | 本文算法 | LZC 算法 | 本文算法 |
| 0.10 | 28.35 | 29.01 | 27.35 | 27.89 | 27.58 | 28.13 |
| 0.25 | 31.10 | 31.65 | 29.35 | 29.85 | 31.15 | 31.64 |
| 0.50 | 33.70 | 34.09 | 31.89 | 32.08 | 32.45 | 32.76 |
| 1.00 | 37.01 | 37.30 | 35.32 | 35.41 | 35.62 | 35.73 |
| 2.00 | 41.52 | 41.61 | 38.69 | 38.88 | 40.23 | 40.36 |
| 2.66 | 43.35 | 43.45 | 42.16 | 42.27 | 41.52 | 41.61 |

由表 4 可以看出,本文编码算法获得的 PSNR 高于 LZC 算法.由图 3 可以看出,在不同码率的情况下,复原图像质量都没有下降.并且在编码过程中减少了内存的占用,提高了编码速度,且便于硬件实现.

4 结论

静态图象经过双正交小波变换分解,采用本文编码算法,减少了内存的占用量,降低了树编码的复杂度,提高了编码速度.在相同的 PSNR 情况下,提高了压缩比,也为硬件实现图像压缩编码找到了一种有效的方法.

参考文献:

- [1] M Antonini, M Barlaud, P Mathieu, I Daubechies. Image coding using wavelet transform[J]. IEEE Trans. Image Processing, 1992 (4): 205 - 220.
- [2] J M Shapiro. Embedded image coding using zerotrees of wavelet coefficients [J]. IEEE Trans. Signal Process, 1993, 41 (12): 3445 - 3462.
- [3] A Said, W A Pearlman. A new fast, and efficient image codes based on set partitioning in hierarchical trees [J]. IEEE Trans. on Circuits Syst. Video Technol, 1996, 6(3): 243 - 249.
- [4] Zandi A, Allen JD, Schwartz EL, Boliek M. CREW: Compression with reversible embedded wavelets [J]. IEEE Compression Data Compression Conf. snowbird, 1995(3): 212 - 221.
- [5] Frederick W Wheeler, W A Pearlman. Low-memory packetized SPIHT Image compression[A]. Asilomar Conf. on Signals, System and computers [C]. Pacific Grove, CA, 1999.
- [6] Wen Kuo Lin, N Burgess. Listless zerotree coding for color images [A]. Proc Of the 32nd Asilomar Conf. On signals, System & computers [C]. 1998. 231 - 235.
- [7] Wen Kuo Lin, N Burgess. Low memory color image zerotree coding, information, decision and control [J]. Info. Decision and Controls, 1999 (2): 91 - 95.

作者简介:



王 娜 女,1977 年 11 月生于河北省,博士生,研究方向为多媒体信息处理,主要研究兴趣为图像编码、小波理论、模式识别等.



贾传炎 男,1940 年 7 月生于上海市,教授,博士生导师,主要研究兴趣:信号处理、图像分析以及图像和视频的压缩编码技术,国内外发表论文 50 余篇.