第35卷 第15期 计 算 机 工 程 2009年8月
Vol.35 No.15 Computer Engineering August 2009
·多媒体技术及应用· 文章编号：1000—3428(2009)15—0229—03 文献标识码：A 中图分类号：TP301.6

# 3D-SPIHT算法在视频编码中的应用

张 宏，蒋金华
(哈尔滨理工大学计算机科学与技术学院，哈尔滨 150080)

**摘 要：**提出一种改进的三维等级树集分割编码算法。该算法针对三维等级树集算法对纹理复杂、边缘较多的图像编码效率低，存储空间和时间开销大等缺点进行了改进。经试验证明，改进后的三维等级树集分割编码算法提高了压缩效率，节约存储空间和时间开销，降低扫描编码复杂度，无方块效应，具有较好的性能。

**关键词：**三维小波；视频编码；三维等级树集分割编码算法

# Application of Improved 3D-SPIHT Algorithm in Video Coding

ZHANG Hong, JIANG Jin-hua
(Computer Science and Technology College, Harbin University of Science and Technology, Harbin 150080)

**【Abstract】**This paper introduces an improved 3D-SPIHT algorithm. This algorithm can deal with the problems such as low efficiency when coding pictures with complex textures or many-sided pictures, large storage requirement and long time consumption, which are common to traditional 3D-SPIHT algorithm. The test shows that improved 3D-SPIHT algorithm can increase the compress ratio, save storage and time consumption, reduce the complexity of scanning codes. This algorithm avoids diamond effect and has a good performance.

**【Key words】**three dimension wavelet; video coding; 3D-SPIHT algorithm

## 1 概述

传统的视频编码方法是运动补偿加帧内DCT变换编码。该方法已经被多种编码标准所采用，如H.261/H.263, MPEG-1, MPEG-2及MPEG-4等。但是这种编码方法在较高压缩比时，解码图像的块效应明显，使其在低码率压缩的性能不令人满意。而在小波域选择适当的编码方法进行图像压缩可以从根本上消除块效应，所以，小波域零树编码(EZW)的思想提出后[1]，得到了广泛关注。文献[2]对零树编码进行改进，提出了等级树分割编码方法(SPIHT)，后又提出了三维等级树分割编码(3D-SPIHT)方法，被认为是目前最好的编码方法之一。

本文提出了一种改进算法，该方法采用三维小波分析法，对分解后的三维时-空子带利用改进的3D-SPIHT算法编码，经实验证明，取得了比较好的效果。

## 2 三维小波变换

三维小波编码可以看成是二维小波编码在三维空间上的推广。

有2种方法实现三维小波变换[3]：(1)将二维小波变换的Mallallt金字塔算法推广到三维，即利用可分离滤波器，先对时间维进行变换，然后对空间的行和列进行变换，从而获得代表原序列图像不同时间和空间频率特性的子带图像。对时间低频和空间低频的子带信号重复进行这样的分解，可完成多级三维小波变换。(2)将序列图像时间一维变换和空间二维变换分开进行。先对图像在时间维按一维小波变换的方法进行分解，获得代表不同时间频率特性的时间子带图像，再对各子带进行空间二维小波变换，从而得到代表原图不同时间和空间频率的三维时-空子带。最后充分利用小波系数的各种特性，对三维子带进行量化、编码，和运动矢量一起形成最后的码流。本文采用第2种方法。如图1所示。

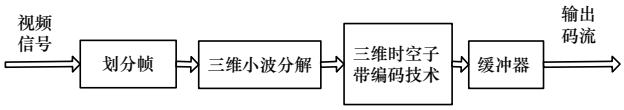


**图1 三维小波编码系统**

## 3 三维等级树集分割编码(3D-SPIHT)算法

3D-SPIHT算法是对SPHIT算法的扩展[4]，它在二维空间方向树的基础上加上时间轴，形成三维空间方向树，即八叉树结构。其中，系数重要性测试函数为

$$S_n(T)=\begin{cases}1 & \max_{(i,j,k)}(|C(i,j,k)|)\geqslant 2^n\\ 0 & \max_{(i,j,k)}(|C(i,j,k)|)<2^n\end{cases} \quad (1)$$

算法过程描述如下[4]：

(1)初始化：令$n=\lfloor \mathrm{lb}(\max_{(i,j,k)}(|C(i,j,k)|))\rfloor$，并传送$n$，令LSP表为空，LIP为所有$(i,j,k)\in H$的根坐标，LIS为所有$(i,j,k)\in H$且有子孙的根坐标。

(2)排序过程：对LIP中的项$(i,j,k)$进行处理，对LIS中的项，分$A$, $B$ 2种类型进行处理。

(3)细化过程：对LSP中的每一项$(i,j,k)$，不包括步骤(2)中加入的，输出系数的二进制表示的第$n$位有效位。

(4)量化更新：$n=n-1$，转向步骤(2)，直到预定的压缩比。

3D-SPIHT算法是在图像高频边缘信号较少、平滑部分较多、小波变换后的能量主要是集中在低频部分的假设基础上提出的，因此，该算法对复杂的图形编码效果不佳。另外该

**作者简介：**张 宏(1962－)，男，副教授、博士，主研方向：计算机系统设计；蒋金华，硕士研究生

**收稿日期：**2009-02-27 **E-mail：**jjh886@163.com

算法需要 3 个表：LSP, LIP 和 LIS，存储开销大。而要判断一个集合是否重要，是把这个集合的小波系数和阈值进行比较，因此对于小的小波系数，每一次扫描都要和当前阈值进行比较，这使得计算复杂度较高，不易于硬件实现，需对其进行改进。

## 4 3D-SPIHT的优化算法

针对 3D-SPIHT 算法存储空间大、计算复杂度高、编码时间长和对复杂图像的压缩效果不佳等缺点，提出了以下的改进方案：

(1)废弃 3D-SPIHT 算法中的 3 个列表，并定义了不同的树结构，将排序过程和精细化过程合为一体。引入了 2 个新的符号 $F_C(i,j,k)$和 $F_D(i,j,k)$，分别表示 $C(i,j,k)$和 $D(i,j,k)$的重要性状态，且重要性判断公式为

$$S_n(T(i,j,k))=\begin{cases}1 & \max_{(m,n,p)\in T}\{|C(m,n,p)|\}\geqslant 2^n\\ 0 & \max_{(m,n,p)\in T}\{|C(m,n,p)|\}<2^n\end{cases} \tag{2}$$

其中，$T(i,j,k)$表示 $C(i,j,k)$或 $D(i,j,k)$；$F_C$ 和 $F_D$ 的初始状态为“0”，当系数变为重要时为“1”。存储只要一位，极大地降低了存储空间。

(2)3D-SPIHT 算法在编码中判断一个集合是否重要的计算复杂度较高。本文引入了 $M_D(i,j,k)$表示子孙系数集 $D(i,j,k)$中阈值最大者，为节省空间，将其表示为 2 的对数：

$$M_D(i,j,k)=\lfloor 1\mathrm{b}\{\max(D(i,j,k))\}\rfloor \tag{3}$$

判断一个系数的子孙 $D(i,j,k)$是否为零树，只需用 $M_D(i,j,k)$与当前阈值进行比较，无需对 $D(i,j,k)$中所有的小波系数进行比较，有效地减少了扫描比较所需的时间。

(3)SPIHT 算法是对零树编码的改进，而零树编码的缺点在于：当一棵树不能满足零树的定义时，需要增加判断位用以判断该树包含的每棵子树是否为零树。因而，当树中包含的绝大多数子树都不是零树时，按照树结构方式编码这棵系数树所需的比特位要大于直接编码所有系数所需的比特位。这也是该算法对纹理复杂图像编码效果较差的主要原因。而且随着阈值的减小，孤立零点增多，零树减少，树结构方式编码所需的开销增大，增益相对减少。为了不出现负增益的情况，本文在对图像进行编码时先判断编码树结构所需位是否要大于它所代表的系数数。当树结构编码所需的位大于其代表的系数时，不用树结构的方式编码，而采用位平面的传送方法。判断过程如下：

对于某一阈值 $T=2^n$，记 $a[1\ldots N]$为 2~$N$ 级子带上满足 $M_D(i,j,k)<n$ 的系数数，它表示对于 $T$，系数$(i,j,k)$的后代是零树，在下次编码时需要处理。记 $b[1\ldots N]$表示 2~$N$ 级子带上满足 $M_D(i,j,k)<n$-1 的系数数，它表示对下次编码的阈值 $T/2$，系数$(i,j,k)$的后代为零树。如果对于阈值 $T$ 为零树的系数树(标记为 $Z$-$T0$)，对于阈值 $T/2$ 都不为零树，则在阈值 $T/2$ 下编码 $Z$-$T0$ 的树结构所需要的位为

$$N1=a_n\times G+\sum_{i=2}^{n-1}(a_n-4\times a_{n-1})\times G \tag{4}$$

其中，$G$ 表示每个有后代的节点的子孙不是零树的重要性。对于阈值 $T/2$ 为零树的树(标记为 $Z\_T1$)所能表示的系数数为

$$N2=b_n\times E+\sum_{i=2}^{n-1}(b_n-4\times b_{n-1})\times E \tag{5}$$

其中，$E$ 表示每个节点的重要性，$Z\_T1$ 所能表示的所有后代的节点数(即其中包含的子树的棵树)为

$$N3=b_n\times G+\sum_{i=2}^{n-1}(b_n-4\times b_{n-1})\times G \tag{6}$$

编码 $Z\_T1$ 所需的位数为

$$N4=b_n+\sum_{i=2}^{n-1}(b_n-4\times b_{n+1}) \tag{7}$$

在给定阈值 $T/2$ 编码时，实际用于表示树结构所需要的位数为

$$N5=N1-N3+N4 \tag{8}$$

其中，$N2$ 表示用树结构编码所带来的增益；$N5$ 表示用树结构编码所需要的开销。如果 $N2<N5$，此时采用位平面的方式进行编码，即仅对系数的重要性进行编码和传送，不再考虑其后代的状态，有效地提高了编码效率。

本改进算法中 $F_C$ 和 $F_D$ 有 3 种取值：1 表示该节点重要，0 为不重要，2 表示该节点已在零树中，无须处理。改进算法描述如下：

(1)初始化：令 $n=\lfloor 1\mathrm{b}(\max_{(i,j,k)}(|C(i,j,k)|))\rfloor$，并传送 $n$，$F_C$ 和 $F_D$ 值为 0，$a[1\ldots N]$, $b[1\ldots N]$也初始化为 0，$L=N$, $Flag=0$，对每个 $F_D$ 表示的树结构，用式(2)计算 $M_D$。

(2)以 $Z$ 扫描方式处理各子带的每个系数，对子带中的任意系数 $C(i,j,k)$：

1)若 $F_C=1$，输出 $C(i,j,k)$的细化位；若 $F_C=0$，判断系数是否重要，如果 $C(i,j,k)\geqslant 2^n$，则 $F_C=1$，输出 $F_C$，并输出 $C(i,j,k)$的符号位和细化位。如果 $C(i,j,k)\leqslant 2^n$，则 $F_C=0$，并输出。如 $F_C=2$，则 $F_D=2$，判断该子带中系数的 $F_C$ 是否处理完，若处理完，则转到 2)，否则回到 1)。

2)判断 $Flag$ 是否为 1，如果是转到 3)，否则处理 $F_D$。若 $F_D=0$，判断该系数是否重要，如果 $M_D(i,j,k)\geqslant n$，则 $F_D=1$，输出 1；如果 $M_D(i,j,k)<n$ 且 $F_D=0$，则输出 0，并将 $D(i,j,k)$中的所有系数 $C(m,n,p)$的重要性标志 $F_C$ 置为 2，且 $a[L]=a[L]+1$；判断 $M_D$ 取值是否大于 $n$-1，若不是，则 $b[L]=b[L]+1$；若 $F_D=2$，则 $a[L]=a[L]+1$，如果 $M_D(i,j,k)<n$-1，则 $b[L]=b[L]+1$；判断该子带的 $F_D$ 是否处理完，若没有，回到 2)，否则转到 3)。

3)判断该级的所有子带是否处理完，若没有，回到 1)；否则 $L=L$-1，判断 $L$ 是否小于 1，如是转到(3)，否则回到 1)。

(3)判断是否达到编码效果，是则结束，否则 $n=n$-1。将 $F_C$ 和 $F_D$ 中所有的元素取值非 1 的置为 0，根据 $a[1\ldots N]$, $b[1\ldots N]$的值和式(4)~式(8)计算 $N1$, $N2$, $N3$, $N4$, $N5$，如果 $N2<N5$，则 $Flag=1$，输出标志，表示以后不需要再处理 $F_D$，并把 $a[1\ldots N]$, $b[1\ldots N]$初始化为 0，$L=N$，回到(2)。

## 5 试验仿真

为了验证以上改进算法，分别对 CIF 格式的标准视频序列 Clarie, Salesman 进行仿真试验，采用 Daubichies9/7 小波基。取 16 帧为一组进行三维小波变换，分别对时间和空间上进行 4 级小波分解。重建图像的平均峰值信噪比(*PSNR*)对比如表 1 与表 2 所示。

**表 1 Clarie 序列峰值信噪比对比表**

| 平均码率/(bit·pixel$^{-1}$) | 原 3D-SPIHT 峰值信噪比(*PSNR*) | 改进 3D-SPIHT 峰值信噪比(*PSNR*) |
|---|---|---|
| 0.25 | 34.403 | 36.961 |
| 0.50 | 37.992 | 38.513 |
| 1.00 | 42.757 | 43.213 |

**表 2　Salesman 序列峰值信噪比对比表**

| 平均码率 /(bit·pixel$^{-1}$) | 原 3D-SPIHT 峰值信噪比 (*PSNR*) | 改进 3D-SPIHT 峰值信噪比 (*PSNR*) |
|---|---|---|
| 0.25 | 33.325 | 34.268 |
| 0.50 | 35.834 | 37.651 |
| 1.00 | 39.719 | 42.582 |

由仿真数据可以看出，改进后的 3D-SPIHT 算法对平滑图像 Clarie，可以保持三维小波变换编码的特点，在高压缩比下，重建图像仍能得到较高的图像质量，并有所提高。对于 Salesman 这样的图像背景比较复杂、场景中的物体运动也较大的图像，重建图像的质量有明显改善，在较大的压缩比下仍能得到重建质量良好的视频图像。

以计算时间为比较标准，分别取 Clarie 和 Salesman 序列图像的前 16 帧为例比较 2 种算法的复杂度，如表 3 所示(其中扫描编码时间不包括算术编码时间)。由表可以看出，改进算法的编码时间明显下降，尤其是对 Salesman 这样背景比较复杂的图像，效果更加明显。

**表 3　Clarie 和 Salesman 序列图像编码时间对比表**

| 平均码率 /(bit·pixel$^{-1}$) | 原 3D-SPIHTClarie 编码时间/s | 改进 3D-SPIHTClarie 编码时间/s | 原 3D-SPIHTSalesman 编码时间/s | 改进 3D-SPIHTSalesman 编码时间/s |
|---|---|---|---|---|
| 0.25 | 22.64 | 20.71 | 24.53 | 21.91 |
| 0.50 | 30.92 | 25.47 | 35.94 | 30.37 |
| 1.00 | 66.26 | 60.58 | 72.38 | 62.76 |

另外，改进的 3D-SPIHT 算法还大大节省了存储空间，如以 $L$ 帧为一组，图像大小为 $M\times N$ 的图像，像素个数为 $M\times N\times L$。原 3D-SPIHT 算法要 $2\times M\times N\times L\times 27$ 位的存储空间，此外 3D-SPIHT 算法中的列表长度是可变的，还要对每个列表留有足够的空间，因此存储开销大。而对于本文提出的改进算法，$F_C$ 要 $M\times N\times L$ 位存储空间，$F_D$ 要$(M\times N\times L)/8$ 位，$M_D$ 需要 $M\times N\times L$ 位，所以总共要 $2\times M\times N\times L+2\times(M\times N\times L)/8+M\times N\times L$ 位，只需要原算法的存储空间的 5.804%，极大地节省了存储空间。

## 6　结束语

本文提出的改进算法保持了 3D-SPIHT 对平滑图像的压缩效果，提高了对复杂图像的压缩质量，节约了时空开销，从主客观上都取得了良好的效果。在今后的研究中，还应对三维小波变换进行深入的研究，以减少时间冗余，提高压缩效率，进一步提高图像的压缩比。

## 参考文献


[1] Shaprio J M. Embedded Image Coding Using Zero Trees of Wavelets Coefficients[J]. IEEE Transactions on Signal Processing, 1993, 41(12): 3445-3462.

[2] Hsiang S T. High Scalable Subband/Wavelet Image and Video Coding[D]. Troy, New York, USA: Rensselaer Polytechnic Institute, 2002.

[3] 马　宣, 马海武, 周　杰. 基于三维小波变换的视频图像的压缩算法研究[J]. 计算机技术与发展, 2006, 16(10): 89-93.

[4] Kim B J, Pearman W A, Xiong Zixiang. Low Bit-rate Scalable Video Coding with 3-D Set Partitioning in Hierarchical Trees (3D-SPIHT)[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2000, 10(8): 1374-1387.


编辑　任吉慧

（上接第 228 页）

度及 *PSNR* 值。对 $N\times N$ 图像，空间复杂度 $O_S$ 为处理一行像素需要的最少存储空间；时间复杂度 $O_T$ 为单个像素进行加、减、乘、除等运算的次数，大于 2 000 为“高”，500~2 000 为“中”，小于 500 为“低”。

**表 1　逆有序抖动算法的性能比较**

| 算法 | $O_S$ | $O_T$ | *PSNR* /dB |
|---|---|---|---|
| MAP 改进算法[1] | $81N$ | 中 | 27.5 |
| 自适应算法[3] | $3N^2$ | 中 | 26.7 |
| 小波算法[2] | $9N^2$ | 中 | 26.9 |
| 非线性滤波算法[4] | $28N$ | 中 | 27.6 |
| 本文算法 | $42N$ | 低 | 28.5 |

从表中可以看出，该算法在 *PSNR* 值上有较大的优势，算法在空间复杂度上优于小波算法、MAP 算法、改进 MAP 算法和自适应算法，但不及其非线性滤波算法。在时间复杂度上最低，由于它主要基于滤波运算。

## 5　结束语

本文提出了一种基于 Laplacian 金字塔的逆有序抖动方法。算法采用分治策略，在有效抑制噪声的同时，保持了图像的重要细节。从仿真实验可以看出，该算法复杂度低，在细节保持、*PSNR* 值等方面有较大的优势，有利于实际应用。

## 参考文献


[1] 郑海红, 曾　平. 基于最大后验概率的逆半调改进算法[J]. 西安交通大学学报, 2005, 39(12): 1340-1343, 1357.

[2] Xiong Zixiang, Orchard M T, Ramchandran K. Inverse Halftoning Using Wavelets[J]. IEEE Transactions on Image Processing, 1999, 8(10): 1479-1483.

[3] Chen Liming, Hang Hsuehming. An Adaptive Inverse Halftoing Algorithm[J]. IEEE Transactions on Image Processing, 1997, 6(8): 1202-1209.

[4] 王　波, 曾　平. 一种非线性逆有序抖动的方法[J]. 计算机工程与应用, 2004, 40(21): 63-65.

[5] Kitamura M, Ono F. Efficient Inverse Halftoning of Ordered Dither Images Utilizing Matrix Addrcjsses and Flat Area Detection[C]// Proc. of IEEE Region 10 Annual International Conference. Hong Kong, China: [s. n.], 2007: 339-414.

[6] Saika Y, Tahara N, Yamasaki T. Image Restoration from Corrupted Halftone Images Using the ε-filter, Discriminant Analysis Method and the Generalized Statistical Smoothing[C]//Proc. of International Conference on Control, Automation and Systems. Seoul, Korea: [s. n.], 2008: 75-78.

[7] 宋春林, 冯　瑞, 金　炜, 等. 改进的多分辨率 SPIHT 算法[J]. 计算机工程, 2008, 34(4): 241-243.

[8] Kite T D, Venkata N D, Evans B L, et al. A Fast, High Quality Inverse Halftoning Algorithm for Error Diffused Halftones[J]. IEEE Transactions on Image Processing, 2000, 9(9): 1583-1592.


编辑　任吉慧