易于硬件实现的内嵌图像编码算法

杜伟娜,周 磊,孙 军

(上海交通大学 图像通信与信息处理研究所,上海 200030)

摘要:EZBC 算法综合利用了子带内和子带间系数的相关性,把零树/零块结构和基于上下文编码的优点有机结合在一起,获得了比 SPHIT 算法更好的压缩性能,比 EBCOT 更高的压缩效率。但是 EZBC 算法编码中的两个排序链表需要很大且非固定的存储空间,这使得 EZBC 算法的硬件实现非常困难。在 EZBC 算法的基础上提出了一种易于硬件实现、低存储量、高压缩性能的内嵌零块图像编码算法。该算法利用比特平面节点重要性状态表和上下文查找表来完成整个编码过程和形成上下文。实验结果表明,所提出的算法具有与 EZBC 算法基本相同的高压缩性能,但所需存储空间约为 EZBC 算法的四分之一,所以该算法更易于硬件实现。

关键词:零块编码; 四叉树; 内嵌编码

中图分类号:TN919.81 **文献标识码**:A **文章编号**:1007-2276(2005)03-0352-04

Embedded image coding algorithm for hardware implementation

DU Wei-na, ZHOU Lei, SUN Jun

(Institute of Image Communication & Information Processing, Shanghai Jiaotong University, Shanghai 200030, China)

Abstract: EZBC algorithm combines the advantages of zeroblock/zerotree coding and context modeling of the subband/wavelet coefficients by utilizing the correlation of inter-band and intra-band. In EZBC, the sophisticated context models were designed for coding quadtree nodes at different levels and subbands. Thus, EZBC outperforms SPHIT and can be competitive with EBCOT in compression efficiency. But a large amount of memory is required to maintain two lists that are used to store the coordinates of the quadtree nodes needed to be coded, also a great amount of operations to read and write the memory are required in each coding pass. These become drawbacks for a hardware implementation. An improved EZBC algorithm based on zeroblock and quadtree, with low complexity and high performance is presented in this paper. The improved algorithm utilizes the significance state table of bitplane nodes and the context look-up table to complete the coding passes and form the context, the comparison reveals that the PSNR results of the proposed algorithm are nearly the same performance as that of EZBC, furthermore, the algorithm requires low memory and reduces the implementation complexity.

Key words: Zeroblock coding; Quadtree; Embedded coding

收稿日期:2004-07-16; **修订日期**:2004-09-08

作者简介:杜伟娜(1976-),女,河南洛阳人,博士生,主要从事视频压缩方面的研究。

0 引 言

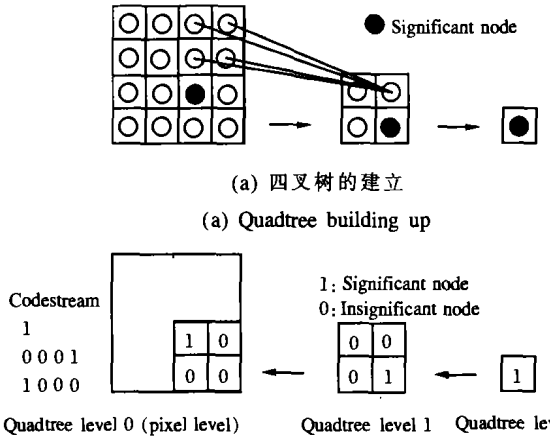
基于零树结构和零块结构的编码算法是两种常用的内嵌小波图像编码算法,如基于分层树集合分割的 SPIHT 算法[1]和基于零块结构的 EBCOT 算法[2,3],目前已在小波图像压缩领域得到了广泛应用。基于零树结构的编码算法通常利用子带间小波系数的相关性,而基于零块结构的编码算法则一般利用子带内小波系数的相关性。Shih-Ta Hsiang 提出的 EZBC 算法[4,5],综合利用了子带内与子带间系数的相关性,从而把零块/零树结构和基于上下文编码的优点有机结合起来,获得了比 SPIHT 算法更好的压缩性能,在不同的比特率下,EZBC 算法的 PSNR 比 SPIHT 高约 0.18~0.87 dB[4]。EZBC 算法采用了简单高效的四叉树编码结构,比 EBCOT 算法具有更高的压缩效率,在不同比特率下,EZBC 算法的每个像素的平均编码长度比 EBCOT 减少约 0.1~1.5 bit[4]。EZBC 算法可广泛应用于数字图像和视频压缩的分级编码[5]中,结合运动补偿的 EZBC 算法则是当今数字电影压缩标准的候选方案之一。

EZBC 算法定义了两个链表:非重要节点链表(LIN)和重要像素链表(LSP),来存储重要像素和非重要节点的坐标值。在编码过程中,两个控制链表需要很大且非固定的存储空间,这些都成为 EZBC 算法硬件实现的最大障碍。本文提出了一种采用比特平面节点重要性状态表和上下文查找表来实现的零块图像编码算法,该算法利用这两个表来完成整个编码过程,所需内存大约为 EZBC 算法的四分之一,能在较低、固定存储空间的情况下获得与 EZBC 基本相同的高压缩性能,且易于硬件实现。

1 改进的零块图像编码算法

首先建立每个子带的四叉树结构,最底层的节点由每个子带的系数幅值构成,称为像素级,上一层四叉树节点由当前层中相应的四个节点的最大值表示,如图 1(a)所示,最上层的四叉树节点表示这个子带中所有小波系数的最大幅值。与传统内嵌图像编码器一样,本文算法也是从最重要的比特 MSB 到最不重要的比特 LSB 渐进编码。如果节点在当前阈值为重要(即节点的幅值不小于阈值),那么将被分裂为四个子节点,此分裂过程递归进行直至像素级,如图 1(b)所示[4,5]。

四叉树中的节点在比特平面编码时根据其重要性加入到相应链表(LSP 或 LIN)中,在 MSB 比特平面编码时,LIN 或 LSP 中的节点数量为零或很少,随着编码比特平面的降低,链表中的节点也逐渐增多,这样就需要大量的存储空间储存节点的坐标信息,并且链表的长度必须动态更新。另外,随着处理图像的复杂度和编码比特率的不同,所需存储空间也不同。



(a) 四叉树的建立

(a) Quadtree building up

(b) 四叉树的分裂

(b) Quadtree splitting

图 1 四叉树分解示意图

Fig.1 Illustration of quadtree decomposition

本文算法建立了比特平面节点重要性状态表,利用 EZBC 算法本身的上下文查找表[5],通过判断节点的重要性状态对需要编码的节点直接查询这两个表,向算术编码器(A.E.)输出相应的编码符号和上下文,而不保留这些节点的坐标信息,这样可以大大减少所需内存,且所需存储空间是固定的。

本文算法描述如下:

(1) 参数定义

$c_k(i,j)$为子带 k 中坐标(i,j)处的小波子带系数;

$Q_k[l](i,j)$为子带 k、l 级中坐标(i,j)处的四叉树节点,其值定义为:

$Q_k[l](i,j)\equiv$

$$\begin{cases} |c_k(i,j)|, & l=0 \\ \max\{Q_k[l-1](2i,2j), Q_k[l-1](2i+1,2j), \\ \qquad Q_k[l-1](2i,2j+1), \\ \qquad Q_k[l-1](2i+1,2j+1)\}, & l>0 \end{cases}$$

D_k 为子带 k 的四叉树级数；

$\chi_k(i,j)$为 $c_k(i,j)$的符号位；

$S_n(i,j)$为判断节点在比特平面 n 的重要性状态的函数。如果 $Q_k[l](i,j)\geq 2^n$ ，则 $S_n(i,j)=1$，否则 $S_n(i,j)=0$。

$\sigma_k[l](i,j)$为标记节点 $Q_k[l](i,j)$的重要性状态。如果 $Q_k[l](i,j)$重要，那么 $\sigma_k[l](i,j)=1$，否则 $\sigma_k[l](i,j)=0$。

(2) 初始化

输出 $n=\left\lfloor \log_2 \left(\max_{(k,i,j)}\{|c_k(i,j)|\}\right)\right\rfloor$，$\sigma_k[l](i,j)$置零。

(3) 重要性编码

对于 $\sigma_k[l](i,j)=0$ 且 $\sigma_k[l+1](\lfloor i/2, j/2 \rfloor)=1$ 的节点，输出 $S_n(i,j)$；

若 $S_n(i,j)=1$，则更新 $\sigma_k[l](i,j)=1$；若 $l=0$，输出 $\chi_k(i,j)$，否则进行四叉树分裂：QuadtreeCode(i,j,l)。

(4) 幅值细化

对于 $\sigma_k[0](i,j)=1$ 且 $|c_k(i,j)|\geq 2^{n+1}$ 的系数，输出 $c_k(i,j)$在比特平面 n 处的比特值。

(5) QuadtreeCode(i,j,l)

对于属于 $l-1$ 级的 $(x,y)\in\{(2i,2j),(2i,2j+1),(2i+1,2j),(2i+1,2j+1)\}$，输出 $S_n(x,y)$；

若 $S_n(x,y)=1$，则更新 $\sigma_k[l-1](x,y)=1$；若 $l=1$，输出 $\chi_k(x,y)$，否则进行四叉树分裂：QuadtreeCode$(x,y,l-1)$；

(6) 比特平面更新

n 减 1，转至(3)。

2 实现结构

非固定的链表长度和所需的大量内存使 EZBC 硬件实现比较困难。本文提出一种改进算法：利用一个比特平面节点重要性状态表和上下文查找表来完成编码过程，设计了编码器的实现结构，克服了 EZBC 算法的不足，如图 2 所示。首先是符号位的输入和四叉树结构的建立。符号位寄存器把在不同方向的符号位贡献和符号位本身传送到相应编码单元。在编码器实现架构中采用状态变量存储器来存储形成上下文所需的节点重要性状态，利用状态变量寄存器来加载状态变量、更新状态变量并将邻近节点在不同方向(H,V,D)的重要性状态、子带间节点的相关性(P) 以及四叉树中四个子节点的重要性(σ_{sb})[5]传送到相应处理模块，然后通过比特平面节点重要性状态表，输出算术编码器所需的编码符号。算术编码器所需要的上下文直接通过上下文查找表获得。上下文模型利用了同一四叉树级内邻近节点和父子带中相同空间位置节点的重要性状态，这些复杂的上下文模型可以有效地对四叉树结构编码[4]，获得高的压缩性能。

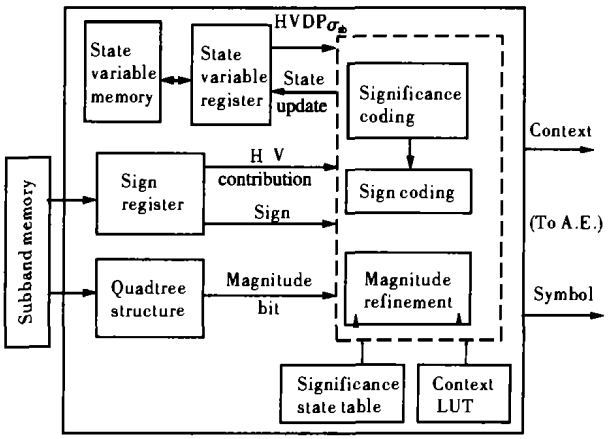


图 2 实现结构的框图

Fig.2 Diagram of the proposed architecture

3 实验结果

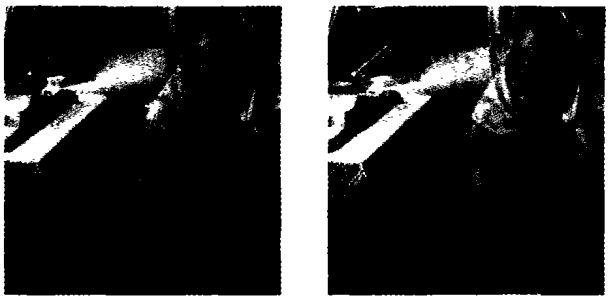
表 1 给出了本文算法和 EZBC 算法的 PSNR 比较，

表 1 本文算法和 EZBC 算法的 PSNR 性能比较

Tab.1 Peak signal to noise ratios of proposed algorithm and EZBC **dB**

Image	Rate bit/pixel	EZBC	Proposed algorithm
Barbara(512×512)	1.0	37.15	37.17
	0.5	32.07	32..08
	0.25	28.18	28.20
Bike(2048×2560)	1.0	38.22	38.20
	0.5	33.51	33.52
	0.25	29.55	29.55

仿真结果是在本文算法和 EZBC 算法均未进行上下文相关性反量化[5]的条件下得到的，若进行反量化处理，那么这两种算法的 PSNR 均可提高大约 0.35 dB。测试图像为 512×512×8 bit 标准灰度图像 Barbara 和 2048 ×2560×8 bit 标准灰度图像 Bike,变换部分采用了 5 级 Daubechies 9/7 小波变换。图 3 为采用本文算法对 Barbara 压缩 32 倍前后的主观效果比较。实验结果表明本文算法具有和 EZBC 基本同样高的压缩性能。



(a) 原始图像　　(b) 恢复图像

(a) Origenal image　　(b) Restored image

图 3 Barbara 在比特率为 0.25 bit/pixel 下的主观效果比较

Fig.3 Comparison of original Barbara and the proposed algorithm at 0.25 bit/pixel

EZBC算法通过 LIN 链表和 LSP 链表完成编码过程并利用相邻节点和父子带节点的重要性来形成上下文，所以 EZBC 算法需要大量的内存来存储链表。经过实验仿真,256 灰度级、512×512 的图像,在 1 bit/pixel 的压缩条件下,EZBC 的 LSP、LIN 链表各自需要大约 140 和 50 K 字节;256 灰度级、2048×2560 的图像,在 1 bit/pixel 的压缩条件下,LSP、LIN 链表各自需要大约 3250 和 1150 K 字节,并且其所需要的内存随着处理图像复杂度和编码比特率的不同而不同,这就增加了硬件实现的复杂度。本文算法仅利用一个节点重要性状态表和查找表就可完成编码过程和形成上下文,大大减少了所需内存量,经过实验仿真,256 灰度级、512×512 的图像,本文算法需要 40 K 字节;256 灰度级、2048×2560 的图像,需要 880 K 字节,并且在处理相同分辨率的图像时所需内存固定,与编码比特率无关。本文算法和 EZBC 算法的内存需求比较如表 2 所示,可见本文算法更易于硬件实现,是一种低复杂度的编码算法。

表 2 本文算法和 EZBC 算法编码所需存储空间比较

Tab.2 Memory comparison algorithm of proposedalgorithm and EZBC　　byte

Image resolution	EZBC	Proposed algorithm
512×512	190 K	40 K
2048×2560	4400 K	880 K

4 结 论

EZBC 算法充分利用了子带内和子带间小波系数的相关性以及简单高效的四叉树编码结构,从而获得了很高的压缩性能和压缩效率。本文算法是在 EZBC 算法基础上提出的一种改进的内嵌图像编码算法,利用了比特平面节点重要性状态表和上下文查找表来完成编码过程和形成算术编码器所需要的上下文,这样不但可以获得和 EZBC 算法基本一致的压缩性能还有利于硬件实现。

参考文献:

[1] Said A, Pearlman W A. A new, fast, and efficient image codec based on set partitioning in hierarchical trees[J].IEEE Trans on Circuits and System for Video Technology,1996,6(3):243-249.

[2] Taubman D. High performance scalable image compression with EBCOT[J].IEEE Trans on Image Processing,2000,9(7):1158-1170.

[3] Rabbani M,Joshi R. An overview of the JPEG 2000 still image compression standard[J].Signal Processing:Image Communication, 2002,17(1):3-48.

[4] Hsiang Shih-Ta,John W Woods. Embedded image coding using zeroblocks of subband/wavelet coefficients and context modeling [A].IEEE Int Symposium on Circuits and System(ISCAS 2000)[C]. Switzerland:Geneva,2000,3.662-665.

[5] Hsiang Shih-Ta.Highly scalable subband/wavelet image and video coding[D].New York:Rensselaer Polytechnic Institute,2002.