

基于多级查询表的 JPEG2000 位平面扫描优化方法

朱悦心, 付 昀, 吴宗泽, 郑南宁

(西安交通大学人工智能与机器人研究所, 陕西西安 710049)

摘 要: 针对 JPEG2000 位平面编码的 VLSI 实现, 提出一种基于多级查询表的扫描优化方法. 该方法以树型查找表为基准, 在有效性传播通道中提前记录编码必要性信息, 实现了后续通道的启发式扫描. 与 K Chen 等人提出的 GOCS 方法比较, 不仅消除了扫描中出现的冗余组, 而且消除了冗余列, 进一步提高了扫描速率. 该方法的另一优点是可以完成通道扫描和状态变量的并行刷新, 支持 Y Hsia 存储优化技术而不影响系统整体效率. 实验结果表明, 在 VLSI 设计中引入该方法, 位平面编码的扫描时间可缩短 25%, 存储资源将节省 8.72%.

关键词: JPEG2000; EBCOT; 位平面; 编码通道; 上下文; 状态变量

中图分类号: TP391 文献标识码: A 文章编号: 0372-2112 (2004) 05-0810-04

A JPEG2000 Optimized Bi2Plane Scanning Method Based on Multilevel Query Table

ZHU Yuexin, FU Yun, WU Zongze, ZHENG Nan2ning

(Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China)

Abstract: Aiming at VLSI implementation of the bi2plane encoding in JPEG2000, a new structure, CBQT (Column2Based Query Table), and its corresponding scanning method, are presented. The main idea is to record the NBC (Next2Block2Coding) messages of pass 2 and 3 in advance by a certain polling tree, so heuristic search can be fulfilled in the subsequent pass2scanning. Compared with K. Chen's GOCS scheme, this method achieved the single2column2skipping. Another advantage is that CBQT could manipulate pass2coding and stat2updating simultaneously, so Y. Hsia's memory saving technique was imported without affecting system's coding efficiency. Experimental results show that whole time of bi2plane scanning is reduced by 25% of the original, and 8.72% required memory space is curtailed as well.

Key words: JPEG2000; EBCOT; bi2plane; coding pass; context; state variable

1 引言

位平面编码是一种基于将多灰度值图像分解成一系列二值图, 然后对每一幅二值图再进行二元压缩的技术. 新一代静止图像压缩标准 JPEG2000 中的 EBCOT 算法就采用了这种技术, 其表现方式为通过位平面编码器生成上下文, 再基于上下文进行自适应算数编码. 位平面编码器的作用是提供熵编码所必需的上下文和位流, 相对于 JPEG2000 中的其它运算单元, 具有较高的复杂度, 另外由于通道扫描的低效性, 使这里往往成为硬件实现的一个难题.

针对这个问题, K Chen 等人提出了 GOCS (Group of columns Skipping) 扫描方案^[4], 将位平面上的 8 个相邻列编为一组. 通过提前记录该组编码必要性信息的方式, 对后续通道进行启发式扫描. 这种方案的确使原有扫描方法得到了很大改进, 也拓宽了人们的思路, 但是它没有解决组内的冗余搜索问题, 在

组内仍然需要对每一列依次进行处理. 本文提出一种基于多级查询表的顺序扫描法, 不仅记录每一组的编码必要性信息, 组内每一列的信息也会被标明, 各级查询表之间采用树型结构. 用这种方法进行 VLSI 设计, 可以更有效的去除冗余搜索, 从而使扫描效率得到显著提高; 该方法的另一大特点是支持 Y Hsiao 等人提出的状态变量优化技术^[5], 这使得编码必要性信息的提取更加快速便捷, 优化后状态变量的刷新也可以借助多级查询表与编码过程并行操作, 节省了一个位平面的刷新周期. 实验结果表明, 采用本设计, 位平面的扫描时间可以缩短到未优化前的 75% 以下, 速度比 GOCS 方法提高了 5%, 而存储空间可以削减 81.72%.

2 通道扫描的基本方法

JPEG2000 的位平面编码器是在扫描过程中实现编码的. 因此扫描的速率和质量直接决定了编码器的效率.

扫描按照 Z 字型顺序进行, 其优先顺序是: 带 y 列 y 点. 连续的 4 行小波系数被划分为一个带(stripe), 在一个带内垂直方向上的 4 点被认为是一个编码列(column). 每一个位平面都需要顺序扫描三次, 分别对应三种通道编码形式. 这三种通道为: (1) 有效性传播通道(Pass1, Significance propagation pass): 位平面上的像素点进入 Pass1 的条件为: 像素 8 邻域的当前有效性状态至少有一个为 1^X. 由于有效性状态在编码过程中被不断刷新, 因此该通道的编码过程可以认为是像素有效性的传播过程; (2) 幅度细化通道(Pass2, Magnitude refinement pass): 上一个位平面有效性状态为 1 的那些点, 在当前位平面进入幅度细化通道进行编码; (3) 清除通道(Pass3, Cleanup pass): 当前位平面被 Pass1 和 Pass2 过滤后的残余点, 进入通道 3 进行清除通道编码.

每一次扫描都必须遍历位平面上的所有列. 协议规定: 第一个有效位平面只进行清除通道扫描, 后面的位平面则依次进行有效性传播、幅度细化和清除通道三次扫描. 因此对于一个有效位平面个数为 n, 大小为 x@y 的码块, 需要进行通道扫描的次数为 3n-2. 整个编码过程中需要遍历的总列数为 $[x@y@((3n-2)/4)]$.

3 基于多级查询表的通道扫描方法

JPEG2000 协议给出的通道扫描方法, 虽然简单直接, 但扫描冗余度过大, 效率很低, 不利于实现 VLSI 设计, 必须进行改进. J Chiang 等人曾提出一种三通道并行扫描方案^[3], 使用多个窗口编码器对三个通道同时进行扫描. 这种方法的优点是速度快, 不需要进行多余的查询运算, 但未考虑到通道内各像素间的分布特征, 因此硬件利用率不高. 由于通道扫描过程本身是顺序关系, 所以多个窗口编码器之间的相位差不是很容易控制. 另外这种方式下的输出码字是乱序的, 需要整理才可送入后级.

另一种具有代表性的方法是采用启发式顺序扫描. K Chen 等人注意到通道像素非均衡分布的特征, 采用组跳跃(GOCS, Group of column skipping)方式, 直接跳过不需要编码的组(即 8 个相邻列), 从而在很大程度上削减了冗余扫描, 且实现时只需要增加很少的存储资源. 这种方法的缺点是无法进行组内跳列扫描, 在组内仍然要顺序判断每一列的编码必要性.

另外, 还有一种状态变量优化技术, 由 Y Hsiao 等人提出. 使用中可以节省 20% 的存储资源, 但需要付出的代价是增加一个位平面的刷新周期. 这种资源优化技术一般很难与速度优化技术配合使用, J Chiang 和 K Chen 的方法都不能很好支持. 进行状态变量优化而又不影响编码速率造成太大影响, 唯一的办法就是并行处理通道扫描与状态刷新.

针对以上几种现有扫描机制的缺点, 本文提出一种基于多级编码查询表的扫描优化方法.

3.1 单级查询表扫描优化

一般情况下, 位平面编码首先进入有效性传播通道(Pass1), 因此 Pass1 扫描中可以先行判断当前列在后续通道中的编码必要性(Need block coding, NBC). 将此必要性标志写

入列编码查询表(Column based query table, CBQT), 这样在进行 Pass2 和 Pass3 扫描时, 可以通过查询 CBQT 得知当前列是否需要编码, CBQT 记录为空的列直接跳过, 不需要花费时间进行 NBC 判断. 此表在扫描过程中需要不断进行更新和维护, 但在流水设计中, 更新可与编码并行工作(在任一时间片内, 进行本列的编码和前一系列的更新), 因此并不会影响运行效率. 具体过程参见图 1.

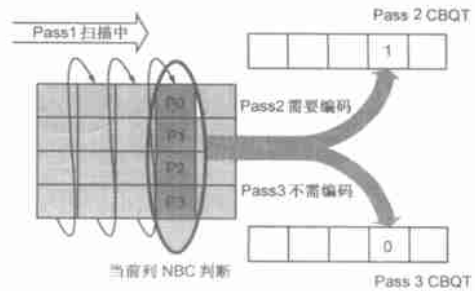


图 1 Pass1 扫描中向 CBQT 写入后续通道 NBC 信息

CBQT 中的表项对应位平面上的列(4 像素). 有效性传播通道扫描结束时, Pass2 和 Pass3 的 CBQT 已经生成(第一个位平面由于只进行清除通道扫描, 因此初始化时需要将 Pass2 的 CBQT 全部清 0, Pass3 的 CBQT 全部置 1). 后续通道编码中, 先查询 CBQT, 只有当前列对应状态为 1 的情况下才进行处理. 该列 4 个像素点通道编码全部完成后, 将 CBQT 的 NBC 信息置零. 这种模式下, Pass2 和 Pass3 扫描中所出现的大量冗余列只需要花费一个时钟的处理时间来读取 CBQT 表, 效率得到较大提高.

3.1.2 多级查询表扫描优化

只使用 1 级 CBQT 进行通道扫描, 每列仍然需要花费至少 1 个时钟的处理时间. 可以采用多级 CBQT 进一步提高效率.

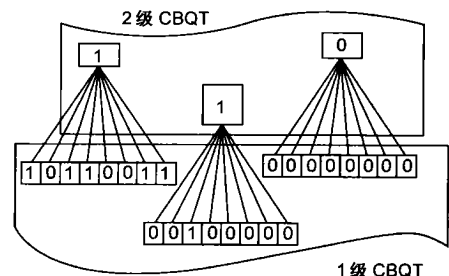


图 2 多级 CBQT 之间的拓扑关系

多级 CBQT 采用 K Chen 等人提出的 GOCS 扫描思想, 不同之处在于高级 CBQT 完全由低级按照树结构自动回溯生成, 建表不需要花费系统时间; 而 K Chen 方法没有使用低级查询, 所以必须在 pass1 扫描时生成 GOCS, 只支持组旁路, 不支持列旁路. 2 级 CBQT 的处理流程是这样的: 将位平面上连续的 8 列编为一组(group), 每组与一个 2 级表项相对应. 组中任何一列如果在后续通道需要编码, 则将 2 级对应表项置 1. 2

^X 且该像素自身的有效性状态为 0.

级 CBQT 蕴含的所有信息可以完全由 1 级获取,这样就形成一个树型结构,高级状态由低级通过组合逻辑自动维护和更新(见图 2). Pass2 和 Pass3 扫描时,首先查询 2 级 CBQT,遇到 0,说明该组所有列都不需要编码,直接跳过;若出现 1,说明该组有需要编码的列,进入 1 级 CBQT 进行 2 次查询,直到确认当前列需要编码才进入通道编码流程,因此所有的无效列均被忽略.完整的通道扫描流程参见图 3.

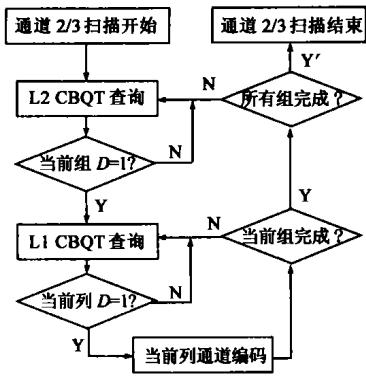


图 3 加入 2 级 CBQT 后的通道扫描流程

4 在多级查询表中应用状态变量优化技术

多级查询表除了可以更好的降低扫描冗余度之外,与 J Chiang 和 K Chen 方法相比,还有一个特点:能够非常方便的完成通道扫描与状态刷新之间的并行处理,因此支持 Y Hsiao 等人提出的状态变量优化技术^[5],使得在扫描时间显著缩短的同时存储空间也相应节约,扫描效率和资源利用率同时得以提高.

4.1 状态变量的优化

按照 JPEG2000 协议和 Taubman 的思想^[2],需要如下五个状态变量才能描述位平面上一个点的全部信息:

v: 在当前位平面上的数据位; s: 在当前位平面上的符号位; R: 在有效性平面上的当前值; P: 是否已被前一个通道处理过; N: 是否是第一次进入幅度细化通道.

采用这种方式,对于 64@64 大小的码块,所需的存储空间共有: 64@64@5= 20Kbit.

事实上,考虑到这五种状态信息之间的相关性,可以将状态变量的个数优化为 4 个.令这 4 个新的状态变量为: vc、sc、Rc 和 Pc,它们分别表示如下含义:

vc: 与 v 意义相同; sc: 与 s 意义相同; Rc: 上一个位平面全部通道扫描结束时有效性平面上的遗留值.也就是说,当前位平面扫描过程中该值是不发生变化的; R: 与 P 的意义基本相同.区别只在幅度细化通道.如果对一个点进行编码时发现 Pc 的值为 1,那么表示该点是第一次进入幅度细化通道.

原来未经优化的 5 个状态变量可以用 vc、sc、Rc 和 Pc 表示为:

$$\begin{aligned}
 v &= vc \\
 s &= sc \\
 R &= Rc | (vc \& Rc) \\
 P &= \begin{cases} Pc, & (\text{pass}3) \\ \text{don't care}, & (\text{pass}1 \text{ or } \text{pass}2) \end{cases} \\
 N &= \begin{cases} Rc, & (\text{pass}2) \\ \text{don't care}, & (\text{pass}1 \text{ or } \text{pass}3) \end{cases}
 \end{aligned}$$

位平面扫描结束后,必须对变量 Rc 与 Pc 进行刷新:

$$R_{n+1} = R_n | vc_n \quad (1)$$

$$P_{n+1} = (! R_n) \& vc_n \quad (2)$$

用这种表示方法,可以减少一个状态变量,节约 20% 的存储开销.但是必须对优化后的变量进行状态刷新.

4.2 借助查询表刷新状态变量

CBQT 在通道编码的同时,可以并行完成优化后状态变量的即时刷新.

通过观察我们发现,当一列小波系数在特定通道内完成编码后,如果对其状态信息进行强制刷新,所带来的影响无外乎有以下两点: (1) 后续通道也可能扫描本列,这时 Rc、Rc 值已被刷新,先前值被覆盖,幅度细化与清除通道 NBC 可能判断错误; (2) 本列被其他列作为邻域引用,此时 Rc 值更新不会带来任何问题, Rc 刷新后已变为 Taubman 意义下的值,会导致有效性状态引用错误.

我们可以通过下面的步骤加以解决: (1) 一列编码结束且 CBQT 列表更新后,观察本列后续通道的 CBQT 列表,如果为 1,说明后续通道还要处理本列,此时不刷新;否则刷新; (2) 邻域引用时,观察邻列的 MR2CBQT 和 CLUP2CBQT,如果邻列在两表中对应项均为零,说明该列必被刷新,直接使用 Taubman 意义下的 R 值进行有效性判断,否则用 Rc | (vc&Pc) 进行判断.

这种方式下,状态信息的刷新可以和通道编码过程并行处理,扫描结束时,刷新工作也同时结束.在重要性传播通道扫描中,由于后续通道是幅度细化和清除,因此 MR2CBQT 和 CLUP2CBQT 都要考虑;在幅度细化通道中,只需考虑 CLUP2CBQT;而对于清除通道,由于已经没有后续通道,因此全部进行刷新.

4.3 用优化后的状态变量提取查询表所需的 NBC 信息

CBQT 表所带来的好处显而易见,然而必须有一种方法能够在通道 1 扫描时就提前获取通道 2 和通道 3 的 NBC 信息. NBC 可以借助组合逻辑通过很多种途径来得到,但复杂度不尽相同.使用优化后的状态变量提取 NBC 是一种较为便利的方法,这种方法可以使生成 NBC 所需的组合逻辑比较简单.

在 Pass1 的扫描过程中,我们可以利用优化后的状态变量直接提取出 Pass2 和 Pass3 的 NBC 信息.假设 Pass1 其中一列像素已经扫描完成,得到 NBC 向量: $K = [K_1 \ K_2 \ K_3 \ K_4]^T$. (当前列的第 i 个点(1 ≤ i ≤ 4) 如果属于 Pass1,则 $K_i = 1$,否则 $K_i = 0$).再考虑前面我们提到的状态变量: $Rc = [Rc_1 \ Rc_2 \ Rc_3 \ Rc_4]^T$.由 K 和 Rc,我们可以直接得到 Pass2 和 Pass3 在当前列的 NBC 信息:

$$D_{\text{pass}2} = Rc_1 | Rc_2 | Rc_3 | Rc_4 \quad (3)$$

$$D_{\text{pass}3} = \sim ((K_1 | Rc_1) \& (K_2 | Rc_2) \& (K_3 | Rc_3) \& (K_4 | Rc_4)) \quad (4)$$

将 $D_{\text{pass}2}$ 和 $D_{\text{pass}3}$ 写入 Pass2 和 Pass3 的 CBQT,这样在后续通道扫描时即可参照.

5 实验结果及其分析

为了验证 CBQT 扫描优化方法的效果,我们用 Lena, air2 plane 等几幅图像进行了实验,并与 K Chen 等人的 GOCS 扫描优化方法做了比较.由于通道编码时间是不确定的,因此我们

表 1 通道扫描所需要的时钟数(单位: clk)

输入图像	原始扫描时间	K Chen 等人 GOCS 优化后的扫描时间	2 级 CBQT 优化后的扫描时间	GOCS 优化法扫描时间百分比	2 级 CBQT 优化法扫描时间百分比
Lena	2, 682, 880	2, 108, 080	1, 956, 946	78. 6%	72. 9%
Airplane	2, 652, 160	2, 059, 744	1, 954, 486	77. 7%	73. 7%
Baboon	2, 744, 320	2, 115, 312	1, 976, 442	77. 1%	72. 0%
Boats	2, 529, 280	2, 022, 320	1, 856, 898	80. 0%	73. 4%
Peppers	2, 652, 160	2, 111, 984	1, 944, 860	79. 6%	73. 3%

只考虑通道扫描时间(通常扫描一行并判断编码必要性需要 10 个时钟, 查询 CBQT 表一次需要 1 个时钟)。

所有输入图像为 256@256 的灰度图像, 码块大小 64@64, 9/7 小波分解 2 级。实验结果见表 1。三种方式下位平面编码所需要的存储资源见表 2(码块大小 64@64)。

表 2 位平面编码所需的内部存储资源统计(单位: bit)

	状态变量存储器	附加资源	总存储开销
无 CBQT	20K/16K*	0	20K/16K*
GOCS 优化	20K	256	20K+256
2 级 CBQT+ 状态变量优化	16K	2K+256	18K+256

* 注: 状态变量优化后可以减少 4K 的存储空间, 但是扫描速率会变得更慢。

由表 1 和表 2 可以看出: 本文提出的 2 级 CBQT 通道扫描方法比 K Chen 等人的 GOCS 扫描法效率更高, 扫描时间已经压缩到基本方法的 75% 以下, 比 GOCS 法缩短了 5% 左右。而存储空间由于引入状态变量优化技术也节约了 1-18, 256/20, 000=81.72%。若进一步提高列内 NBC 先行判断法^[4], 则每一列的编码时间也可以缩短。同时对通道扫描和通道编码进行优化, 可以取得更好的效果。

6 结论

本文针对 JPEG2000 位平面编码器扫描速率过慢的问题, 提出一种基于多级查询表(CBQT)的通道扫描优化法。这种方法比 GOCS 扫描法效率更高, 并且支持状态变量优化技术。实验结果表明: 将该方法应用于 VLSI 设计, 可以使扫描时间缩短到原来的 75% 以下, 存储开销削减 81.72%。若在优化通道扫描的基础上进一步优化通道编码, 则位平面编码器的效率可以得到进一步提高。

参考文献:

- [1] Boliek M, Christopoulos C, Majani E. ISO/IEC JTC1/SC29/WG1 N1890, JPEG2000 Part I final draft international standard[S]. ISO/IEC JTC1/SC29/WG1, N1890, 2000.
- [2] D Taubman. High performance scalable image compression with EBCOT[J]. IEEE trans Image Processing, 2000, 9(7): 1158-1170.
- [3] Jer-Shiun Chiang, Yu-Sen Lin, Chang-Yo Hsieh. Efficient Pass-Parallel Architecture For EBCOT In JPEG2000[A]. IEEE International Symposium on Circuits and Systems(ISCAS 2002)[C]. Arizona: IEEE Circuits and Systems Society Press, 2002(1): 773-776.
- [4] Kuan-Fu Chen, Chung-Jr Lian, et al. Analysis and architecture design of EBCOT for JPEG2000[A]. The 2001 IEEE International Symposium on Circuits and Systems(ISCAS 2001)[C]. Sydney, Australia: IEEE Circuits and Systems Society Press, 2001. 5(2): 765-768.
- [5] Yun-Tai Hsiao, Hing-Der Lin, et al. High-speed memory-saving architecture for the embedded block coding in JPEG2000[A]. IEEE International Symposium on Circuits and Systems(ISCAS 2002)[C]. Arizona: IEEE Circuits and Systems Society Press, 2002. (5): 133-136.

作者简介:



朱悦心 男, 1978 年 12 月出生于陕西省西安市, 西安交通大学人工智能与机器人研究所硕士研究生, 主要研究方向为模式识别与智能系统、图象压缩编码算法与 ASIC 设计。



付 昀 男, 1979 年出生于陕西西安, 西安交通大学人工智能与机器人研究所硕士研究生, 主要研究方向为计算机视觉与图像图形学、模式识别与智能系统、人像处理与人脸识别、视频图像处理等已发表学术论文 10 余篇。

吴宗泽 男, 1975 年出生于重庆西安交通大学人工智能与机器人研究所博士生, 研究方向为图像压缩与 ASIC 设计。