

文章编号:1001-9081(2005)03-0603-03

## 改进的无链表零树编码算法

王慧斌,李会方,李 正,王立波

(西北工业大学 电子信息学院,陕西 西安 710072)

(whuibin1979@sina.com)

**摘 要:**通过对 SPIHT 算法和 LZC 算法的研究,提出一种改进的无链表零树编码算法。该算法改进了树编码过程,降低了编码复杂度,易于硬件实现,并且用小波系数的最高比特位来存储标志图,使得编码所需内存进一步减少。实验结果表明在相同的压缩比下,新算法重建图像的峰值信噪比明显优于 LZC 算法,而仅比 SPIHT 算法有略微的降低。

**关键词:**小波变换;零树编码;SPIHT;LZC;改进

**中图分类号:**TP391.41 **文献标识码:**A

## Modified coding algorithm based on listless zerotree wavelet

WANG Hui-bin, LI Hui-fang, LI Zheng, WANG Li-bo

(College of Electronic and Information, Northwestern Polytechnical University, Xi'an Shaanxi 710072, China)

**Abstract:** The modified coding algorithm based on listless zerotree wavelet was proposed by studying SPIHT and LZC. The zerotree coding process was improved, and the complexity of the encoding procedure was reduced. The novel algorithm was easy to be realized by hardware. The top bits of transformed coefficients were used to store flag maps, and the memory requirements of coding process were further reduced. Experiment results show that PSNR(Peak Signal Noise Ratio) values of the novel algorithm are obviously better than those of LZC, and less than those of SPIHT appreciably at the same compression ratio.

**Key words:** wavelet transform; zerotree coding; SPIHT; LZC; modification

### 0 引言

图像的数据量很大,为了使图像数据的传输、存储成为可能,并尽量降低通信成本,图像数据的压缩处理是必须的。近年来,图像压缩编码方法层出不穷,其中小波编码领域尤为活跃。1992 年,Shapiro<sup>[1]</sup>提出了嵌入零数小波算法(EZW),通过利用节点间的父子关系,使得小波系数的编码效率得以提高。1996 年,Said 和 Pearlman<sup>[2]</sup>在 EZW 算法的基础上,提出了分层树集合划分算法(SPIHT),它仍然采用了树状结构来组织小波系数,所不同的是利用集合的划分来进行编码,在压缩效率和实现简便性等方面都有了很大的提高。

然而由于在编码过程中,需要使用 3 个链表来存储系数的坐标和集合的状态,这就使得 SPIHT 算法需要大量的内存。例如,对于一幅 512×512 的彩色图像来说,链表中的每一个节点至少需要 18bit 的内存来存储它的坐标。因此在 DSP 和 VLSI 应用中,势必提高其实现的成本,不利于硬件实现。为此,Wen<sup>[3]</sup>提出了无表零树编码算法(LZC),采用两个状态标志图来替代 SPIHT 中的 3 个链表,大大减小了内存的消耗。本文提出的改进的无链表零树编码算法,采用的就是 LZC 算法的标志图结构,但改进了树编码过程,从而降低了树编码过程的复杂度,并且用小波系数的最高比特位来存储标志图,使得编码所需内存进一步减少。实验结果表明在相同的压缩比下,新算法重建图像的峰值信噪比(PSNR)优于 LZC 算法,而仅比 SPIHT 算法有略微的降低。

### 1 SPIHT 算法简介

图像数据经小波分解后,系数成树状分布。根据这个特点,可以定义一种数据结构——空间向量树。图 1 表示的是一个 4 级小波分解后的空间向量树结构。可以看出,除最后一级的低频分量集合(图中标有“★”的部分)外,每一个节点都有 4 个子节点,一级一级连接起来,构成树状结构。

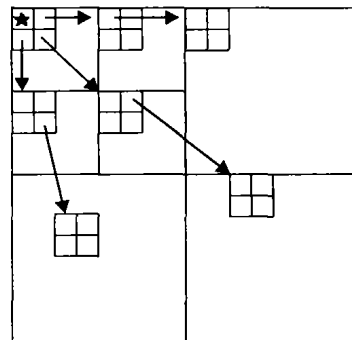


图 1 空间向量树结构

在上述向量树的基础上,Said 和 Pearlman 提出了 SPIHT 算法。SPIHT 算法是 EZW 算法的改进,实现了内嵌编码的逐渐浮现式的图像传输。内嵌编码中首先传输的是重要信息,这里是幅值较大的变换系数。重要性测试函数定义为:

$$S_n(X(i, j)) = \begin{cases} 1, & \max_{C(k, l) \in X(i, j)} |C(k, l)| \geq 2^n \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

收稿日期:2004-08-06;修订日期:2004-10-21

**作者简介:**王慧斌(1979-),男,山西大同人,硕士研究生,主要研究方向:信号与信息处理;李会方(1962-),男,陕西西安人,副教授,博士,主要研究方向:图像处理、计算机网络;李正(1968-),男,四川南充人,高级工程师,硕士研究生,主要研究方向:通信、计算机网络;王立波(1979-),男,黑龙江齐齐哈尔人,硕士研究生,主要研究方向:图像处理、计算机通信。

式中  $X(i, j)$  表示一节点坐标集合,  $C(k, l)$  为节点  $(k, l)$  的小波变换系数。

在 SPIHT 算法中所使用的符号定义如下:  $H$  表示空间向量树根的所有节点;  $O(i, j)$  表示节点  $(i, j)$  的 4 个子代节点;  $D(i, j)$  表示节点  $(i, j)$  的所有子孙节点;  $L(i, j)$  表示节点  $(i, j)$  的间接子孙, 即  $L(i, j) = D(i, j) - O(i, j)$ 。在该算法中, 还定义了 3 个链表 LIS, LIP 和 LSP。其中 LIS 为不重要像素集合链表, 它分为两种类型: A 型和 B 型, A 型用来表示集合  $D(i, j)$ , B 型用来表示集合  $L(i, j)$ ; LIP 为不重要像素点链表; LSP 为重要像素点链表。

SPIHT 算法中集合分割的策略如下:

- 1) 初始化  $(i, j), D(i, j), (i, j) \in H$ ;
- 2) 如果  $S_n(D(i, j)) = 1$ , 则  $D(i, j)$  分割为  $D(i, j) = L(i, j) + O(i, j)$ ;
- 3) 如果  $S_n(L(i, j)) = 1$ , 则  $L(i, j)$  分割为 4 个  $D(k, l), (k, l) \in O(i, j)$ 。

## 2 LZC 算法简介

LZC 算法继承了 SPIHT 算法的零树结构, 算法中的树标号与 SPIHT 算法稍有区别, 它的树标号如下:  $C(i, j)$  为节点  $(i, j)$  的小波系数;  $O(i, j)$  为节点  $(i, j)$  的 4 个子代节点;  $D(i, j)$  为节点  $(i, j)$  的所有子孙节点;  $R(i, j)$  为子带 LL 上的节点集合。

与 SPIHT 算法相似, LZC 算法也用两个树标号  $C(i, j)$  和  $D(i, j)$  来指出小波系数的位置信息。LZC 算法允许最低子带 (LL) 是奇数维, 而 SPIHT 算法则必须是偶数维, 这在硬件实现时不得不用额外的电路来调整图像尺寸, 因此 LZC 算法更便于硬件实现。

与 SPIHT 算法不同, 在 LZC 算法中分别用两个状态标志图 Fc, Fd 来表示节点  $(i, j)$  和后代集合  $D(i, j)$  的重要性。如图 2 所示, Fc 图与图像的大小相同, 而 Fd 图仅是图像大小的四分之一, 这是因为第一层的小波系数没有后代。对于一幅  $512 \times 512$  的彩色图像来说, 两个标志图需要的编码内存为 120kB, 而在 SPIHT 算法中, 如果链表的长度为图像系数量的两倍, 则需 3.539MB 的编码内存。这样大的内存要求, 导致 SPIHT 算法的硬件实现变得非常困难, 而 LZC 算法所需内存大约仅是 SPIHT 算法的三十分之一。

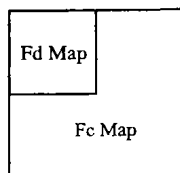


图 2 Fd 图和 Fc 图

## 3 改进的无链表零树编码算法

改进算法采用了 EZW 算法的零树结构, 如图 3 所示。在新算法中, 引入了 4 个新的树标号: 不重要系数数量  $N_c(i, j)$ , 不重要集合数量  $N_d(i, j)$ , 不重要系数数量导出值  $V_c(i, j)$  和不重要集合数量导出值  $V_d(i, j)$ 。并且, 对重要性测试函数进行了修正:

$$S_n(X(i, j)) = \begin{cases} 1, & \max_{C(k, l) \in X(i, j)} 2^{n+1} > |C(k, l)| \geq 2^n \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

式中  $X(i, j)$  代表  $C(i, j)$  或  $D(i, j)$ 。

算法主程序如下:

```
output n = [log2(max_{(i, j)} |C(i, j)|)], clear Fc 和 Fd
for each entry (i, j) do:
    if Fc(i, j) = 1 then
        output the n-th bit of |C(i, j)|
    if Fc(i, j) = 0 & S_n(C(i, j)) = 1 then
        output sign of C(i, j) and set Fc(i, j) = 1
    if Fd(i, j) = 0 & S_n(D(i, j)) = 1 then
        set Fd(i, j) = 1, calculate N_c(i, j) 和 N_d(i, j)
    if N_c(i, j) > 0 then output V_c(i, j)
    if N_d(i, j) > 0 then output V_d(i, j)
    if Fd(i, j) = 1 then
        calculate N_c(i, j) 和 N_d(i, j)
        if N_c(i, j) > 0 then output V_c(i, j)
        if N_d(i, j) > 0 then output V_d(i, j)
decrease n by 1 until n = 0
```

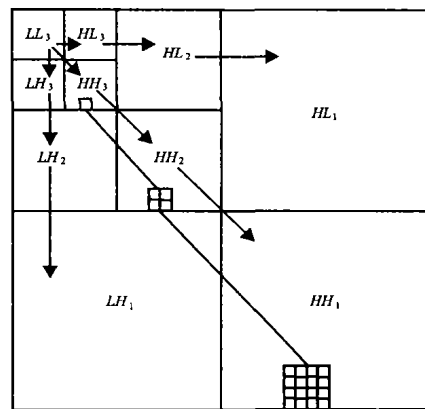


图 3 改进算法的零树结构

改进算法中, 分类过程和细化过程合并为一个过程, 这样使得计算时间减少。解码时只需知道从不重要到重要的转变, 而这个信息可通过  $N_c(i, j)$  和  $N_d(i, j)$  的变化来获得, 并通过  $V_c(i, j)$  和  $V_d(i, j)$  的值来判定变化的情况, 我们通过一个例子来解释如何获取  $V_c(i, j)$  和  $V_d(i, j)$  的值。 $D(i, j)$  分割为 4 个节点:  $C(2i, 2j), C(2i, 2j+1), C(2i+1, 2j)$  和  $C(2i+1, 2j+1)$ 。假如从标志图 Fc 得知仅有一个系数  $C(2i+1, 2j+1)$  是不重要的, 则  $N_c(i, j) = 1$ 。如果  $S_n(C(2i+1, 2j+1)) = 0$ , 则  $V_c(i, j) = 0$ ; 如果  $S_n(C(2i+1, 2j+1)) = 1$ , 则  $V_c(i, j) = 1$ 。假如从标志图 Fd 得知有两个后代集合  $D(2i, 2j), D(2i+1, 2j+1)$  是不重要的, 则  $N_d(i, j) = 2$ 。如果  $S_n(D(2i, 2j)) = 0$  &  $S_n(D(2i+1, 2j+1)) = 0$ , 则  $V_d(i, j) = 0$ ; 如果  $S_n(D(2i, 2j)) = 0$  &  $S_n(D(2i+1, 2j+1)) = 1$ , 则  $V_d(i, j) = 1$ ; 如果  $S_n(D(2i, 2j)) = 1$  &  $S_n(D(2i+1, 2j+1)) = 0$ , 则  $V_d(i, j) = 2$ ; 如果  $S_n(D(2i, 2j)) = 1$  &  $S_n(D(2i+1, 2j+1)) = 1$ , 则  $V_d(i, j) = 3$ 。

为了验证小波系数的最高比特位的确可以用来存储标志图, 我们必须对小波系数幅值的上界进行讨论。图 4 为 S-尺度离散小波变换的一个过程。

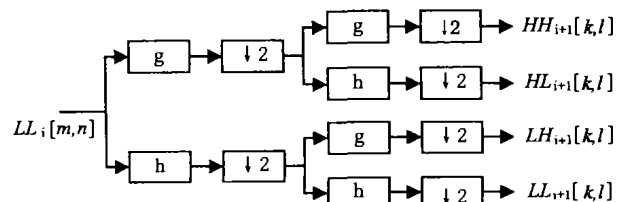


图 4 S-尺度离散小波变换的一个过程

当  $i = 0$  时,  $LL_0[m, n]$  等于输入图像  $a[m, n]$ , 输出系数如下:

$$HH_1[k, l] = \sum g[n] \sum g[m] a[2k - m, 2l - n] \quad (3)$$

$$HL_1[k, l] = \sum g[n] \sum h[m] a[2k - m, 2l - n] \quad (4)$$

$$LH_1[k, l] = \sum h[n] \sum g[m] a[2k - m, 2l - n] \quad (5)$$

$$LL_1[k, l] = \sum h[n] \sum h[m] a[2k - m, 2l - n] \quad (6)$$

从(3)式可知,  $HH_1[k, l]$  的幅值满足:

$$|HH_1[k, l]| = \sum |g[n]| \sum |g[m]| a[2k - m, 2l - n] \leq \sum |g[n]| \sum |g[m]| a_{\max} \quad (7)$$

式中  $a_{\max} = \max_{(m, n)} |a[m, n]|$ 。令  $S_h = \sum |h[n]|$  和  $S_g = \sum |g[n]|$ , (3) 式简化为  $|HH_1[k, l]| \leq S_g^2 a_{\max}$ 。同理得出,  $|HL_1[k, l]| \leq S_g S_h a_{\max}$ ,  $|LH_1[k, l]| \leq S_h S_g a_{\max}$ , 和  $|LL_1[k, l]| \leq S_h^2 a_{\max}$ 。进一步分解为  $HH_2, HL_2, LH_2$  和  $LL_2$ 。这4个子带小波系数的上界分别为:  $|HH_2[k, l]| \leq S_g^2 S_h^2 a_{\max}$ ,  $|HL_2[k, l]| \leq S_g S_h^3 a_{\max}$ ,  $|LH_2[k, l]| \leq S_g^3 S_h a_{\max}$  和  $|LL_2[k, l]| \leq S_h^4 a_{\max}$ 。对于S-尺度离散小波变换,通过迭代得出:

$$|HH_S[k, l]| \leq S_g^2 S_h^{2(S-1)} a_{\max} \quad (8)$$

$$|HL_S[k, l]| \leq S_g S_h^{2S-1} a_{\max} \quad (9)$$

$$|LH_S[k, l]| \leq S_g S_h^{2S-1} a_{\max} \quad (10)$$

$$|LL_S[k, l]| \leq S_h^{2S} a_{\max} \quad (11)$$

如果使用 9/7 滤波器<sup>[4]</sup>, 则  $S_h = 1.952\ 108\ 3$ ,  $S_g = 1.835\ 127\ 5$ 。对于一个4-尺度的小波分解,小波系数幅值的上界(从(11)式得出)为  $210.878 \times a_{\max}$ 。如果  $a_{\max} = 2^8 - 1$ , 幅值上界小于  $2^{16}$ 。因此,4个字节的整型变量用来存储小波系数值和标志图  $F_c, F_d$  是足够的。

## 4 实验结果

分别用 LZW, SPIHT 和本文算法对3幅图像进行实验,结果如表1所示。3幅图像为  $512 \times 512 \times 8\text{bit}$  的灰度图像,如

图5所示。小波滤波器采用 db9/7 双正交滤波器,对图像进行6级小波分解,对于图像的边界处采用周期延拓。

表1 3种压缩算法的峰值信噪比

码率	Lena			Baboon			Goldhill		
	LZC	SPIHT	本文	LZC	SPIHT	本文	LZC	SPIHT	本文
0.10	27.24	28.01	27.93	26.46	27.02	26.94	26.23	26.78	26.71
0.25	29.96	30.50	30.41	30.04	30.51	30.43	28.24	28.73	28.64
0.50	32.53	32.93	32.86	31.32	31.65	31.58	30.76	30.95	30.89



图5  $512 \times 512 \times 8\text{bit}$  的灰度图像

由表1可以看出,在相同压缩比下,本文算法获得的 PSNR 明显高于 LZC 算法,而略微低于 SPIHT 算法。

## 参考文献:

- [1] SHAPIRO JM. Embedded image coding using zerotrees of wavelet coefficients[J]. IEEE Transactions on Signal Processing, 1993, 41(12): 3445-3462.
- [2] SAID A, PEARLMAN W. A new fast and efficient image codec based on set partitioning in hierarchical trees[J]. IEEE Transactions on Circuit and System for Video Technology, 1996, 6(3): 243-250.
- [3] WEN KL, BIRGESS N. Listless zerotree coding for color images [A]. Proceedings of the 32nd Asilomar Conference on Signals, System & Computers[C], 1998. 231-235.
- [4] ANTONINI M, BARLAUD M, MATHIEU P, et al. Image coding using wavelet transform[J]. IEEE Transactions on Image Processing, 1992, 1(2): 205-220.

(上接第597页)

总碎片数  $N$  为10。其中  $\text{MAX} = 5$ ,  $\text{MIN} = 2$ , 则匹配信息表 Pair 为  $11 \times 7$  (见表2)。

结合表2, 根据序列查找算法, 得到匹配关系表如下:

$$\text{List}[10][9] = \begin{bmatrix} 9 & 2 & 0 & 0 & 1 & 10 & 0 & 0 & 0 \\ 1 & 5 & 9 & 0 & 9 & 10 & 3 & 5 & 6 \\ 10 & 3 & 9 & 1 & 9 & 1 & 3 & 0 & 0 \\ 3 & 3 & 1 & 10 & 1 & 10 & 5 & 0 & 0 \\ 5 & 5 & 1 & 3 & 1 & 3 & 6 & 8 & 7 \\ 6 & 4 & 1 & 5 & 1 & 5 & 8 & 4 & 0 \\ 8 & 5 & 5 & 6 & 5 & 6 & 7 & 4 & 2 \\ 7 & 3 & 5 & 8 & 5 & 8 & 2 & 0 & 0 \\ 4 & 3 & 6 & 8 & 6 & 8 & 0 & 0 & 0 \\ 2 & 3 & 8 & 7 & 8 & 7 & 0 & 0 & 0 \end{bmatrix}$$

碎片拼合序列为  $\text{List}[10][1] = \{9, 1, 10, 3, 5, 6, 8, 7, 4, 2\}$ 。

## 参考文献:

- [1] 王敏, 李君, 黄心汉. 平面图形的分类与识别方法研究[J]. 微处理机, 2001, (3): 22-25.
- [2] 杨洛斌. 形状匹配技术在文物复原中的研究与应用[D]. 西北大学, 2002.
- [3] da GAMA LEITAO HC, STOLFI J. A Multiscale Method for the Reassembly of Two-Dimensional Fragmented Objects[J]. Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(9): 1239-

1251.

- [4] 周石林, 廖文和, 王琪. 平面非规则曲线匹配的一个判定条件[J]. 南京航空航天大学学报, 2003, 35(5): 561-564.
- [5] 何斌, 马天予, 王运坚, 等. Visual C++ 数字图像处理[M]. 北京: 人民邮电出版社, 2001.
- [6] 苏步青, 胡和生, 沈纯理, 等. 微分几何[M]. 北京: 高等教育出版社, 1985.

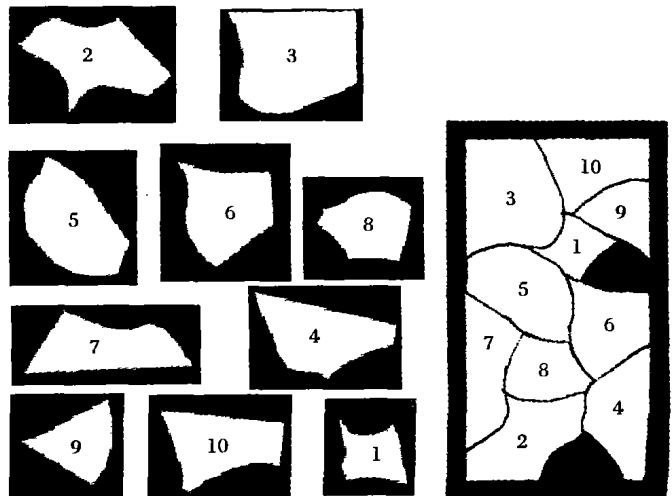


图2 原始碎片

图3 拼合后碎片