

快速3D-MSPIHT视频编码算法

刘春辉 龚声蓉 张书奎 刘纯平

(苏州大学计算机学院, 苏州 215006)

摘要 多媒体传感器节点的处理能力、存储能力和通信能力相对较弱,给视频编码传输带来了极大的挑战。提出了一种改进的3维等级树集分割(3D-MSPIHT)的视频编码方法,将等级树集分割(SPIHT)算法予以改进并推广应用于3维视频编码中,该算法采用提升小波进行3维子带分解,通过改进的根和分割方法实现对系数重排序,保证更多重要的信息更早地编码于比特流中;同时采用无链表的编码结构,不需要进行复杂繁重的运动估计和补偿,实现了较少的内存和能量消耗,能够提供较好服务质量(QoS)。实验结果表明,此方案优于3D-SPIHT算法,低比特率情况下尤为突出;与H.264等编码方案相比,在图像质量基本相同的情况下,编码速度提高了11.5~11.8倍。

关键词 视频编码 提升小波 3维等级树集分割 多媒体传感器网络

中图法分类号: TP301.6 **文献标识码:** A **文章编号:** 1006-8961(2009)10-2159-06

Fast Video Coding Algorithm with 3D-MSPIHT

LIU Chun-hui, GONG Sheng-rong, ZHANG Shu-kui, LIU Chun-ping

(College of Computer Science and Technology, Soochow University, Suzhou 215006)

Abstract Multimedia sensor node's processing power, storage capacity and communication is relatively weak, which posed a great challenge on video coding and transmission. This article will improve SPIHT(set partitioning in hierarchical trees) algorithm and propose an improved three-dimensional set partitioning in hierarchical trees(3D-MSPIHT) video coding algorithm. This algorithm uses three-dimensional lifting wavelet decomposition, improved methods of the root and partition, to achieve the resorting of coefficients and ensure that more important information can be encoded earlier in the bit stream. At the same time, it adopted the listless coding structure, as well as no need for complex and heavy motion estimation and compensation, achieved less memory and energy consumption, and considerable QoS(quality of service). The experimental results show that this algorithm is better than 3D-SPIHT, particularly in low bit rate circumstances; and compared to coding schemes such as H.264, besides the all square image quality, encoding speed enhance 11.5~11.8 times.

Keywords video coding, lifting wavelet, 3D-MSPIHT, multimedia sensor networks

1 引言

作为一种全新的信息获取和处理技术,多媒体

传感器网络较之传统传感器网络更多地关注于音频、视频、图像等大数据量、大信息量媒体的采集与处理,在军事、民用及商业领域中具有广阔的应用前景^[1]。在多媒体传感器网络中,实时流媒体的大量

基金项目:国家自然科学基金项目(60873116);江苏省高校自然科学基金项目(07KJD520186);江苏省现代企业信息化应用支撑软件工程技术研发中心开放基金项目(SX200903);江苏省自然科学基金项目(BK2009116;BK2009593);江苏省科技支撑计划(工业)项目(BE2009048)

收稿日期:2009-07-01;**改回日期:**2009-07-22

第一作者简介:刘春辉(1985~),男,苏州大学计算机科学与技术学院计算应用与技术专业硕士研究生。主要研究方向为图像与视频处理、视频编码与传输等。E-mail:spring_lch@gmail.com

通讯作者:龚声蓉,E-mail:shrgong@suda.edu.cn

应用对音频/视频编码提出了更高的技术要求。无线传感器网络的自身特点决定了单个传感器节点很难支持复杂的 MPEGx 或者 H. 26x 系列编码^[2]。小波压缩多层编码能够灵活地适应信道状态,选择最优层数等。较之其他编码方法,该方法能提供较好的媒体服务质量。但传统小波节点计算时耗较长,消耗能量较大。

小波变换用于图像压缩已经相当成熟。由 Shapiro 提出的嵌入式零树编码 (EZW)^[3] 以及 Said 等人提出的层次树集分割编码 (SPIHT) 均展示出以小波变换为基础的图像压缩具有较好的效果^[4]。近年来,将离散小波变换应用到视频编码中也成为了一个研究热点。Chen 等人实现了将 2 维零树编码 (2D-IEZW) 扩展到 3 维 (3D-IEZW)^[5], 在不需要进行任何运动估计和补偿的情况下实现了高效和运算简单的视频编码系统。Kim 等人提出了嵌入式 3 维层次树集分割的视频编码 (3D-SPIHT)^[6], 得到了很高的编码效率和性能, 而且编码流也是完全嵌入式的, 码率的分配在编码过程中自动完成。然而, 尽管 SPIHT 算法具有出色的率失真性能, 同时提供了一个嵌入式流, 可以准确成就任何比特率。然而, 该

算法使用效率低下的集分割方法, 使得该算法不能完全提供所需的逐步传输功能和空间的可扩展性, 难以适合多媒体传感器网络这类的应用。

2 SPIHT 算法分析

SPIHT 算法的本质是优先传输重要信息, 但这也正是其未能真正实现的地方。用来表示不重要系数的数据位虽没有编码图像信息, 但却是必须的, 而它恰恰也是导致重建图像失真的根源。树根中各系数的重要性级别有很大区别, LL 频带系数重要性显著高于其他子带, 却没有给予优先编码。所有树根的重要性是每一个系数依次判别, 而它的孩子们是以 4 个一组的形式。如果树根中某 4 个系数是不重要的, 那么需要 4 位来表示, 然而, 如果它们也是 4 个一组, 却只需要 1 位就能表示这些不重要的信息。如果这些树根系数很小, 那么这些没有携带图像信息的码字将被多次的传输。在低比特率情况下, 由于码流中高比例的无信息码字将导致图像质量的严重下降。如图 1(a) 所示。

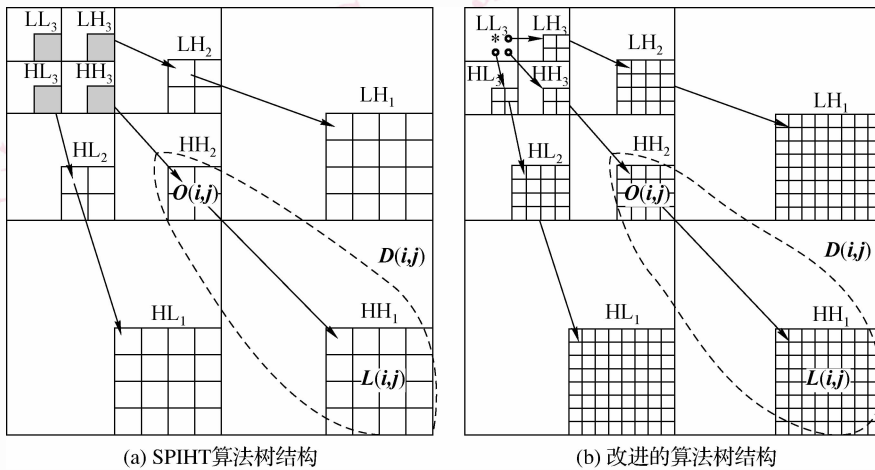


图 1 算法树结构比较

Fig. 1 Comparison of partitioning structure of algorithm

SPIHT 编码算法使用一个链表结构来保存和索引各分割集。链表的使用伴随着一个可变的, 随数据变化的内存需求, 需要同时维护两个链表。而且节点的添加、删除、移动需要大量的内存管理操作, 随着比特率的增加, 链表节点将可能高达系数总量 2 倍之多, 苛刻的内存需求和高昂的能量需求导致

了实际应用的不尽如人意。

针对以上的缺陷, 不少研究者进行了改进, Danyali 等人的 3D-VSPIHT 对 LL 频带再进行一次虚拟的分解^[7], 虽然改进了树的结构, 但却加剧了算法的复杂度。丁文奇等人利用消除树间冗余的方法改进了 3D-SPIHT 算法^[8], 提高了视频的峰值信

噪比 (PSNR), 但却是以繁重的消除冗余算法为代价, 这些改进都不能适应于多媒体传感器节点能量低, 处理能力弱的要求。本文提出了一种低复杂度的 3 维的改进等级树集分割算法 (3D-Modified SPIHT, 3D-MSPIHT)。

3 3D-MSPIHT 算法

3.1 算法思想

算法采用可扩展的低复杂度的完全嵌入式的小波压缩的多层编码算法, 可以实现精确的比特率控制, 以达到所要求的码率和视频服务质量。首先, 视频序列进行 3 维提升小波分解, 由于提升小波的运用, 每组数据每级分解仅需 8 次加法和 6 次乘法共 14 次浮点运算, 而原算法在同等情况下需 23 次, 因此, 总的运算代价将为原来的 60.87%, 大大地提高小波分解的速度。其次, 对分解的频带, 仅 LL₃ 作为树根, 系数每 2 × 2 × 2 为一组, 其他子带为 LL₃ 的子孙, 建立 3 维改进集分割等级树。然后, 采用 3D-MSPIHT 进行编码, 它具有许多优势, 首先, 它缩短了根的长度, 这意味着排序环节时间的缩短和速度的提高。其次, 通过使用这种新的分割结构所实现的重排序, 只需要一位来确定系数的重要性, 原算法中需要 4 位。另外, 如果某根数据是不重要的, 那么它所代表的那个区域也不需要进一步的细化了。因此, 此改进不仅减少了编码根中不重要信息的比特数, 同时也将更多根中重要信息编码于细化环节之前, 提高了重要信息更早编码的比例, 提高了低码率情况下视频序列的 PSNR。编码时引入一个静态的表, 每个表项用一个两位的特定符号来标示其状态; 同时, 内存需求是固定的, 且不要进行复杂费时的链表操作。采用这种结构不仅提高了编码的速度, 也极大地减少了编码的内存需求, 以 352 × 288 灰度图像为例, SPIHT 算法需要: 352 × 288 × 2 × 18 / (8 × 1 024) = 446K bytes, 然而, 静态表仅需要: 352 × 288 × 2 / (8 × 1 024) = 25K bytes。

3.2 算法流程

首先, 视频帧每 16 帧为一组, 如图 2(a) 所示, 在时空域进行 3 级小波分解, 这里采用提升格式的 Daub 9/7 双正交小波^[9]。通过在分解滤波器组和合成滤波器组中应用多相描述, 小波正变换和逆变

化可分别表示为

$$\begin{bmatrix} \lambda \\ \gamma \end{bmatrix} = \tilde{\mathbf{P}}(z^{-1})^T \begin{bmatrix} \chi_e(z) \\ z^{-1}\chi_o(z) \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} y_e(z) \\ zy_o(z) \end{bmatrix} = \mathbf{P}(z) \begin{bmatrix} \lambda \\ \gamma \end{bmatrix} \quad (2)$$

x, y 为输入信号和重建之后的信号, λ, γ 为低通滤波 $\tilde{h}(z^{-1})$ 和高通滤波 $\tilde{g}(z^{-1})$ 的输出信号; $\mathbf{P}(z)$ 为多相矩阵, $\tilde{\mathbf{P}}(z)$ 是其对偶矩阵。

给出任一个互补滤波器对 (h, g), 总存在 Laurent 多项式和一个非零常量与之对应。所以多相矩阵 $\mathbf{P}(z)$ 可以分解因式为

$$\begin{aligned} \mathbf{P}(z) &= \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} \\ &= \prod_{i=1}^m \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} k & 0 \\ 0 & 1/k \end{bmatrix} \quad (3) \end{aligned}$$

原始的 提升的 对偶的 提升的 标准化

因此, 包含分解滤波器组的对偶矩阵的共轭转置表示为

$$\begin{aligned} \tilde{\mathbf{P}}(z^{-1})^T &= \begin{bmatrix} \tilde{h}_e(z^{-1}) & \tilde{h}_o(z^{-1}) \\ \tilde{g}_e(z^{-1}) & \tilde{g}_o(z^{-1}) \end{bmatrix} \\ &= \begin{bmatrix} k & 0 \\ 0 & 1/k \end{bmatrix} \prod_{i=m}^1 \begin{bmatrix} 1 & 0 \\ -t_i(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & -s_i(z) \\ 0 & 1 \end{bmatrix} \quad (4) \end{aligned}$$

标准化 对偶的 提升的 原始的 提升的

其次, 小波分解后, 对分解的子带采用改进的分割算法构造等级树。最高层 LL 子带每 2 × 2 × 2 像素为一组。每一组中除一个节点没有孩子, 其余每个节点都有 8 个孩子。如图 2(b) 所示。

图中从树根节点到 2 × 2 × 2 块的箭头表明了从父节点到子节点的父子关系。除了 LL 子带和没有孩子的最底层节点, 其余层次的节点父子关系可表述如下:

$$\begin{aligned} \mathbf{O}(i, j, k) &= \{ (2i, 2j, 2k), (2i, 2j+1, 2k), \\ & (2i+1, 2j, 2k), (2i+1, 2j+1, 2k), \\ & (2i, 2j, 2k+1), (2i+1, 2j, 2k+1), \\ & (2i, 2j+1, 2k+1), \\ & (2i+1, 2j+1, 2k+1) \} \end{aligned}$$

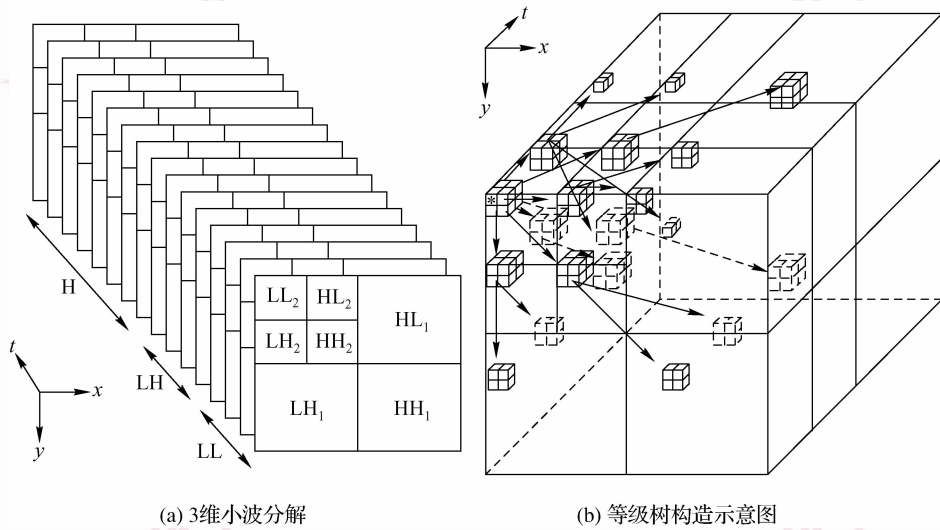


图 2 3 维小波分解和等级树

Fig. 2 3D wavelet decomposition and hierarchical trees

然后,对构造的等级树,进行等级树集分割算法编码,算法具体实现见 3.3 节。

整个算法的流程图如图 3 所示。编码端,视频序列首先经过 3 维小波分解,构造等级树,然后进行 3D-MSPIHT 编码,编码后的码流通过传感器网络信道传输至汇聚节点。在解码端,对接收到的码流进行 3D-MSPIHT 解码,再进行 3 维小波逆重建,得到重建的视频序列。

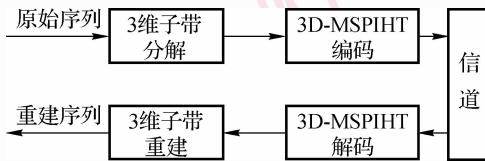


图 3 3D-MSPIHT 编码流程图

Fig. 3 The flow chart of 3D-MSPIHT algorithm

3.3 改进集分割算法实现

编码的可扩展和渐进式性能可以通过引入重要性重排序得以提升,定义新的根和改进的分割结构,如图 1(b)所示,树根仅由 LL_3 节点构成, LL_3 中的系数每 $2 \times 2 \times 2$ 像素为一组,其中每组有一个没有后代,由 * 标记。这样,根节点数量将仅为原来的 $1/4$;改进的分割结构将低频带的重要信息比 SPIHT 算法更早的编码到比特流中,来更好地利用能量压缩小波系数。采用静态表的结构,每个表元素对应一个系数,每个状态标志占用 2 个比特被存放在固定的表元素中,标志着所对应系数的状态。每个量

的意义如下:

ROOT:仅包含 LL_3 子带;

OP:子节点像素,将马上被判定重要性;

SP:重要的像素点;

IP:不重要的像素点;

RP:细化过程中定义的将被细化的重要像素点。

在编码之前,可以预先设置一个最大值矩阵 $\max(i, j)$:记录的是 (i, j) 的兄弟节点及其所有子节点的最大值,这样可以减少编码中的重复运算,提高编码的效率。算法描述如下:

(1) 初始化

首先输出门限值 $= \lfloor \log_2(\max_{(i,j)} \{ \|C_{i,j}\| \}) \rfloor$ 的位数。设置 **ROOT** 集合中系数所对应的静态表元素为 **OP**。

(2) 排序过程

判断每一组 $2 \times 2 \times 2$ 的 **OP** 系数所对应的最大值矩阵 $\max(i, j)$ 中的值。如果值小于门限值,输出 0,不需要进行任何接下来的操作,只用 1 比特就对此树进行了编码。如果大于门限值,就输出 1,进而判断每一个系数的值。如果大于门限值,输出 1,并输出相应的符号,正值输出 0,负值输出 1,接着将此系数对应的表元素状态设置为 **SP**,如果小于门限值,输出 0,修改相应的表元素状态位 **IP**,如果系数有子孙(其中一个系数没有子孙),则设置它儿子所对应的表元素状态为 **OP**。

(3) 细化过程

对于有所状态为 *RP* 的元素,输出此门限值位的值。修改表中所有状态为 *SP* 的元素状态为 *RP*,它们将在下一轮细化过程中输出。

(4)更新门限

将门限值除以 2,设置所有状态为 *IP* 的 *ROOT* 元素为 *OP*,跳至步骤(2),进行新一轮细化编码。

4 实验结果

针对本文提出的视频编码算法,对两种不同类型的 QCIF(quarter common intermediate format)(分辨率为 176×144) 视频序列进行了编码实验: Carphone 背景和前景都在变化,运动相对较为剧烈; M&D 只有前景变化,运动相对较少。实验硬件环境: Intel 奔腾处理器 2.8 GHz, 512 M 内存。

首先针对本文提出的码流重要性重排序进行了分析,对编码后的码流分析表明,本文的方法更多的拥有图像信息的码字被更早的编码于码流中,在越低码率情况下,改善越明显。实验结果如表 1 所示。

表 1 在不同低比特率下重要性码字所占比率

Tab.1 Percentage of significant bits at different low bit rates

比特率 (bpp)	3D-SPIHT	本文方法
0.005	26.74	70.46
0.01	32.08	74.56
0.05	43.26	76.39
0.1	48.87	78.24
0.2	51.35	79.92

其次,对视频编码算法的分解阶段和码字编码阶段的处理时间和内存消耗与原算法进行比较,结果如表 2 所示。算法处理时间大部分在小波变换阶段,本文采用的提升小波方案,处理时间仅为原方案的 35% ~ 60%,提高的速率是相当可观的,相较而言,内存消耗提高不多;在码字编码阶段,由于本文采用了静态表的编码方式,相较于原来的动态链表,内存消耗显著减少,仅为原方案的 5% ~ 6%,而且不需要花费大量的时间对链表进行操作,处理时间也仅为原来的 47% ~ 53%。由于视频序列本身的不同,处理时间和内存消耗也存在不同程度的波动。

表 2 时间和内存消耗比较

Tab.2 The ratio of time and memory consumption

		使用方法	处理时间	内存消耗
3 维小波变换	3D-SPIHT	Daub 9/7	35% ~ 60%	85% ~ 95%
	3D-MSPIHT	提升小波		
码字编码	3D-SPIHT	SPIHT	47% ~ 53%	5% ~ 6%
	3D-MSPIHT	MSPIHT		

然后,对 Carphone 和 M&D 两个序列分别进行编码后的视频信噪比分析,并与 3D-SPIHT 编码系统以及 H. 264 标准(JM 13.2) 进行比较。实验结果如图 4 所示,3D-MSPIHT 与 3D-SPIHT 相比较大大提高了低码率情况下的 PSNR 达 1 ~ 5 dB,比特率越低越明显。同时此算法接近甚至超出了 H. 264 的视频质量。

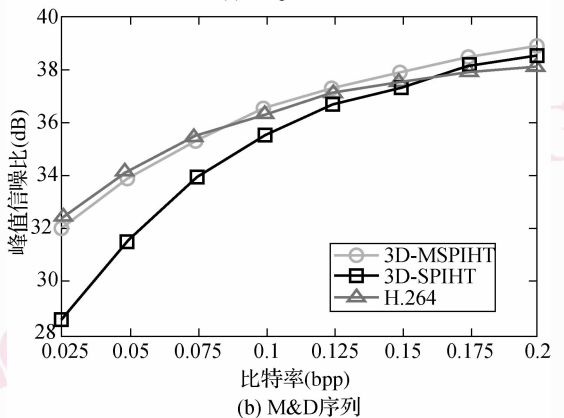
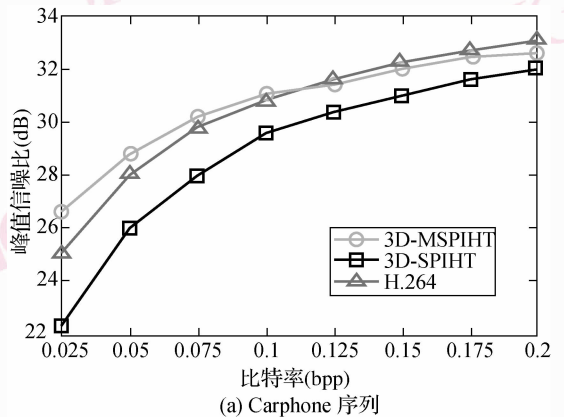


图 4 PSNR 比较

Fig.4 Comparison of PSNR

最后,针对 M&D 序列在不同低比特率下对各种编码方案的编码速度进行了比较,结果如表 3 所示。3D-MSPIHT 分别为 3D-SPIHT, H. 263 和 H. 264 的 1.6 ~ 1.7 倍, 4.0 ~ 4.3 倍和 11.5 ~ 11.8 倍。可

见速度优势是非常突出的。

表 3 各编码方案每秒编码的帧数

Tab. 3 The number of frames coded per second under each coding schemes

比特率 (bpp)	3D-MSPIHT	3D-SPIHT	H. 263	H. 264
0.025	33.8	19.9	7.9	2.9
0.05	29.4	18.4	7.3	2.7
0.1	28.7	16.9	6.7	2.5

5 结 论

本文针对多媒体传感器节点的处理能力、存储能力和通信能力相对较弱,单个传感器节点很难支持复杂的 MPEGx 与 H. 26x 系列编码这一问题,将 SPIHT 算法予以改进应用于视频编码中,提出了一种适合于多媒体传感器网络的快速 3D-MSPIHT 视频编码方法。此算法采用提升格式的 Daub 9/7 双正交小波,处理时间仅为原方案的 35% ~ 60%;对系数重排序,保证更多重要的信息更早地编码于比特流中,大大提高了低码率情况下视频序列的 PSNR 达 1 ~ 5 dB,同时也能更灵活地适应信道状态;采用无链表的树结构,实现了较少的内存和能量消耗,以及不需要进行复杂繁重的运动估计和补偿,在一定程度上适用于多媒体传感器网络资源受限和能量不足的情况,并能够提供较好 QoS。实验结果表明,此方案优于 3D-SPIHT 算法,低比特率情况下尤为突出;与 H. 264 等编码方案相比,在图像质量基本相同的情况下,编码速度是本算法最突出的优势。

参考文献 (References)

1 Ma Hua-dong, Tao Dan. Multimedia sensor network and its research

progresses[J]. Journal of Software, 2006, 17(9): 2013-2028. [马华东,陶丹. 多媒体传感器网络及其研究进展[J]. 软件学报, 2006, 17(9): 2013-2028.]

- Sun Yan, Ma Hua-dong. The QoS guarantee problem for wireless multimedia sensor networks [J]. Chinese Journal of Electronics, 2008, 36(7): 1412-1420. [孙岩,马华东. 无线多媒体传感器网络 QoS 保障问题[J]. 电子学报, 2008, 36(7): 1412-1420.]
- Shapiro J M. Embedded image coding using zero-trees of wavelet coefficients [J]. IEEE Transactions on Signal Processing, 1993, 41(2): 3445-3462.
- Said A, Pearlman W A. A new, fast, and efficient image codec based on set partitioning in hierarchical trees [J]. IEEE Transactions on Circuits and Systems for Video Technology, June 1996, 6(3): 243-250.
- Chen Y, Pearlman W A. Three-dimensional sub-band coding of video using the zero-tree method [A]. In: Proceedings of SPIE Volume 2727 Visual Communications and Image Processing [C], Orlando, FL, USA, 1996: 1302-1312.
- Kim B J, Xiong Z, Pearlman W A. Low bit-rate scalable video coding with 3D set partitioning in hierarchical trees (3-D SPIHT) [J]. IEEE Transactions on Circuits and Systems for Video Technology, December 2000, 10(8): 1374-1387.
- Danyali H, Mertins A. A 3-D virtual SPIHT for scalable very low bit-rate embedded video compression [A]. In: Proceedings of 6th International Symposium on Digital Signal Processing for Communication Systems [C], Sydney, NSW, Australia, 2002: 123-127.
- Ding Wen-qi, Hu Jia, Zhang Li-ming. Optimal 3D-SPIHT video coding method by reducing redundancy between trees [J]. Journal of Computer-aided Design & Computer Graphics, 2005, 17(3): 563-569. [丁文奇,胡佳,张立明. 充分减少树间冗余的优化三维 VSPiHT 视频编码方法 [J]. 计算机辅助设计与图形学报, 2005, 17(3): 563-569.]
- Hong Pan, Siu W C, Law N F. A fast and low memory image coding algorithm based on lifting wavelet transform and modified SPIHT [J]. Signal Processing: Image Communication, 2008, 23(3): 146-161.