

# 一种基于小波变换的序列图像压缩编码

林 忠, 叶 清

(福州大学无线电工程系, 福建 福州 350002)

摘要: 研究一种动态图像小波域系数的压缩方法. 根据图像小波分解和人类视觉的特点及其子图像间的相关性, 对系数进行不同间隔的量化, 同时结合简单有效的数据结构——帧比特平面对系数进行扫描及压缩编码.

关键词: 图像编码; 序列图像; 小波变换; 比特平面

中图分类号: TN919. 81

文献标识码: A

利用小波变换进行图像压缩编码是一种十分有效的图像编码方案, 由于它可以对低频及高频特征同时进行处理, 有效地克服了 ADCT 方法中的方块效应及振铃效应, 同时也更适用于人类视觉系统(HVS)的特点, 得到了优于当前其他编码方法的编码结果<sup>[1,2]</sup>. 随着小波理论及其滤波器设计技术的不断完善, 目前已经有研究人员把小波变换植入了 JPEG、MPEG 标准. 运用小波变换进行图像编码虽然有多种方法, 但究其根本就是如何结合人类视觉特性更有效地对不同尺度的小波系数进行量化、编码的问题. 本文根据不同分解级小波系数幅度值间的相关特性, 提出一种基于小波系数比特平面结构的编码方法, 以提高编码效率.

## 1 图像的三维小波分解及其系数特性

为实现基于小波变换的序列图像编码, 将二维图像的多分辨分析扩展为三维, 采用三维可分离模型, 此时尺度函数为:  $\Phi(x, y, t) = \Phi(x) \Phi(y) \Phi(t)$ , 令  $\Psi(x)$  为与  $\Phi(x)$  相应的一维小波, 则在第  $k$  个分辨层, 三维的二进小波可表示为以下 7 个可分离的正交小波基函数.

$$\Psi^1(x, y, t) = \Phi(x) \Psi(y) \Phi(t)$$

$$\Psi^2(x, y, t) = \Psi(x) \Phi(y) \Phi(t)$$

$$\Psi^3(x, y, t) = \Psi(x) \Psi(y) \Phi(t)$$

$$\Psi^4(x, y, t) = \Phi(x) \Phi(y) \Psi(t)$$

$$\Psi^5(x, y, t) = \Phi(x) \Psi(y) \Psi(t)$$

$$\Psi^6(x, y, t) = \Psi(x) \Phi(y) \Psi(t)$$

$$\Psi^7(x, y, t) = \Psi(x) \Psi(y) \Psi(t)$$

将序列图像按每 8 帧为一个图像包作  $M = 3$  层小波分解. 图 1 所示为一级分解后形成的 8 个子带图像块. 图像包经 3 层分解可得到  $7M + 1 = 22$  个子带图像块, 其系数的特点主要表现在:

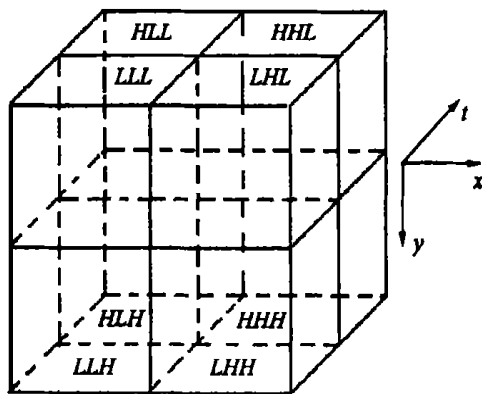


图 1 三维小波变换子带示意图

1) 图像的能量主要集中于最低精度的子图像块  $LLL$  中,  $LLL$  代表了图像的低频信息, 而  $LLH$ 、 $LHL$ 、 $LHH$  则表现出明显的水平、垂直、对角方向的细节信息。

2) 不同分解层同方向子带图像相应位置的系数幅值间存在明显的相关性。例如, 采用适当的量化门限作系数量化之后, 若  $LHL_3(x, y, t)$  为 0, 那么较高分辨层相应位置的系数  $LHL_2(2x, 2y, 2t)$ 、 $LHL_2(2x+1, 2y, 2t)$ 、 $LHL_2(2x, 2y+1, 2t)$ 、 $LHL_2(2x+1, 2y+1, 2t)$ 、 $LHL_2(2x, 2y, 2t+1)$ 、 $LHL_2(2x+1, 2y, 2t+1)$ 、 $LHL_2(2x, 2y+1, 2t+1)$ 、 $LHL_2(2x+1, 2y+1, 2t+1)$  的值为零或接近于零的概率可以达到 80% 以上。

## 2 小波域数据压缩算法

由于数据压缩的需要, 不可能对小波变换后的所有图象数据编码输出, 必须进行选择。量化后的系数利用阈值判断后形成的帧比特平面来决定是否进行编码。比特平面作为一种数据结构, 通过它可以充分地利用小波域各子图像相应系数间的相关性, 有效地表示数据, 而且其结构十分简单。以下给出具体的算法过程。

1) 首先对小波变换后的数据进行量化。考虑到 HVS 及小波系数的特点, 其量化间隔规定如下:

$$\text{Step} = w_1 \max(r) + w_2 \text{Average}(r, s, x, y, t)$$

其中:  $w_1$  和  $w_2$  是比例常数,  $r$  表示子图像的分解层次,  $s$  表示子图像的分解方向, 如图 1 所示,  $\max(r)$  分别取各层小波系数中的最大绝对值。

$$\text{Average}(r, s, x, y, t) = \begin{cases} \frac{1}{\text{number3}} \sum_{x, y \in (r=3), s \neq LLL} |I(x, y, t)| & (\text{if}(r = 3 \text{ and } s = LLL)) \\ \frac{1}{\text{number2}} \sum_{x, y \in (r=2)} |I(x, y, t)| & (\text{if}(r = 3 \text{ and } s \neq LLL)) \\ \frac{1}{\text{number1}} \sum_{x, y \in (r=1)} |I(x, y, t)| & (\text{if}(r = 2 \text{ or } r = 1)) \end{cases}$$

number1, number2, number3 分别为参与计算该层平均值时的像素总数。采用这种量化间隔的思路就是舍弃人眼不敏感的细节信息。

2) 对于量化后的系数依据阈值  $T$  确定其比特平面的初始分布。对于绝对值大于等于该阈值的系数, 相应位置比特平面的值定为 1; 对于绝对值小于该阈值的系数, 相应位置的比特平面的值定为 0, 即:

$$\text{if} (|I(x, y, t)|) \geq T \quad \text{bit-plane}(x, y, t) = 1 \\ \text{else} \quad \text{bit-plane}(x, y, t) = 0$$

3) 根据帧比特平面的分布, 对图象逐层进行搜索, 确定是否参与编码并调整帧比特平面。

① 集中了大部分能量的  $LLL$  块全部参与编码, 同时设置  $\text{bit-plane}(x, y, t) = 1, (x, y \in LLL)$ 。

② 对于  $LHL_3$ 、 $LLH_3$ 、 $LHH_3$ 、 $HLL_3$ 、 $HHL_3$ 、 $HLH_3$  及  $HHH_3$  块进行逐层搜索, 判断是否参与编码。例如: 如果  $LHL_3$  中  $(x, y, t)$  处的  $\text{bit-plane}$  为 1, 则  $(x, y, t)$  处的系数参与编码, 同时继续搜索下一层相应位置  $(2x, 2y, 2t)$ 、 $(2x+1, 2y, 2t)$ 、 $(2x, 2y+1, 2t)$ 、 $(2x+1, 2y+1, 2t)$ 、 $(2x, 2y, 2t+1)$ 、 $(2x+1, 2y, 2t+1)$ 、 $(2x, 2y+1, 2t+1)$ 、 $(2x+1, 2y+1, 2t+1)$  的  $\text{bit-plane}$  值, 以决定它们是否参与编码; 如果  $LHL_3$  中  $(x, y, t)$  处的  $\text{bit-plane}$  为 0, 则  $(x, y, t)$  处的系数不参与编码, 同时根据下一层相应位置的  $\text{bit-plane}$  值来

决定是否继续搜索及调整 bit-plane 值. 该算法可描述如下(其中 width 和 height 分别表示图像的宽度和高度, frameNumber 为一个图像包中所包含的帧数):

if ( $x, y \in LHL_3, LLH_3, LHH_3, HLL_3, HHL_3, HLH_3$  其中之一)

if (bit-plane( $x, y, t$ ) = 1)

{ OutputToEncodeFile(); // 输出到待编码文件

FindFileBit( $x, y, t, \text{bit-plane}, \text{width}, \text{height}$ ); // 搜索下一层相应位置的 bit-plane 值}

else { // 如果 bit-plane( $x, y, t$ )  $\neq$  1 则判断下一层相应位置的 bit-plane 值

if (FindNextLevelBit( $x, y, t, \text{bit-plane}$ ) = 1)

{

FindFileBit( $x, y, t, \text{bit-plane}, \text{width}, \text{height}$ );

}

else // 将下几层相应位置的 bit-plane 都设为 0

SetBitPlane( $x, y, t, \text{bit-plane}, \text{width}, \text{height}$ );

}

其中, FindFileBit( $x, y, t, \text{bit-plane}, \text{width}, \text{height}$ ) 算法描述如下:

if ( $x \geq (\text{width}/2) \parallel y \geq (\text{height}/2) \parallel t \geq (\text{frameNumber}/2)$ ) return;

for ( $k = 0; k \leq 1; k++$ )

for ( $i = 0; i \leq 1; i++$ )

for ( $j = 0; j \leq 1; j++$ )

{  $xx = 2x + i; yy = 2y + j; tt = 2k + t;$

if (bit-plane( $xx, yy, tt$ ) = 1)

{ OutputToEncodeFile(); // 输出到待编码文件

FindFileBit( $xx, yy, tt, \text{bit-plane}, \text{width}, \text{height}$ );

}

else // 将下几层相应位置的 bit-plane 设为 0

SetBitPlane0( $xx, yy, tt, \text{bit-plane}, \text{height}, \text{width}$ );

}

因此, 最后 bit-plane( $x, y, t$ ) = 1 的含义是指该位置的系数需要传送; 而 bit-plane( $x, y, t$ ) = 0 的含义是指该位置的系数不需要传送, 而并非第 2 步中阈值判断的结果.

③ 由于需要传送的系数已被量化至一个较小的范围, 可采用 6 比特编码. 对于帧比特平面进行游程编码.

### 3 实验结果及结论

采用本算法对标准的 8bit/pixel  $256 \times 256$  Lena 及 milkdrop 图象所生成的序列图像进行实验(取 8 帧标准图像, 每帧比前一帧右移一个像素, 左边空出的部分用原来最左边的像素值替代), 由实验的结果显示, 在一定压缩比的情况下, 重构图象均可获得较好的主观视觉质量, 具体结果(帧内采用双正交  $D_97$  小波,  $t$  方向采用 Haar 小波)参见图 2、图 3 及其说明.

利用这种图象压缩方法, 将小波域系数重要性、相关性与简单实用的帧比特平面数据结构相结合, 在一定压缩率下获得了较好的主观视觉质量. 如果在编码阶段加入熵编码(如算术编码)则可以进一步提高压缩率.



(a)原图像 Lena256×256 (8bpp)

(b)恢复的第3帧图像0.8bpp  
PSNR=29.95dB( $T=5.9$ )(c)恢复的第3帧图像0.5bpp  
PSNR=28.20dB( $T=9.6$ )

图2 Lena 图像处理结果



(a)原图像 Milkdrop256×256 (8bpp)

(b)恢复的第3帧图像0.7bpp  
PSNR=30.53( $T=3.7$ )(c)恢复的第3帧图像0.4bpp  
PSNR=29.45dB( $T=8.8$ )

图3 Milkdrop 图像处理结果

### 参考文献:

- [1] Lewis A, Knowles G. Image compression using the 2-D wavelet transform [J]. IEEE Trans on Image Processing, 1992, IP-2(2): 244-250.
- [2] Jerome Shapiro M. Embedded image coding using zerotrees of wavelet coefficients[J]. IEEE Trans on Signal Processing, 1993, SP-41(12): 3445-3462.

## A wavelet transform based image sequence compressive coding

LIN Zhong, YE Qing

(Department of Radio Engineering, Fuzhou University, Fuzhou, Fujian 350002, China)

**Abstract:** We present a novel dynamic image coding algorithm based on wavelet transform. In the light of the human visual system (HVS) and the strong correlativity of coefficients among the sub-images, a quantizer with different step-size is applied to each coefficient, and then by simple and efficient frame bit-plane structure, the coefficients are scanned and compressively encoded.

**Keywords:** image coding; image sequence; wavelet transform; bit-plane