改进的 SPIHT 算法 [1]

李洪刚 王 桥 吴乐南

(东南大学无线电工程系 南京 210096)

摘 要 SPIHT 算法是一种高效的嵌入式的零树编码算法，然而，它需要大量的内存空间，不利于 DSP 或 VLSI 的实现。LZC 算法可以极大地降低编解码器的内存需求，但同时也降低了编解码器的性能。该文利用 LZC 算法的思想，改进了原来的 SPIHT 算法，使得在仅仅在 LZC 算法的内存需求基础上，达到 SPIHT 算法的性能要求。同时又提出了一种近似搜索算法来提高编码器的速度。

关键词 零树编码，SPIHT, LZC

中图号 TN911.73

1 引 言

由于图像的数据量大，图像压缩一直是多媒体数据通信的瓶颈之一。寻求一个简便的、快捷的、高效的压缩算法具有很重要的实用意义。Shapiro[1] 首先提出的嵌入式零树小波 (EZW, Embedded Zerotree Wavelet) 编码方案，已被公认为是静止图像变换编码领域中重要的算法之一。在低码率条件下，它有效地克服了传统基于 DCT 的编解码器所造成的方块效应，使其性能明显优于后者。

正是由于 EZW 算法良好的性能，人们在其基础上又提出了各种改进的方法。其中，由 Said 和 Pearlman 提出的 SPIHT 算法 [2,3] 是一种高效的改进算法。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。实验证明这种表示系数的方法更为有效，但是由于编码过程中需要使用 3 个链表，因此 SPIHT 算法需要大量的内存。在 DSP 和 VLSI 中，大容量内存势必提高其实现的成本，给硬件实现 SPIHT 算法带来了很大的困难。为此，Lin 和 Burgress 提出了一种改进算法—— LZC(Listless Zerotree Coding) 算法 [4]。它简化了原来的 SPIHT 算法，并利用标志位图代替了原来的链表，从而极大地降低了内存的需求。不过这种算法也降低了编码器的性能。

通过对 SPIHT 算法和 LZC 算法的研究，本文提出了一种改进的 SPIHT 算法，它结合了 SPIHT 和 LZC 两者的优势，只需要与 LZC 算法相同的内存大小，就可以达到甚至超过 SPIHT 算法所能提供的性能。

为了在不增加内存需求的条件下，降低算法的时间复杂度，本文通过分析小波系数的统计特性，提出了一种近似算法，通过极少量的图像质量的损失来极大地提高编码器的速度。

2 SPIHT 和 LZC 算法

在讨论 SPIHT 和 LZC 算法之前，首先引入几个表征多级树结构的符号。如图 1 所示。$T(i,j)$ 表示以节点 (i,j) 为根的子树；$C(i,j)$ 表示节点 (i,j) 的小波变换系数；$D(i,j)$ 表示节点 (i,j) 的所有后代节点；$O(i,j)$ 表示节点 (i,j) 的所有儿子节点；$L(i,j)=D(i,j)-O(i,j)$ 表示节点 (i,j) 所有儿子节点的后代节点，即除了儿子节点外的所有后代节点；

R 表示小波分解后所有的顶层节点；

[1] 2001-02-11 收到，2001-07-11 定稿
国家自然科学基金 (60002008) 资助

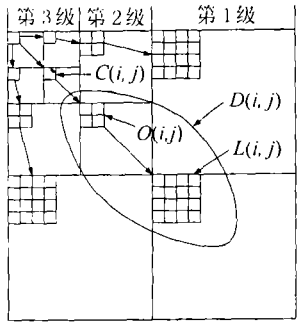

图 1 小波系数的多级树状图

$$S_n(T)=\begin{cases}1, & \max\limits_{(i,j)\in T}\{|C(i,j)|\geq 2^n\}\\ 0, & \text{其它}\end{cases}$$

由于模值大的小波系数对于图像的重建影响也大，所以在嵌入式零树编码中都采用了优先编码传输模值大的系数的方案。为了有效地编码传输节点的坐标位置，它们均基于系数的多级树状结构，利用集合的划分来对位置进行编码。其基本思想是判断集合中的所有系数的模值是否都小于给定阈值，如果成立则只需要一个符号就可以表示集合中所有的节点，否则按照一定的规则将集合划分成若干小的集合。所不同的是，传统 EZW 算法选取的是子树集合 $T(i,j)$，而 SPIHT 算法选用的是 A 型集合 $D(i,j)$ 和 B 型集合 $L(i,j)$。为了充分利用已编码过的信息， SPIHT 算法使用了 3 个链表：无意义像素链表 (LIP ； List of Insignificant Pixels)，有意义像素链表 (LSP ； List of Significant Pixels)，无意义集合链表 (LIS ； List of Insignificant Sets) 来存储编码过的节点和集合。其算法过程如下：

(1) 输出 $n=[\log_2(\max\{|C(i,j)|\})]$，将 LSP 置空，把 R 中的所有节点 (i,j) 放入 LIP，并且将其后代集合放入 LIS，并标记 A 型集合。

(2) 对所有 LSP 中的节点 (i,j)，输出 $|C(i,j)|$ 的第 n 位比特值。

(3) 对所有 LIP 中的节点 (i,j)，输出 $S_n(i,j)$。如果 $S_n(i,j)=1$，则输出 $C(i,j)$ 的符号，并将其移到 LSP 中。

(4) 对所有 LIS 中的集合 (i,j)，如果是 A 型集合，即 $D(i,j)$，则输出 $S_n(D(i,j))$；如果 $S_n(D(i,j))=1$，则

(a) 对所有 $O(i,j)$ 中的节点 (k,l)，输出 $S_n(k,l)$。如果 $S_n(k,l)=1$，则输出 $C(k,l)$ 的符号，并且将其置入 LSP，否则将其置入 LIP；

(b) 删除这个集合。如果其对应的 $L(i,j)$ 非空，则将集合 $L(i,j)$ 置入 LIS，并标记为 B 型集合。

如果是 B 型集合，即 $L(i,j)$，则输出 $S_n(L(i,j))$。如果 $S_n(L(i,j))=1$，则删除这个集合，将所有的 $D(k,l)$，$(k,l)\in O(i,j)$ 置入 LIS，并标记为 A 型集合。

(5) 如果 $n=0$，则终止，否则 $n=n-1$，并转到 (2) 继续执行。

由于链表中的每个元素都必须保存其节点的位置，因此对于一个 512×512 的灰度图象，链表所需要的存储空间至少为 18bit×512×512/8bit/1k=576kB。再加上小波变换系数所需要的存储空间，这对于 DSP 和 VLSI 实现来说成本太高了。为此 Lin 和 Burgress 提出了一种改进算法——LZC 算法 [4]。它的重要思想是利用标志位图 F_C 和 F_D 和递归算法来代替原来的链表结构。由于 LZC 算法只使用了一种集合 $D(i,j)$ 来编码节点的位置信息。可以用一个比特 $F_D(i,j)$ 来指示集合 $D(i,j)$ 是否被划分，一个比特 $F_C(i,j)$ 表明该点是否有意义。由于底层节点没有后代，因此对于 512×512 的灰度图像，LZC 算法只需要 (512×512bit+256×256bit)/8bit/1k=40kB，极大地降低了编解码器的内存需求，从而有利于硬件实现。然而由于简化了 SPIHT 算法，使得编码性能也随之降低。

3 改进的 SPIHT 算法

根据集合划分的准则可知，对于给定阈值 2^n，当且仅当 $S_n(T)=1$ 时，集合 T 才被划分成若干个小集合。如果节点 $(i,j)\in$ LIS，则表明集合 $D(i,j)$ 和 $L(i,j)$ 中有且仅有一个属于

LIS，即 $S_{n+1}(D(i,j))=0$ 或 $S_{n+1}(L(i,j))=0$。由于集合 $D(i,j)=O(i,j)\cup L(i,j)$，因此必然有 $S_{n+1}(L(i,j))=0$。所以有

命题 当节点 $(i,j)\in$ LIS 时，集合 $D(i,j)\in$ LIS 的充要条件是 $S_{n+1}(O(i,j))=0$。通过这个结论，我们就可以在不增加任何存储空间的条件下，很容易地区分集合 $D(i,j)$ 和 $L(i,j)$。

基于上述分析，结合 LZC 编码思想，本文提出了一种基于 SPIHT 的改进算法，其算法的伪代码如下：

(1) Output $n=[\log_2(\max\{|C(i,j)|\})]$, clear F_C and F_L。

(2) for all$(i,j)\in R$ do EncodeTree(i,j,n)

(3) decrease n by 1 and go back to step2 until $n=0$

其中核心部分为一个递归过程 EncodeTree，它需要用到 EncodeNode 函数。它们的伪代码分别如下：

函数 EncodeTree(i,j,n)

(1) if $F_C(i,j)=1$ then output the n-th bit of $|C(i,j)|$

else EncodeNode(i,j,n)

(2) if $F_L(i,j)=1$ then for each$(k,l)\in O(i,j)$ EncodeTree(k,l,n)

else if $F_C(k,l)$for all $(k,l)\in O(i,j)$ then

output $S_n(D(i,j))$

if $S_n(D(i,j))=1$ then

output $S_n(L(i,j))$

if $S_n(L(i,j))=1$ then

set $F_L(i,j)=1$

for each $(k,l)\in O(i,j)$ EncodeTree(k,l,n)

else for each $(k,l)\in O(i,j)$ EncodeNode(k,l,n)

else output $S_n(L(i,j))$

if $S_n(L(i,j))=1$ then

set $F_L(i,j)=1$

for each $(k,l)\in O(i,j)$ EncodeTree(k,l,n)

else for each $(k,l)\in O(i,j)$ EncodeNode(k,l,n)

函数 EncodeNode(i,j,n)

(1) output $S_n(i,j)$

(2) if $S_n(i,j)=1$ then output the sign of $C(i,j)$ and set $F_C(i,j)=1$

它也采用类似的标志位图 F_C 和 F_L，其中 $F_L(i,j)$ 表示集合 $L(i,j)$ 是否被划分。利用上述结论，通过对所有的 $(k,l)\in O(i,j)$ 来判断 $F_C(k,l)$ 是否置位来识别集合 $D(i,j)$ 和 $L(i,j)$。

尽管沿用了 SPIHT 算法的思想，但由于采用了递归过程，编码的顺序发生了较大的变化，由原先的广度优先，即按小波系数从高层到底层的顺序，变为了深度优先，即按照横向、纵向以及对角线索那个方向依次编码，而且将孤立点与其相应的集合一同编码，使其在性能上与 SPIHT 有极其微小的差异，有时甚至超越 SPIHT 的性能。

4 实验结果

根据本文提出的改进算法，我们对多幅图像进行的编解码实验。实验中采用了具有提升结构 (Lifting Scheme) 的 5-3 整系数小波变换，对于图像边界处采用不重复的对称延拓，而且编码后不再经过算术编码或其它熵编码。表 1 记录了实验的结果。图 2 显示了本算法和 SPIHT 及 LZC 算法在主观和客观性能上的比较。

表 1 3 种压缩算法的峰值信噪比 (PSNR) 的比较

	Lenna(256×256)(dB)			Mandrill(512×512)(dB)			Goldhill(512×512)(dB)		
码率	LZC	SPIHT	本文	LZC	SPIHT	本文	LZC	SPIHT	本文
0.1	22.91	23.31	23.58	20.36	20.46	20.60	26.32	26.72	26.78
0.25	26.10	26.68	26.94	21.63	21.97	21.98	28.60	29.08	29.18
0.5	29.48	30.06	30.39	23.55	23.87	23.89	30.75	31.35	31.35

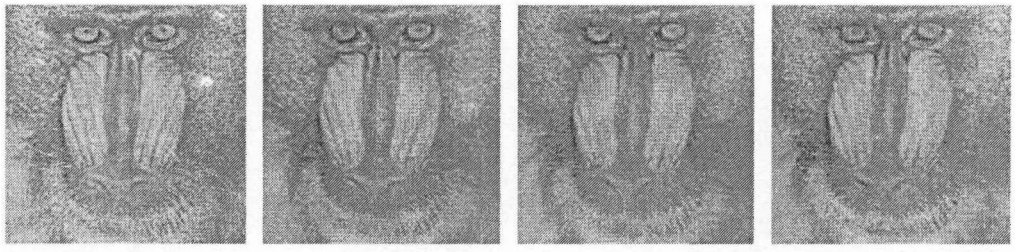

图 2 不同压缩算法在码率 0.2bpp 下对 Mandrill 的主观性能比较
(a) 为原始图像, (b), (c), (d) 分别为 LZC, SPIHT 和本文算法的重建图像

5 快速近似搜索算法

由于输出 $S_n(T)$ 必须搜索集合 T 中的所有元素，可见它的时间复杂度等于集合 T 中元素的个数。对于高层的节点其后代集合的元素很多，因此计算函数 $S_n(\bullet)$ 成为制约基于 EZW 的编解码器速度的主要原因。如何提高编解码器的速度是这一节的主要问题。

考虑到小波系数的分级树状结构和基于集合划分算法的特点，原始的 SPIHT 算法采用了预先生成一个最大图 M，其中 $M(i,j)=\max\{|C(k,l)|;(k,l)\in D(i,j)\}$，记录了节点 (i,j) 的所有后代中系数模值的最大值。因此 $S_n(D(i,j))=\begin{cases}1, & M(i,j)\geq 2^n\\ 0, & M(i,j)<2^n\end{cases}$，其时间复杂度降为固定的常数。它实际上是用空间复杂度来换取时间复杂度。对于 512×512 的彩色图像而言，它需要额外的存储空间 $256\times 256\times 3\times 2\text{Byte}/1\text{k}=384\text{kB}$。

本文提出了一种近似的搜索方法。计算 $S_n(D(i,j))$ 时只搜索节点 (i,j) 最上层的两代，即儿子节点和孙子节点，对于 $S_n(L(i,j))$ 只要搜索其孙子节点即可。显然这种近似方法不能用于无损压缩，但对于有损压缩，尤其是高压缩比的编码而言，这个近似搜索方法显得非常有效。它是基于如下的统计结果而提出来的。在计算最大图 M 时，从最大值节点出现位置的统计结果可以发现，即使除去最底的两层系数 (它们只有儿子或孙子节点)，绝大多数的最大值仍然是包含在儿子节点以及孙子节点的范围内，如表 2 所示。当去除这些不在这个范围内的最大值点，图像质量只是略微有所降低。因此近似搜索算法等效于预先将图像作有限的降质处理。表 3 以平均每个节点搜索的次数来比较了使用近似算法对编码速度的影响，图 3 显示了使用近似搜索方法重建的图像。由此可见，使用近似搜索方法可以有效地提高编码速度，而且在高压缩比的情况下，可以避免多级树的深层划分所造成的额外比特分配，在一定程度上可以增强图像的质量。

表 2 最大图中最大值点所在位置的统计数据

	儿子和孙子节点	其它后代节点
Lenna(256×256)	4081(99.6%)	15(0.4%)
Mandrill(512×512)	16267(99.3%)	117(0.7%)
Goldhill(512×512)	16308(99.5%)	76(0.5%)

表 3　使用近似搜索后编码速度的比较 (Mandrill)

	全部搜索		近似搜索	
码率	平均次数	PSNR(dB)	平均次数	PSNR(dB)
0.05	7.74	19.81	0.20	19.82
0.1	8.20	20.60	0.37	20.62
0.25	9.07	21.98	0.80	21.81
0.5	9.77	23.88	1.29	23.70
1	10.49	27.13	1.94	26.90
2	11.16	31.98	2.59	31.46

6 结　论

本文在研究了 SPIHT 和 LZC 算法的基础上，提出了一种改进的 SPIHT 算法。在使用少量的存储空间条件下，达到甚至超过 SPIHT 的性能。同时，本文针对有损图像压缩提出了一种快速的近似搜索方法，在不增加任何存储空间和细微的图像质量损失的条件下，大大地加快了编码速度，在一定程度上提高了编码性能。

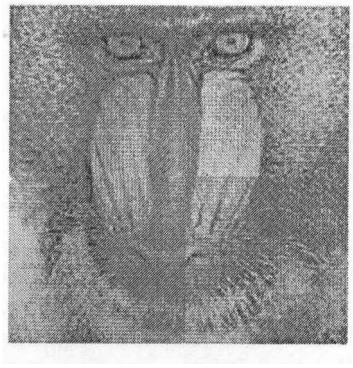

图 3　采用近似算法 0.2bpp 码率下的重建图像

参　考　文　献

[1] J. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, IEEE Trans. on Signal Processing, 1993, 41(12), 3445–3462.

[2] A. Said, W. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees, IEEE Trans. on Circuit and System for Video Technology, 1996, 6(3), 243–250.

[3] 吴乐南，数据压缩，北京，电子工业出版社，2000 年 6 月，146–148.

[4] W. K. Lin, N. Burgress, Listless zerotree coding for color image, In 32nd Asilomar Conference on Signals, Systems and Computer, Monterey, CA, Nov. 1998, 231–235.

MODIFIED SPIHT ALGORITHM

Li Honggang　　Wang Qiao　　Wu Lenan

(*Department of Radio Engineering, Southeast University, Nanjing* 210096, *China*)

Abstract　Among the wavelet image coding algorithms, SPIHT is the most well-known coding algorithm because of its outstanding preformance. But it needs too many memories for hardware implementation. LZC reduces the requirement of memory as well as its preformance. In this paper, SPIHT algorithm is modified by using the idea of LZC to reduce the momery and preserve its high performance. Also an approximated algorithm is proposed to accelerate the algorithm with insignificant loss.

Key words　Embeded Zerotree Wavelet, Set Partitioning in Hierarchical Trees, Listless Zerotree Coder

李洪刚：　男，1974 年生，博士生，研究方向图像及视频处理，小波变换及其应用.

王　桥：　男，1966 年生，博士，教授，《IEEE 信息论会刊》、《Zentralblatt Math.》等杂志评审员，现主要从事微局部分析及其在信号分析与多媒体通信中的应用.

吴乐南：　男，1952 年生，博士，教授，现主要从事多媒体信号处理，数字广播技术等方面的研究.