

X 树图像编码¹

蔡灿辉 丁润涛

(天津大学信息工程学院 天津 300072)

摘 要 该文提出一种新的小波图像编码结构——X 树。X 树是除根节点外其余的节点均为 0 的四叉树，它是零树的一种推广。在 X 树的基础上，文中提出了普通 X 树图像编码和嵌入式 X 树图像编码两种编码方案。实验表明该文提出的方案的编码效率明显高于零树 (EZW) 和栈-游程 (Stack-run) 编码。

关键词 小波编码，图像压缩，X 树图像编码，嵌入式编码，栈-游程编码

中图分类号 TN919.8

1 引 言

小波编码提供了一种非常有效的图像编码方案，对极低速率的图像编码更是如此。采用类似 DCT 编码的方法，栈-游程 (Stack-run) 编码把各子带中的小波系数组织起来，形成一种高效、简单的编码方案^[1]。零树 (EZW)^[2] 和在此基础上发展起来的 SPIHT 算法^[3] 根据小波变换各子带间能量相关、功率谱递减的特点，采用空间四叉树来表示小波系数，具有很好的编码效率。空-频量化 (SFQ)^[4]，估值量化 (EQ)^[5] 和晶格量化^[6] 采用不同的量化方法，对小波系数进一步处理，收到良好的效果。采用基于图像结构和条件熵等更复杂的算法^[7]，可以进一步提高压缩效果。虽然提高算法的复杂度可以提高编码压缩效率，但是在实际应用中，往往希望能采用更简单有效的算法。

零树编码按照四叉树结构组织小波系数，与 JPEG 中的 DCT 编码方案相比较，DCT 编码是一维的游程编码而零树编码是二维的游程编码。由于零树编码合理地利用了小波变换图像各子带间系数相关、功率衰减的特点，其效果要比 DCT 编码方案好得多。然而，零树结构并不是最佳的编码结构，对零树结构优化可以进一步提高编码效率。

本文提出的 X 树就是对零树结构的一种优化。与零树相同，X 树的所有子孙节点都取零值，但它的根节点却可取任意值。利用 X 树对 EZW 和 Stack-run 编码进行改造，本文提出了嵌入式 X 树图像编码和普通 X 树图像编码两种新的小波变换图像编码方案。前者是在 EZW 编码中用 X 树代替零树的结果，后者是在 Stack-run 编码中用 X 树代替零游程的结果。通过这种改进，较好地提高了上述两种算法的性能。本文余下部分安排如下：第 2 节介绍 X 树编码结构；第 3 节介绍嵌入式 X 树图像编码方案；第 4 节介绍普通 X 树图像编码方案；第 5 节给出实验结果及分析；第 6 节是本文的总结。

2 X 树编码结构

Shapiro 提出的 EZW 图像编码大大提高了小波变换的图像编码效率。然而，零树结构并非最佳编码结构，合理调整树结构可进一步提高编码效率。X 树编码就是通过对零树结构进行合理调整而产生的一种编码结构。与零树相同，X 树也是一棵生长在各子带间的四叉树，每一个节点对应于小波变换图像的一个像点。在金字塔分解中，上层子带中的节点称为父节点，下层子带中相应的节点 (通常有 4 个) 称为其子节点 (图 1)。令 $N(i, j)$ 表示第 (i, j) 节点的值，对给定的门限值 T ，若 $|N(i, j)| < T$ ，则记为 $N(i, j) = 0$ (这里， i, j 编号是沿着金字塔顶向金

¹ 2001-10-10 收到，2002-01-14 改回
福建省自然科学基金资助项目 (编号 A0110013)

字塔底部的方向进行的)。令 $C(i, j)$ 为节点 (i, j) 的子节点的集合, $Q(i, j)$ 表示节点 (i, j) 所有子孙节点的集合, 则

$$C(i, j) = \{(2i, 2j), (2i + 1, 2j), (2i, 2j + 1), (2i + 1, 2j + 1)\} \quad (1)$$

通过 (1) 式可递归定义 $Q(i, j)$ 。令 $T(i, j)$ 代表空间四叉树, 若 $T(i, j)$ 满足

$$N(p, q) = 0 | (p, q) \in Q(i, j) \quad (2)$$

则称 $T(i, j)$ 为 X 树, 记为 $X(i, j)$ 。为了减少符号定义, 在不会引起混乱的情况下, 我们也用 $X(i, j)$ 表示 X 树根节点的值。相应地, 零树的定义可表示为

$$\left. \begin{aligned} N(i, j) &= 0 \\ N(p, q) &= 0 | (p, q) \in Q(i, j) \end{aligned} \right\} \quad (3)$$

比较 (2) 式与 (3) 式, 可以看出 X 树与零树的区别仅在于根节点。零树的根节点必须为 0, 而 X 树的根节点可取任何值。当根节点 $X(i, j) > 0$ 时, 称该 X 树为 P 树, 根节点 $X(i, j) < 0$ 时, 称为 N 树。如果根节点 $X(i, j) = 0$, X 树就退化为零树。

提出 X 树的目的是为了解决 EZW 算法对满足: $N(i, j) \neq 0, N(p, q) = 0 | (p, q) \in Q(i, j)$ 的四叉树编码效率低的问题。从小波分析的角度上讲, 零树代表图像的平滑部分。当图像足够平滑时, 就会产生零树。零树出现的位置取决于图像的平滑程度和门限 T 的选择。在图像平滑区域, 通常父节点的值大于子节点, 即 $|N(i, j)| > |N(p, q)|, (p, q) \in C(i, j)$ 。当 $|N(i, j)| < T$ 时, 节点 (i, j) 及其子孙节点的集合 $Q(i, j)$ 形成零树 (或零树的一部分)。但是当门限值较低时, 譬如说 $|N(i, j)| > T > |C(i, j)|$ 时 (为了简单起见, 这里我们用 $C(i, j)$ 代表该集合中绝对值最大的元素), $(i, j) \cup Q(i, j)$ 已不再是零树, 而是 P 树或 N 树。如果采用零树结构来表示, 就必须用四棵树和一个独立节点来表示。采用 X 树结构, 只用一棵树就能表达该部分图像, 从而提高了编码效率。

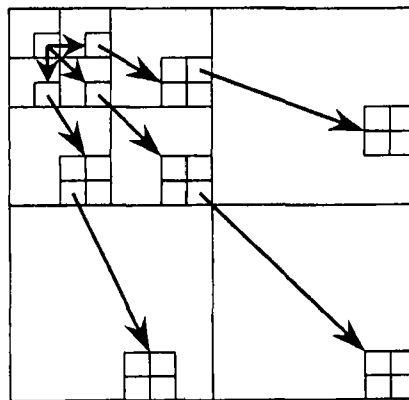


图 1 子带间四叉树示意图

类似于零树的建立过程, 生成 X 树的第一步是对系数进行重要性检查。对给定的门限值 T , 如果 $N(i, j)$ 满足 $|N(i, j)| < T$, 则该系数是不重要的, 令 $N(i, j) = 0$ 。否则该节点是重要的, $N(i, j)$ 取原系数值。如果节点 (i, j) 的子孙节点均为 0 (不妨记为 $Q(i, j) = 0$), 则 $N(i, j)$

属于某 X 树。如果还满足 $N(i, j) \neq 0$ 或 $C(\lfloor i/2 \rfloor, \lfloor j/2 \rfloor)$ 不全为 0, 则 $N(i, j)$ 是 X 树的根节点。X 树的生成沿着最小子带 (金字塔顶) 到最大子带 (金字塔底部) 的方向进行。考虑到在二维小波变换系数中, 低通滤波方向 (L 方向) 比高通滤波 (H 方向) 更能体现出图像的趋势, 因此对每个 HL 和 LH 子带内系数的扫描均沿着 L 方向进行, 以获得较好的编码效率 (见图 2)。在 X 树编码中, 一个节点可取正、负或者 0 值。同时, 该节点可以是 X 树的根节点、独立节点或根节点的子孙节点。为了方便描述, 定义 X 树符号集如下:

(1) {PROOT, NROOT, ZROOT} 代表 P 树, N 树和零树的根节点;

(2) {POS, NEG, ZERO} 表示正数、负数和零等 3 种独立节点;

(3) {DESC} 代表根节点的子孙节点, {XSIG} 代表嵌入式 X 树编码算法中前几遍扫描已编码过的非零节点。

由于 X 树可由其根节点完全描述, DESC 不输出任何码字, 而 XSIG 则因已编码过, 也无需输出代码。在 X 树生成过程中只产生和输出上述第一、第二类符号的代码。在金字塔底部 3 个最大的子带中, 根节点退化为独立节点, 因此只需对第二类符号进行编码。考虑到这时非零节点多数取值为 “+1” 或 “-1”, 可以把这两个符号代替第一类符号进行编码, 也可以对第二类符号独立编码。我们在普通 X 树编码采用第一种方式编码, 而嵌入式 X 树编码则采用第二种方式。

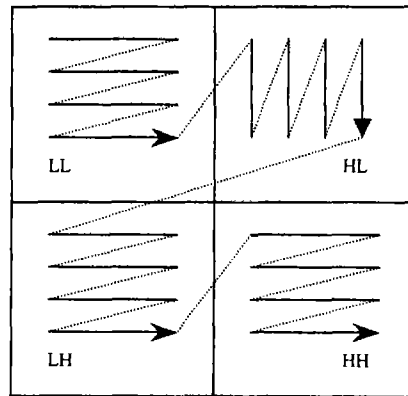


图 2 小波系数扫描顺序

3 嵌入式 X 树图像编码

如前所述, 嵌入式 X 树图像编码是 EZW 的扩展, 因此, 本算法也采用逐次逼近的方法实现嵌入式编码。在完成小波变换图像的量化之后, 我们选择一组门限 T_0, \dots, T_{N-1} , 按上述算法对每个 T_i 生成 X 树。这里, $T_i = T_{i-1}/2$, 且 T_0 满足 $(\text{MAX}|N(i, j)|) < 2T_0$ 。X 树生成过程中产生码字送到两个输出流: XTREE 和 SDATA。其中 XTREE 用来保存 X 树结构, SDATA 用来保存非 0 数据的正负号。如果该节点为零树根, $(i, j) \in \{\text{ZROOT}\}$, 只需输出代码 ‘0’ 到 XTREE; 如果该节点为非零树根, $(i, j) \in \{\text{PROOT, NROOT}\}$, 则: (1) 送代码 ‘1’ 到 XTREE; (2) 把相应的符号代码 ‘1’ 或 ‘0’ 送到 SDATA; (3) 把 $|X(i, j)| - T_i$ 送队列 COEF, 扣除 T_i 原因是相应位的信息已包含在符号位中, 无须另行编码; (4) 标注该节点为 XSIG 节点, 以后不再对其编码。如果被扫描的节点是独立 0 节点 $(i, j) \in \{\text{ZERO}\}$, 只需输出代码 ‘2’ 到 XTREE; 反之, 如果该节点是独立非 0 节点, $(i, j) \in \{\text{POS, NEG}\}$,

则 (1) 输出代码 ‘3’ 到 XTREE; (2) 根据该节点的值是正数还是负数, 将其符号代码送到 SDATA; (3) 把 $|N(i, j)| - T_i$ 送队列 COEF; (4) 标注该节点为 XSIG 节点。如前所述, X 树的子孙节点 DESC 和前面编码过的节点 XSIG 不产生代码输出。

考虑到在 3 个最大的子带中, 节点都是叶子节点, 因此可以只在 SDATA 中输出 0 与非 0 节点信息, 而不需要输出代码到 XTREE。当被扫描的节点为 0 节点时, 输出代码 ‘0’ 到 SDATA, 反之则输出代码 ‘1’ 到 SDATA。对非 0 节点, 再把它正、负号代码 ‘1’ 或 ‘0’ 输出到 SDATA。

与零树编码过程一样, 生成 X 树之后, 还必须对先前各次扫描产生的非 0 系数进行编码。对于 COEF 队列中的每一个系数 c_j , 如果满足 $c_j > T_i$, 则 (1) 令 $c_j = c_j - T_i$; (2) 输出代码 ‘1’ 到 SDATA。反之, 则只输出代码 ‘0’ 到 SDATA。最后, 把 XTREE 和 SDATA 送到算术编码器^[8]进行熵编码。实验表明, 如果对 SDATA 中的数据与符号位分别编码, 可以提高压缩比。

解码算法与上述编码过程相类似。由于有损压缩的重建系数可能出现在重建区间的任何位置, 为了减少计算复杂度, 本算法选用重建区间的中点作为重建值。采用更复杂的计算方法来估计重建值, (如 SPIHT 算法中采用的基于 DPF 的估值方法), 可以使压缩效率有所提高。

4 普通 X 树图像编码

嵌入式 X 树图像编码具有随时截断输出代码不会产生附加失真的优点, 但需要多次扫描生成 X 树, 具有一定的计算复杂性。为此, 本文提出第二种算法——普通 X 树图像编码。该算法只需一次性扫描建立 X 树即可完成编码过程, 具有简单、高效等特点。但是, 在这种算法中, 涉及众多独立节点的处理, 因此不能简单地采用 $N = 1$ 的嵌入式 X 树图像编码算法来实现。为了减少运算, 提高编码效率, 我们把 Stack-run 算法的基本思想引入普通 X 树图像编码。亦即把小波变换系数中的非 0 数据当作 ‘栈’, 用 X 树描述 0 游程, 并采用类似 Stack-run 编码的方法进行编码。

首先, 我们采用带死区的均匀量化器对小波变换图像进行量化。设量化步长为 Δ , 死区参数为 α , 则死区宽度为 $2\alpha\Delta$ 。若节点 (i, j) 的小波系数的绝对值 $|w_{i,j}| < \alpha\Delta$, 量化器输出为 $x_{i,j} = 0$, 否则, 量化器输出为 $x_{i,j} = [(w_{i,j} - \text{sgn}(w_{i,j})\alpha\Delta)/\Delta]$ 。在完成小波变换图像的量化之后, 可按本文第 2 节所述的算法生成 X 树。在 3 个最大的子带中, 由于根节点已退化为独立节点, 而且这些子带中的非零节点多数取值为 “+1” 或 “-1”, 因此可借用 X 树根节点的代码把它们融合在 X 树编码中。

与嵌入式 X 树图像编码算法不同, 本算法只生成单个输出流: XTREE。只有 X 树结构被送到 XTREE, 非 0 系数及符号一起送到队列 COEF。与嵌入式 X 树图像编码中 XTREE 流的生成过程一样, 如果该节点为零树根, $(i, j) \in \{\text{ZROOT}\}$, 输出代码 ‘0’ 到 XTREE; 反之, 如果 $(i, j) \in \{\text{PROOT}, \text{NROOT}\}$, 则送代码 ‘1’ 到 XTREE, 并把该根节点的数值 $X(i, j)$ 送 COEF 队列。如果被扫描的节点是独立 0 节点 $(i, j) \in \{\text{ZERO}\}$, 输出代码 ‘2’ 到 XTREE; 反之, 如果 $(i, j) \in \{\text{POS}, \text{NEG}\}$ 输出代码 ‘3’ 到 XTREE, 该节点的数值 $N(i, j)$ 送 COEF 队列。

由于系数的代码是长度不一的变长码, 需要在各系数间进行隔离。这里是用各系数的符号作间隔符并令正号的代码为 ‘2’, 负号的代码为 ‘3’。与嵌入式 X 树图像编码算法类似, 队列中系数的最高位总为 1, 不必产生码字。为了进一步提高编码效率, 可把 XTREE 流送到算术编码器进行熵编码。

类似地, 我们可以得出解码算法。与嵌入式 X 树编码算法一样, 这里也采用重建区间的中点作为重建值。

5 实验结果与分析

我们用 3 幅常用的 512×512 8 位灰度图像 “Lena”, “Barbara” 和 “Goldhill” 对上述的 2 种算法进行测试。本实验采用 5 级的塔式分解, 小波变换滤波器采用 7/9 双正交滤波器^[9], 普通 X 树图像编码算法的死区参数选为 $\alpha = 0.8$ 。表 1 列出了 EZW, Stack-run, SPIHT 和本文提出的 2 种算法的 PSNR 值。其中, PSNR 定义为

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\text{MSE}} \right) \quad (\text{dB})$$

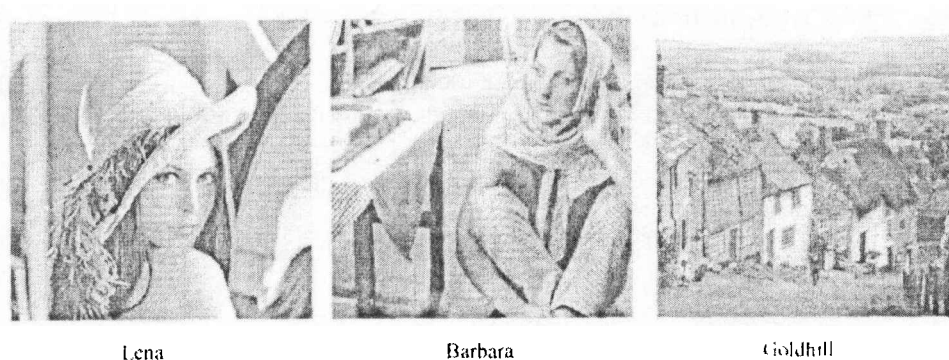
这里 MSE 为原始图像与重建图像的均方误差。图 3 给出了采用 SPIHT 和本文提出的 X 树编码算法的重建图像。由于两种 X 树编码的重建图像基本相同, 考虑到文章的篇幅问题, 图中只给出普通 X 树编码的重建图像。表 1 中 Stack-run、EZW 和 SPIHT 的实验结果分别来自参考文献 [1-3]。其中 N/V 表示原文没提供该实验数据。

表 1 对图像编码的 PSNR 值

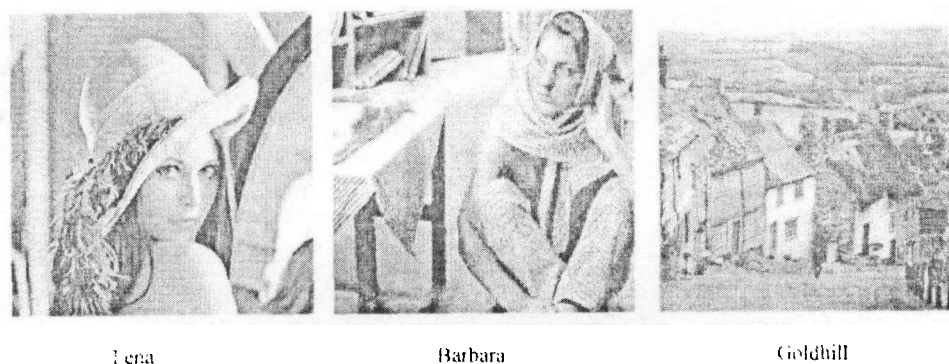
算法	编码图像	码率 (bpp)			
		0.125	0.25	0.50	1.00
嵌入式 X 树	Lena	30.94	33.96	37.05	40.23
	Barbara	24.92	27.63	31.21	36.04
	Goldhill	28.45	30.49	33.03	36.38
普通 X 树	Lena	31.00	34.09	37.13	40.17
	Barbara	24.88	27.78	31.48	36.29
	Goldhill	28.44	30.52	33.14	36.52
EZW	Lena	30.23	33.17	36.28	39.55
	Barbara	24.03	26.77	30.53	35.14
	Goldhill	N/V	N/V	N/V	N/V
Stack-run	Lena	N/V	33.58	36.67	N/V
	Barbara	N/V	27.32	30.94	N/V
	Goldhill	N/V	30.30	32.92	N/V
SPIHT	Lena	31.11	34.14	37.21	40.46
	Barbara	N/V	27.40	31.25	N/V
	Goldhill	28.47	30.56	33.12	36.55

表 1 的数据表明 SPIHT 算法对图像 “Lena” 的压缩效率略高于本算法; 反之, 对图像 “Barbara”, 本算法压缩效果要比 SPIHT 算法略好些; 而本算法和 SPIHT 算法对图像 “Goldhill” 的压缩效果基本一样。从上述的实验结果, 我们可以看到, 本算法的压缩效果明显优于 EZW 和 Stack-run 算法, 与 SPIHT 算法基本相同。虽然普通 X 树编码在树结构方面比嵌入式 X 树编码产生较少的码字, 但在非 0 系数编码上开销较大, 所以两种 X 树图像编码的效率也基本相同。嵌入式 X 树编码方案具有代码随意截断不会产生附加失真的优点, 但计算量要比普通 X 树编码方案大得多。

从图 3 可以看出, 采用本文提出的算法重建的 “Barbara” 图像中桌布、裤子和围巾上的纹理要比采用 SPIHT 算法重建的图像要清楚些。类似地, 采用 X 树编码算法重建的 “Goldhill” 上屋顶的瓦片和 “Lena” 中草帽的纹理也要比采用 SPIHT 重建的图像上要清楚些。必须指出, 在某些场合, 如果重建图像的部分细节出现而相关的另一些细节却没有出现时, 其感官效果可能还不如不出现这些细节。例如, Barbara(图 3(b)) 中右胳膊上面的一些花纹实际上是裤子的纹理, 但很可能被误认为失真或干扰。



(a) 采用 SPIHT 算法重建的图像 (0.25bpp)



(b) 采用普通 X 树算法重建的图像 (0.25bpp)

图 3 X 树算法与 SPIHT 算法重建图像的比较

6 结 论

本文提出了一种新编码结构——X 树, 并在此基础上提出了普通 X 树图像编码和嵌入式 X 树图像编码两种编码方案。X 树是一棵除根节点外其余的节点均为 0 的子带间的四叉树。它是零树的一种推广。一方面, 用 X 树代替 EZW 算法中的零树来描述小波变换图像的系数就产生了嵌入式 X 树图像编码算法, 另一方面, 用 X 树代替 Stack-run 中的零游程, 形成了普通 X 树图像编码方案。基于 X 树的编码算法的压缩效率与 SPIHT 算法基本相同, 但 X 树编码算法更简单、计算复杂度更小。由于 X 树只不过是根节点可取非 0 值的零树, 许多基于零树的改进算法可以直接用于 X 树编码。嵌入式 X 树图像编码的计算复杂度与 EZW 相同, 普通 X 树图像编码与 Stack-run 相似。

参 考 文 献

- [1] M. Tsai, J. Villasenor, F. Chen, Stack-run image coding, IEEE Trans. on Circuits Syst., Video Technol., 1996, 6(5), 519-521.
- [2] J. M. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, IEEE Trans. on Signal Processing, 1993, 41(12), 3445-3463.
- [3] A. Said, W. A. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees, IEEE Trans. on Circuits Syst., Video Technol., 1996, 6(3), 243-250.

- [4] Z. Xiong, K. Ramchandran, M. T. Orchard, Space-frequency quantization for wavelet image coding, *IEEE Trans. on Image Processing*, 1997, IP-6(5), 677-693.
- [5] S. LoPresto, K. Ramchandran, M. T. Orchard, Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework, *DCC'97, Snowbird, UT*, 1997, 221-230.
- [6] R. L. Joshi, *et al.*, Comparison of different methods of classification in subband coding of images, *IEEE Trans. on Image Processing*, 1997, IP-6(11), 1473-1486.
- [7] C. Chrysafis, A. Ortega, Efficient context-based entropy coding for lossy wavelet image compression, *DCC'97, Snowbird, UT: 1997*, 241-250.
- [8] I. H. Witten, R. M. Neal, J. G. Cleary, Arithmetic coding for data compression, *Commun. ACM*, 1987, 30(6), 520-540.
- [9] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, Image coding using wavelet transform, *IEEE Trans. on Image Processing*, 1992, IP-1(2), 205-221.

THE X-TREE IMAGE CODING

Cai Canhui Ding Runtao

(School of Electron. Info. Eng., Tianjin University, Tianjin 300072, China)

Abstract This paper presents a new scheme of wavelet image coding, X-tree. An X-tree is a spatial hierarchical tree whose all nodes are 0 except the root node. It is the generalization of the zero tree. Based on the X-tree scheme, two new coding algorithms, the embedded X-tree approach and the general X-tree approach are proposed in the paper. Experimental results show that the coding efficiency of the proposed algorithms are obviously higher than those of the stack-run and the EZW algorithms.

Key words Wavelet coding, Image compression, X-tree image coding, Embedded coding, Stack-run coding

蔡灿辉: 男, 1954 年生, 在职博士生, 主要研究方向为图像处理.

丁润涛: 男, 1938 年生, 博士生导师, 主要研究方向为图像处理.