

# 基于子带二叉树合并的图像位平面编码

李建欣 朱日宏 丁绪星

(南京理工大学光电学院, 南京 210094)

**摘要** 利用图像经过小波变换后系数的分布特点, 能够有效地进行图像编码。为了充分利用小波系数的带内相关性和带间相关性, 提出了一种有效的位平面编码算法。该算法将各个子带二叉树合并成一棵扫描树, 然后再根据扫描树提供的扫描信息进行位平面编码。该算法能够有效地对纹理丰富的图像区域进行编码, 比 EZW 和 SPIHT 算法具有更好的编码效果。除此之外, 该算法还具有实现简单、执行速度快和消耗存储空间资源少等特点, 降低了硬件实现的难度, 具有很强的实用性。

**关键词** 图像编码 位平面编码 扫描树 小波变换

中图分类号: TN919.81 文献标识码: A 文章编号: 1006-8961(2006)04-0458-07

## Image Bitplane Coding Based on Combination of Subband Quadrees

LI Jian-xin, ZHU Ri-hong, DING XU-xing

(School of Electronic Engineering and Photoelectric Technology, Nanjing University of Science & Technology, Nanjing 210094)

**Abstract** Image can be encoded effectively by using the spatial distribution characteristic of wavelet coefficients. In order to make full use of the intraband correlation and interband correlation of wavelet coefficients, an effective image coding algorithm is proposed. In the algorithm, subband quadrees are combined to a scanning tree, and then bitplanes are encoded with the information provided by the scanning tree. The algorithm has better coding performance than EZW and SPIHT for it effectively encodes the image areas which are rich in texture. In addition, the algorithm is also characterized by features of easy realization, fast execution and little memory consumption. So it reduces the difficulty of hardware realization and has good practicability.

**Keywords** image coding, bitplane coding, scanning tree, wavelet transform

## 1 引言

基于小波变换的嵌入式编码算法能够提供分辨率可伸缩和信噪比可伸缩的特性, 能够很好地满足实际应用场合的要求, 它是近年来图像编码研究的一个热点。嵌入式的图像编码算法有许多种, 例如 EZW (embedded zerotree wavelet)<sup>[1]</sup>、SPIHT (set partitioning in hierarchical trees)<sup>[2]</sup> 和 JPEG2000<sup>[3]</sup> 的核心编码算法 EBCOT (embedded block coding with optimized truncation)<sup>[4]</sup> 等。其中, EZW 和 SPIHT 采用零树嵌入

编码的方法, 有效地利用了小波系数的带内相关性和带间相关性; EBCOT 则采用嵌入式块编码的方式, 充分利用了小波系数的带内相关性。这些编码算法能取得很好的编码效果, 但存在大量繁琐的编码操作, 影响了执行效率。目前很多的编码算法虽然能够取得很高的压缩率, 但算法的实现却非常复杂, 不利于实用化。在保持高压缩率的情况下尽量简化编码算法并使其实用化的研究越来越重要。近来, 一些有特色的位平面编码算法如 EZBC<sup>[5]</sup> (embedded coding using zeroblocks of wavelet coefficients and context modeling), 利用子带二叉树分裂的扫描算法和上下

收稿日期: 2004-04-16; 改回日期: 2005-06-13

第一作者简介: 李建欣(1977 - ), 男, 现为南京理工大学光学工程专业博士研究生。主要研究方向为图像、视频编码与传输。E-mail: ljx@vip.163.com

文模型(context)的合理设计,取得了相当好的编码效果。这种基于四叉树分裂扫描方式的编码方法为图像编码提供了一种新的思路,适当运用这种编码结构可以使算法的实现变得更加简单有效。

充分利用小波系数的带内、带间相关性能够有效地提高小波编码器的压缩率。因此,小波编码器的设计应该尽量使编码策略和小波系数的分布特点紧密联系。但基于四叉树分裂扫描方式的位平面编码算法一般对各个子带独立编码,虽然利用了小波系数的带内相关性,但却忽略了带间相关性。有的算法为了要利用小波系数的带间相关性,设计了复杂的上下文模型。在一定程度上,上下文模型提高了图像的编码质量,但它却使位平面编码操作更加复杂。为了降低算法的复杂度,而且充分利用小波系数的相关性,本文在分析小波系数的空间分布结构的基础上,把所有小波子带中的四叉树进行合并,生成一棵扫描树,然后再根据扫描树提供的扫描信息进行位平面编码。实验结果表明了扫描树的构造不仅使小波系数的扫描操作更加简单、高效和紧凑,而且能够提高图像的编码质量。

## 2 小波系数的分布特点

图像在空间域上是高度相关的,经过小波变换后系数能量进行了集中,并且在空间分布上具有结构相似性,如图 1 所示。小波域上某一分解级的系数与下一分解级相应位置上的几个系数具有一定的相关性,从而引入空间方位树的概念来描述这种系数分布特征。空间方位树的根节点为低频子带中的系数,它的 3 个子节点分别是同一分解级上 3 个高频子带中相应位置上的系数,除了低频子带,每个高频子带中的系数都有 4 个子节点,即下一分解级相应位置上的 4 个系数,如图 2 所示。图中的 A 表示

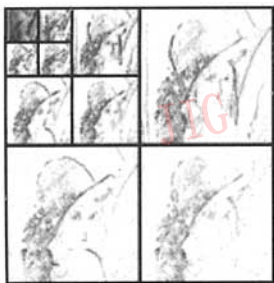


图 1 经过三级小波分解后的 Lena 图像

Fig. 1 Lena image of three level wavelet transform

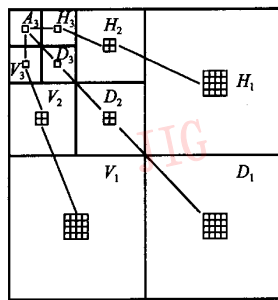


图 2 小波域中的空间方位树

Fig. 2 Spatial orientation tree in wavelet domain

低频子带,  $H$  表示水平方向高频子带,  $V$  表示垂直方向高频子带,  $D$  表示对角方向高频子带。

对于高频子带,大部分系数的幅值很小,能量主要集中在边缘特征突出的区域。利用四叉树分裂的系数扫描方法<sup>[5]</sup>,可以有效地利用小波系数的“零块”<sup>[5,6]</sup>分布特点。目前,基于“零块”特点进行编码的算法有 EZBC、SPECK (set partitioning embedded block)<sup>[6]</sup>和 SWEET<sup>[7]</sup>等。在 EZBC 算法中,每个子带都构造相应的子带四叉树,为该子带的扫描编码提供参考信息。子带四叉树的父节点为下一节点层同一位置上的 4 个子节点的最大值。子带四叉树最低节点层的节点为该子带系数的绝对值,这一节点层记为  $l_{\max}$ ;而最高节点层只有一个节点,该节点是子带四叉树中最大的值,这一节点层记为  $l_{\min}$ 。子带四叉树的节点层  $l$  满足  $l_{\min} \leq l \leq l_{\max}$  的关系。子带四叉树的节点层数  $l_0$  定义为  $L_0 = l_{\max} - l_{\min}$ ,而节点层的深度定义为  $d_0 = l - l_{\min}$ 。在编码某个子带的过程中,编码器从该子带四叉树的最高节点层开始往下扫描。如果节点是重要的,即该节点的绝对值大于或等于给定的阈值,则该节点能够进行分裂,编码器可以继续扫描下一节点层的 4 个子节点。如果节点是不重要的,即该节点的绝对值小于给定的阈值,则该节点不能进行分裂,编码器对该节点的所有子节点都不进行扫描。如图 3 所示,  $E$ 、 $F_0$  和  $F_2$  为重要的节点,则继续分裂;  $F_1$  和  $F_3$  为不重要的节点,则不再分裂。因此,不重要的节点能紧凑地表示一大块不重要的小波系数。由于四叉树的父节点为 4 个子节点的最大值,所以四叉树每层的节点图有很强的相似性,如图 4 所示。对于同一方向上的高频子带,其系数在空间分布上具有很强的相似性,它们的四叉树节点图也具有很强的相似性。对于不同方向上的高频子带,其系数在空间分布上也具有一定的

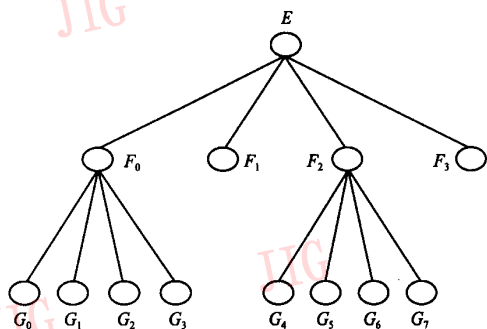


图 3 四叉树的分裂

Fig. 3 Splitting of quadtree

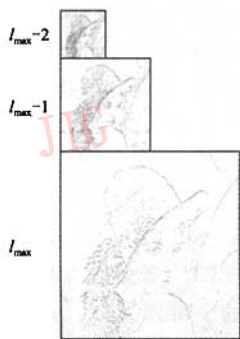


图 4 子带  $V_1$  的四叉树节点图

Fig. 4 Quadtree node map of subband  $V_1$

相似性,它们的四叉树节点图也同样具有一定的相似性。基于这些相似性,将各个子带四叉树进行合并处理。

### 3 四叉树的合并与位平面编码

#### 3.1 编码过程

基于小波变换的图像位平面编码算法一般有两个重要的部分,其中包括小波系数的扫描方式和重要性位平面编码。在基于四叉树分裂扫描方式的位平面编码算法中,四叉树的节点为小波系数扫描过程中四叉树的分裂提供判断信息,它隐含着小波系数的扫描路径。因此,四叉树的构造方式直接影响着小波系数扫描方式的高效性。本文算法在四叉树的构造过程中采用了两次四叉树合并处理,最后形成一棵扫描树。首先,把同一方向特性的高频子带四叉树进行合并,由于有 3 种方向特性的高频子带,因此得到 3 棵新的四叉树。然后,再把这 3 棵四叉树合并成 1 棵四叉树。最后,这棵四叉树和低频子

带组合成扫描树 ST (scanning tree)。扫描树生成后,本文算法再根据扫描树的节点信息进行小波系数扫描和重要性位平面编码。算法的主要过程如图 5 所示。

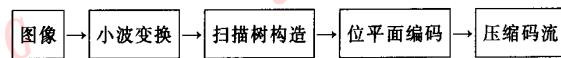


图 5 本文算法的编码过程

Fig. 5 Encoding course of the proposed algorithm

#### 3.2 四叉树的合并和扫描树的构造

图像经过  $L$  级小波分解后,形成一个低频子带、 $L$  个水平方向特性的高频子带、 $L$  个对角方向特性的高频子带和  $L$  个垂直方向特性的高频子带。把这些小波子带进行划分,让它们归属为 3 棵宏空间方位树  $MOT$  (macro orientation tree)。这里的宏空间方位树定义为若干棵空间方位树的集合。低频子带和  $L$  个水平方向特性的高频子带归属为  $MOT_1$ ,低频子带和  $L$  个对角方向特性的高频子带归属为  $MOT_2$ ,低频子带和  $L$  个垂直方向特性的高频子带归属为  $MOT_3$ 。图 6 中,图像进行三级小波变换后形成 10 个小波子带, $MOT_1$  包括  $A_3, H_3, H_2$  和  $H_1$ ,  $MOT_2$  包括  $A_3, D_3, D_2$  和  $D_1$ ,  $MOT_3$  包括  $A_3, V_3, V_2$  和  $V_1$ 。每棵宏空间方位树包括 1 个低频子带和  $L$  个高频子带,其中的高频子带分别生成相应的子带四叉树  $SQT$  (subband quadtree),这些子带四叉树在相同深度的节点层上进行合并,合并的结果为一棵新的四叉树  $QT$ ,其每层中的节点为相同  $d_0$  深度上所有子带四叉树相应节点的最大值。

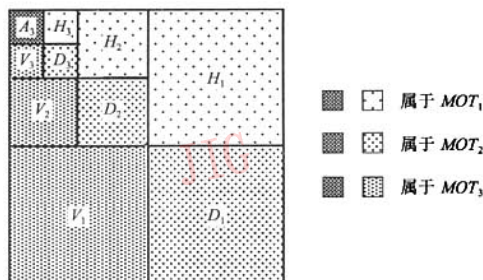


图 6 小波子带与宏空间方位树的关系

Fig. 6 Relation of wavelet subbands and macro orientation trees

图 7 说明了四叉树合并的途径,四叉树 1 只有一个节点层,节点数目为 1 个。

子带四叉树  $SQT$  可表示为

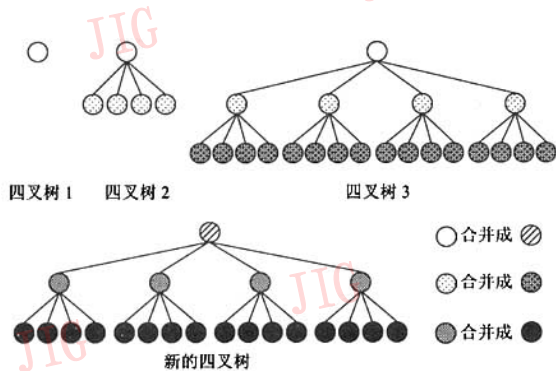


图 7 四叉树的合并

Fig. 7 Combination of quadrees

$$\begin{cases} \text{如果 } l = l_{\max}, \text{ 则} \\ SQT_l(i, j) = \text{abs}(C_{i,j}) \\ \text{如果 } l_{\min} \leq l < l_{\max}, \text{ 则} \\ SQT_l(i, j) = \max(SQT_{l+1}(2i, 2j), \\ SQT_{l+1}(2i, 2j+1), SQT_{l+1}(2i+1, 2j), \\ SQT_{l+1}(2i+1, 2j+1)) \end{cases} \quad (1)$$

其中,  $C_{i,j}$  表示坐标为  $(i, j)$  的小波系数。下标  $l$  表示子带四叉树的节点层。  $SQT_l(i, j)$  表示子带四叉树第  $l$  层的与小波系数分布平面上的坐标  $(i, j)$  对应的节点。

由第 1 次合并得到的 3 棵新四叉树 QT 为

$$QT_{1,l}(i, j) = \max(SQT_l^{H_L}(i, j), SQT_l^{D_L}(i, j), \dots, SQT_l^{V_L}(i, j)), l_{\min}^{(1)} \leq l < l_{\max}^{(1)} \quad (2)$$

$$QT_{2,l}(i, j) = \max(SQT_l^{H_L}(i, j), SQT_l^{D_L}(i, j), \dots, SQT_l^{V_L}(i, j)), l_{\min}^{(2)} \leq l < l_{\max}^{(2)} \quad (3)$$

$$QT_{3,l}(i, j) = \max(SQT_l^{H_L}(i, j), SQT_l^{D_L}(i, j), \dots, SQT_l^{V_L}(i, j)), l_{\min}^{(3)} \leq l < l_{\max}^{(3)} \quad (4)$$

其中,  $l_{\min}^{(n)}$  和  $l_{\max}^{(n)}$  分别表示第  $n$  棵宏空间方位树中的所有高频子带四叉树的最高节点层和最低节点层。  $SQT$  的上标  $H_L, D_L$  等表示该子带四叉树是属于哪一个子带的。  $QT_{n,l}(i, j)$  ( $n = 1, 2, 3$ ) 表示四叉树第  $l$  层的与小波系数分布平面上的坐标  $(i, j)$  对应的节点。对于不同的子带四叉树  $SQT$ , 它们的节点层数  $l_0$  是不相等的, 因此, 在式 (2) ~ (4) 中, 不存在的节点层可认为其节点内容为 0。

由式 (2) ~ (4) 得到的 3 棵四叉树同样具有子带四叉树的父、子节点关系, 即父节点为下一节点层同一位置上的 4 个子节点的最大值。因此, 在不同的阈值下, 新四叉树上不重要的节点能够表示宏空间

方位树中不同子带上的不重要小波系数。由于宏空间方位树中的高频子带具有相同的方向特性, 其小波系数在空间分布上具有很强的结构相似性, 并且四叉树的空间结构与子带系数的分布密切相关, 新生成的四叉树的节点能够有效地描述各个子带小波系数的分布情况, 从另一个角度说, 新四叉树上不重要的节点表示宏空间方位树中不同子带上的不重要小波系数是有效的。另外, 新四叉树中的一个节点同时反映各个高频子带中“一块”小波系数的信息, 因此, 新四叉树充分体现了小波系数的带间相关性和子带内小波系数的“零块”分布特点, 在大的阈值下, 新四叉树上不重要的节点能够紧凑地表示小波域中的不重要小波系数块, 这对于低码率编码是很适合的。

从式 (2) ~ (4) 可以看出, 要得到新四叉树必须先构造出各个高频子带四叉树, 这过程中存在较多的运算操作, 会影响编码器的执行效率, 因此需要简化新四叉树形成的过程。在实际的处理中, 可以通过一种效率更高的方法来构造新的四叉树。在此方法中, 新四叉树的最底层节点设置为空间方位树最底层系数的绝对值, 除了最底层之外, 每一层的节点设置为下一层中的 4 个子节点和宏空间方位树同一层中节点的绝对值的最大值, 由此可以得到以下关系:

$$QT_{n,l}(i, j) = \begin{cases} \text{abs}(MOT_{n,l}(i, j)), l = l_{\max}^{(n)}, n = 1, 2, 3 \\ \max[\text{abs}(MOT_{n,l}(i, j)), QT_{n,l+1}(2i, 2j), \\ QT_{n,l+1}(2i, 2j+1), QT_{n,l+1}(2i+1, 2j), \\ QT_{n,l+1}(2i+1, 2j+1)], l_{\max}^{(n)} - L + 1 \leq l < l_{\max}^{(n)}, \\ n = 1, 2, 3 \end{cases} \quad (5)$$

其中,  $L$  表示小波分解的级数。宏空间方位树的最高节点层为  $A_L$ , 这一层标记为  $(l_{\max} - L)$ , 最底层标记为  $l_{\max}$ 。在式 (5) 中, 高频子带合并的节点层从  $(l_{\max} - L + 1)$  开始。与式 (1) ~ (4) 相比, 式 (5) 中构造新四叉树的操作简化了很多。

通过式 (5) 得到的 3 棵四叉树  $QT_n$  ( $n = 1, 2, 3$ ) 的节点图具有很强的相似性, 把这 3 棵四叉树合并成 1 棵四叉树  $QT_0$  是可能的。因此, 再进行第 2 次四叉树的合并处理:

$$QT_{0,l}(i, j) = \max(QT_{1,l}(i, j), QT_{2,l}(i + h_l^{(0)} - l + 1, j), QT_{3,l}(i + h_l^{(0)} - l + 1, j - w_l^{(0)} - l + 1)) \quad (6)$$

$$l_{\max}^{(0)} - L + 1 \leq l \leq l_{\max}^{(0)}, l_{\min}^{(0)} = \min(l_{\min}^{(1)}, l_{\min}^{(2)}, l_{\min}^{(3)}),$$

$$l_{\max}^{(0)} = \max(l_{\max}^{(1)}, l_{\max}^{(2)}, l_{\max}^{(3)})$$

其中,  $h_l$  和  $w_l$  分别指小波分解级数为  $l$  时低频子带

的高度和宽度。经过式(6)的合并后,小波系数扫描算法所需要的存储空间大大减少。这是因为合并前的扫描算法需要 3 棵四叉树大小的存储空间,而合并后的扫描算法只需要 1 棵四叉树大小的存储空间。另外,3 棵新的四叉树的合并简化了编码时的扫描操作。四叉树  $QT_0$  的节点描述小波域中不同尺度、不同方向特性的高频子带中同一位置的系数块,因此,在特定阈值下,编码器能够一次性地扫描和编码这些系数块的重要比特位。与 EZW、SPIHT 等小波编码器的复杂扫描编码操作相比,利用  $QT_0$  的扫描编码操作具有简单、高效的特点。同时,这种扫描编码方式能够有效地对纹理细节丰富的局部图像区域进行编码,在低码率编码情况下,有利于提高图像的编码质量。

小波域中的低频子带是原图像的一个缩小的平滑版本,它在嵌入式图像编码中具有非常重要的作用。在扫描编码过程中,低频子带可作为本身的扫描参考信息。在完整的扫描编码过程中,需要两部分扫描参考信息: $QT_0$  和低频子带  $A_L$ 。把这两部分的信息整合成一棵扫描树  $ST$ (scanning tree),让它作为扫描编码操作的最终参考信息,可使得整个扫描编码过程更简单紧凑。

扫描树  $ST$  定义为

$$ST_l(i, j) = \begin{cases} QT_{0,l}(i, j), l = l_{\max}^{(0)} - L + 1, l_{\max}^{(0)} - L + 2, \dots, l_{\max}^{(0)} \\ \max(\text{abs}(A_L(i, j)), QT_{0,l+1}(i, j + w_l)), l = l_{\max}^{(0)} - L \end{cases} \quad (7)$$

### 3.3 位平面编码

目前大部分的小波编码器都可以产生嵌入特性的码流,这主要是通过位平面编码的方式来实现的。位平面编码是按照一定的系数扫描方式从小波系数的最高位平面开始编码的,随着编码的位平面越来越低,产生的编码流有越来越高的重建精度。

设坐标为  $(i, j)$  的小波系数定义为  $C_{i,j}$ 。令  $I \in \mathbf{Z}^2$  为坐标  $(i, j)$  的集合。对于一个阈值  $t(t > 0)$  和一个变量  $x(x \in \mathbf{R})$ ,定义阈值函数为

$$\delta(x, t) = \begin{cases} 1 & |x| \geq t \\ 0 & |x| < t \end{cases} \quad (8)$$

定义符号函数为

$$\text{sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \quad (9)$$

定义某一阈值  $t$  下不重要系数的坐标集合为

$$D(t) = \{(i, j) \in I \mid \delta(C_{i,j}, t) = 0\} \quad (10)$$

而阈值  $t$  下重要系数的坐标集合为

$$R(t) = I \setminus D(t) = \{(i, j) \in I \mid \delta(C_{i,j}, t) = 1\} \quad (11)$$

则位平面编码的一般过程可描述为

$$(1) \text{ 初始化 } t_n = 2^n, n = \lfloor \log_2(\max_{(i,j) \in I} |C_{i,j}|) \rfloor;$$

$$D(t_{n+1}) = \{(i, j) \in I\}.$$

(2) 对扫描路径中的每一个小波系数  $C_{i,j}$ ,

① 如果  $(i, j) \in D(t_{n+1})$ , 编码  $\delta(C_{i,j}, t_n)$ ;

若  $\delta(C_{i,j}, t_n) = 1$ , 编码  $\max(\text{sign}(C_{i,j}), 0)$ 。

这一过程称为重要性扫描通道编码(significant scanning pass coding)。

② 如果  $(i, j) \in R(t_{n+1})$ , 编码  $C_{i,j}$  的第  $n$  个位。

这一过程称为幅度细化通道编码(magnitude refinement pass coding)。

(3)  $n = n - 1$ , 返回到步骤 2。

在上述编码过程中,随着位平面  $n$  的减小,重建图像的质量由粗糙到精细,当位平面  $n$  为 0 时,重建图像完全重构。当在小的位平面结束编码时,重建图像有较高的图像质量,但编码流较大;而在大的位平面结束编码时,重建图像的质量相对低一点,但编码流却小一些。因此,在编码或解码的过程中,可以设置一定的目标码率,以获得所要求的重建图像质量。

基于上述位平面编码的基本描述,本文的编码算法可描述为

$$(1) t_n = 2^n, n = \lfloor \log_2(\max_{(i,j) \in I} |C_{i,j}|) \rfloor$$

$$(2) \text{EncodeBitplane}(t_n)$$

$$(3) n = n - 1, \text{ 返回到第 2 步。}$$

EncodeBitplane( $t_n$ )

对低频子带  $A_L$  中的每个坐标  $(i, j)$

$$\text{编码 } \delta(ST_{l_{\max}^{(0)}-L}(i, j), t_n)$$

$$\text{如果 } \delta(ST_{l_{\max}^{(0)}-L}(i, j), t_n) = 1$$

$$\text{EncodeCoefficient}(i, j, t_n)$$

$$\text{EncodeDescendants}(i, j + \bar{w}_L, t_n, L - 1)$$

}

EncodeDescendants( $i, j, t_n, l$ )

$$\text{EncodeSignificance}(ST_{l_{\max}^{(0)}-l}(i, j), t_n)$$

$$\text{如果 } \delta(ST_{l_{\max}^{(0)}-l}(i, j), t_n) = 1$$

对  $\{(i, j), (i + h_{l+1}, j - w_{l+1}), (i + h_{l+1}, j)\}$  中的每个坐标  $(i, j)$

$$\text{EncodeCoefficient}(x, y, t_n)$$

对  $\{(2i, 2j), (2i, 2j + 1), (2i + 1, 2j), (2i + 1, 2j + 1)\}$  中的每个坐标  $(x, y)$

$$\text{EncodeDescendants}(x, y, t_n, l - 1)$$

EncodeCoefficient( $i, j, t_n$ )

$$\text{如果 } C_{i,j} \leq 2t_n$$

```

EncodeSignificance( $C_{i,j,t_n}$ )
  如果  $\delta(C_{i,j,t_n}) = 1$ 
    EncodeSign( $C_{i,j}$ )
  否则
    EncodeBit( $C_{i,j,n}$ )
}
EncodeSignificance( $C,t$ ) {
  如果  $C < 2t$ 
    如果  $\delta(C,t) = 1$ , 输出“1”
    否则, 输出“0”
}
EncodeSign( $C$ ) {
  如果  $C > 0$ , 输出“1”  否则, 输出“0”
}

```

```

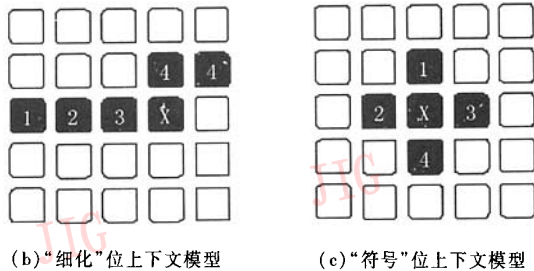
EncodeBit( $C,n$ ) {
  输出系数  $C$  的第  $n$  个比特位
}

```

在位平面编码过程中产生的“0”和“1”二进制符号序列由“重要性”位、“细化”位和“符号”位组成。为了提高压缩效率,对这些二进制符号进行自适应算术编码。和大多数的位平面编码方法一样,本文采用通用的 CACM87 编码器<sup>[8]</sup>。由于这些二进制符号流中存在着由小波变换产生的强相关性,本文采用了文献[9]中的上下文模型(Context)来预测它们的条件概率分布情况,然后再提供给自适应算术编码器进行编码。这里采用的3组上下文模型如图8所示。



(a)“重要性”位上下文模型



(b)“细化”位上下文模型

(c)“符号”位上下文模型

图8 用于算术编码的上下文模型

Fig. 8 Context for arithmetic coding

(图中的符号“X”表示当前需要编码的系数位)

### 4 实验结果与分析

对3幅  $512 \times 512$  的8位灰度图像 Lena、Barbara 和 Goldhill 进行测试,并将本文算法的编码结果分别和 EZW、SPIHT 算法进行比较。在表1中, Bit Rate 表示编码后每个像素所占的位数。PSNR 为峰值信噪比,它定义为

$$PSNR = 10\lg(255^2/MSE)$$

式中的 MSE 为原图像和重建图像之间的均方误差。和大多数的图像编码算法一样,本文算法采用 PSNR 作为编码质量的评价标准。从表1可以看出,本文算法在低码率下比 EZW 和 SPIHT 算法具有更高的重建图像质量。而对于其他的码率,本文算法仍然具有比 SPIHT 算法高的编码质量。

本文算法的实现有两个优点。其一是,在处理小波系数扫描的过程中,只需要很小的存储空间来存放小波系数的扫描参考信息,例如,对于  $M \times N$  大

表 1 本文算法的编码结果

Tab. 1 Encoding results of the proposed algorithm

图像 (512 × 512)	Bit Rate (bpp)	PSNR (dB)		
		EZW	SPIHT	本文算法
Lena	0.25	33.17	34.11	34.19
	0.5	36.28	37.21	37.28
	1.0	39.55	40.44	40.51
Barbara	0.25	26.77	27.58	27.63
	0.5	30.53	31.40	31.46
	1.0	35.14	36.41	36.45
Goldhill	0.25	30.31	30.56	30.64
	0.5	32.87	33.13	33.21
	1.0	36.20	36.55	36.62

小的 8 位灰度图像, 大约只需要  $M \times N/3$  字节的存储空间; 其二是, 由于简化了小波系数的扫描操作, 从而加快了算法的执行速度。本文算法消耗的执行时间主要包括小波变换、二叉树的构造和位平面编码 3 个过程所消耗的时间。在低码率编码的情况下, 本文算法所消耗的时间大部分用于小波变换, 而扫描树的构造和位平面编码只消耗小部分的时间。如图 9 所示, 在码率为 0.5bpp 时, 小波变换在不同大小的图像的编码过程中占了大部分的执行时间。对于本文算法来说, 提高小波变换的速度能够进一步提高算法的执行速度。比起其他很多的小波编码算法, 本文算法具有消耗存储空间资源少、实现简单和执行速度快等特点, 它降低了硬件实现的难度, 因而具有很强的实用性。

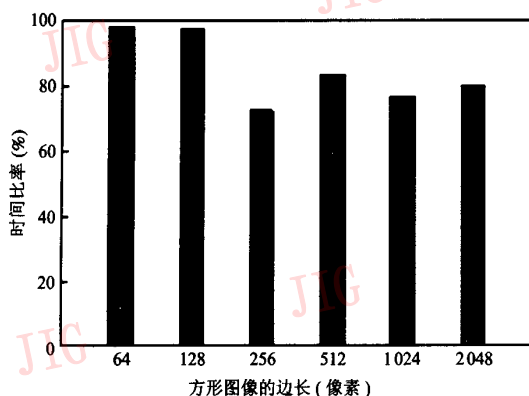


图 9 小波变换在整个编码过程中所占的时间比率 (码率为 0.5bpp)

Fig. 9 Time consumption ratio of wavelet transform in the whole encoding course

## 5 结 论

利用小波系数的空间分布特点, 提出了一种有效的位平面编码算法。该算法将子带二叉树进行合并处理, 利用扫描树中的节点来紧凑地表示小波域中不重要的小波系数块。在不同的阈值下, 本文算法能够一次性地扫描和编码不同尺度、不同方向子带中同一位置的重要系数比特位。这种扫描编码方式不仅能够有效地突出纹理细节丰富的局部图像区域的重要信息, 提高图像的编码质量, 而且简化了小波系数的扫描操作, 提高了算法的执行速度。本文算法实现简单高效, 而且编码效果好, 可适用于低码率的图像编码应用中。

## 参 考 文 献 (References)

- Shapiro J M. Embedded image coding using zerotrees of wavelet coefficients [J]. IEEE Transactions on Signal Processing, 1992, 41(12): 3445 ~ 3462.
- Said A, Pearlman W A. A new, fast and efficient image codec based on partitioning in hierarchal trees [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1996, 6(3): 243 ~ 250.
- ISO/IEC 15444, JPEG2000 image coding system [S].
- Taubman D. High performance scalable image compression with EBCOT [J]. IEEE Transactions on Image Processing, 2000, 9(7): 1158 ~ 1170.
- Hsiang S-T, Woods J W. Embedded image coding using zeroblocks of subband/wavelet coefficient and context modeling [A]. In: Proceedings of the IEEE International Conference on Circuits and Systems (ISCAS-2000) [C], Geneva, Switzerland, 2000: 662 ~ 665.
- Islam A, Pearlman W A. An embedded and efficient low-complexity hierarchical image coder [A]. In: Proceeding of SPIE International Conference on Visual Communications and Image Processing [C], San Jose, CA, 1999: 294 ~ 305.
- Andrew J. A simple and efficient hierarchical image coder [A]. In: Proceedings of the IEEE International Conference on Image Processing (ICIP-1997) [C], Santa Barbara, CA, 1997: 658 ~ 661.
- Witten I H, Neal R M, Cleary J G. Arithmetic coding for data compression [J]. Computing Practices, 1987, 30(6): 520 ~ 540.
- Bilgin A, Sementilli P J, Sheng F, et al. Scalable image coding using reversible integer wavelet transforms [J]. IEEE Transactions on Image Processing, 2000, 9(11): 1972 ~ 1977.