

位平面算术编码的编码分离与 MQ 编码器改进

王继东 鲁 琴 罗武胜 肖学敏

(国防科技大学机电工程与自动化学院, 长沙 410073)

摘 要 为了降低图像压缩中位平面算术编码的计算复杂度,提出了将算术编码从位平面扫描中分离,方便编码的并行计算。以 JPEG2000 的位平面编码为例,实现了位平面扫描和 MQ 编码的分离,然后根据编码分离后的算法特点,设计了一种改进的 MQ 编码器。对该算法在 TMS320C6000 系列 DSP 中的实现进行了研究,通过软件流水、循环展开、条件操作和优化选项等方法提高处理速度。编码独立和编码器改进相结合,使重建图像质量稍有提高,编码速度提高了 8 倍以上。

关键词 图像压缩 算术编码 并行计算 编码分离 MQ 编码器改进

中图分类号: TN919.8 **文献标识码:** A **文章编号:** 1006-8961(2007)10-1806-05

Isolation Coding of Bit Plane Scanning Arithmetic Coding and Modified MQ-coder

WANG Ji-dong, LU Qin, LUO Wu-sheng, XIAO Xue-min

(College of Mechatronics Engineering and Automation, National University of Defense Technology, Changsha 410073)

Abstract To lower the complexity of bit plane scanning arithmetic coding in image compression, arithmetic coding can be isolated from bit plane scanning for the purpose of parallel computing. The MQ-coding in JPEG2000 is isolated from the bit plane scanning. Based on the characteristic of coding algorithm after isolated coding, a modified MQ-coder is designed. Also, this paper studies the implementation in TMS320C6000 DSP of the modified algorithm. The processing speed is improved through software pipelining, loop unrolling, conditional instruction and compiler options. The coding performance of arithmetic coding is superior and its speed is improved by eight times due to the implementation of isolation coding and modified MQ-coder.

Keywords image compression, arithmetic coding, parallel computing, isolation coding, modified MQ-coder

1 引言

算术编码在数字图像压缩中的地位举足轻重^[1,2]。它最主要的优点是输出的码长能逼近信源的熵,并因此广泛应用于各种压缩算法,典型的有: DCT 变换的编码算法^[3]、SPIHT、JPEG 的扩展系统以及新的静止图像压缩标准 JPEG2000。目前,基于上下文的自适应二进制算术编码(context based

adaptive binary arithmetic coding, CABAC)成为主流。CABAC 常和位平面分解相结合,编码从最高有效位平面开始,到最低有效位平面结束,按一定的顺序扫描每个位平面。对每个位确定一个上下文,根据上下文估计概率,然后将位和它的概率传递给算术编码器。

由于 CABAC 编码过程中要用到大量的码表、上下文模型以及中间变量等,它的复杂度比较高,而且需要更大的存储空间。同时,由于编码一般在位

基金项目:国家高技术研究发展计划 863 项目(2006AA701121)

收稿日期:2007-07-05;改回日期:2007-07-19

第一作者简介:王继东(1976~),男。现为国防科技大学测试计量技术及仪器专业博士研究生。研究方向为现代传感技术与系统。

E-mail: wangjd2370@sina.com

平面扫描过程中进行,编码过程依赖已有位及其概率,很难用并行算法优化。因此,本文提出了位平面算术编码的编码分离方法,位平面扫描过程中缓存编码位的上下文模型和概率(实际保存 0、1 计数),扫描完成后集中进行算术编码。由于待编码位的相互独立,适合于算术编码的并行处理。并且本文根据编码分离后的算法特点,提出了一种改进的 MQ 编码器实现。

2 位平面算术编码的编码分离

并行计算是指同时使用多种计算资源解决计算问题的过程。为利用并行计算,通常将工作分离成离散部分,有利于同时解决,以便随时并及时地执行多个程序指令。

立足于算法的硬件实现,将任务分离的着眼点放在计算过程中的循环结构,由于现代微处理器的强大处理能力,单次计算耗时可以忽略。在循环迭代中,计算任务具有多样性,呈现出不同的复杂度和相关性。经常一些不相关的计算任务处于复杂漫长的串行计算链中,无法进行并行处理。

此时希望将这些不相关的计算任务人为地分离出来,集中处理,方便处理器使用软件流水的方式并行执行。

以 JPEG2000 为例,编码时将小波子带划分成码块(code-block),以码块为单位进行编码。与传统的依次对每个系数进行熵编码不同,JPEG2000^[4]把量化后的系数组织成二进制位平面,从最高有效平面开始,依次对每个位平面的小波系数位进行算术编码,由此得到嵌入式的压缩码流。编码时,为减小所需的缓存空间,对码块中的小波系数按规定顺序进行扫描。由于侧重点不同,这里对每个比特平面编码的扫描顺序和 3 个编码过程不进行详述,编码过程中对每一位编码包括以下 3 个步骤:

- (1) 确定它的上下文;
- (2) 估计上下文的概率;
- (3) 将位和它的概率传送给算术编码器。

简化的位平面编码流程如图 1(a) 所示。考虑将位编码的第 3 个步骤分离出来,第 2 步只要按照第 1 步确定的上下文对该模型下的 0 或 1 进行计数,扫描完成后各上下文的概率可以根据计数的结果得到。分离后的位平面编码流程如图 1(b) 所示。

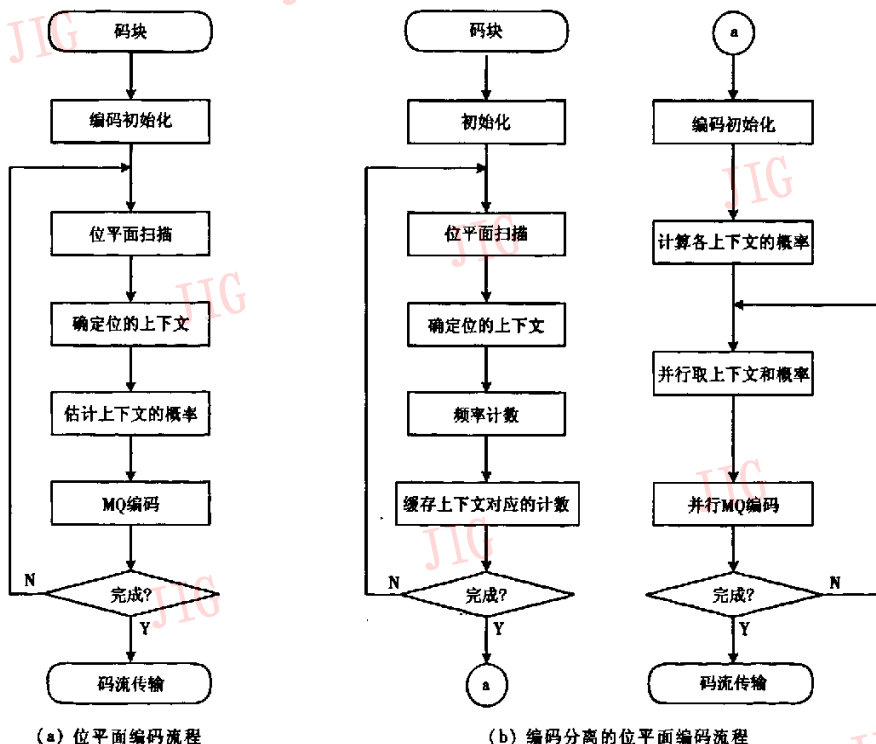


图 1 编码独立前后的位平面编码流程

Fig. 1 Bit plane coding flow of pre-and post-isolate coding

3 改进的 MQ 编码器

MQ(多路量化)编码是一种用于二进制数据的自适应算术编码,是 Pennebaker 等人于 1988 年提出的无乘法器的 Q 编码算法的改进,增加了条件交换机制和概率估计状态机,采用原始 Q 编码的位填充策略。MQ 编码器是 Ueno 等人于 1999 年专门为 JPEG 2000 标准设计的。MQ 编码器的输入是上下文(context, CX)和数据决策(decision, D),输出是压缩数据(compression data, CD),如图 2 所示。

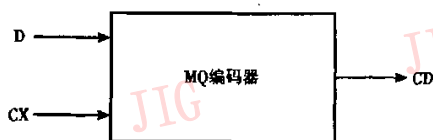


图 2 MQ 编码器的输入和输出

Fig. 2 Input and output of MQ-coder

由于编码分离以后概率估计的不同,结合硬件实现的特点,对 MQ 编码器进行了改进。

3.1 保留乘法运算

在 JPEG2000 的算术编码中,输入数据流中的信源符号被分成大概率符号(MPS)和小概率符号(LPS),把 LPS 的概率记作 Q_s 。通过及时的重整(renormalization)操作,编码间隔 A 始终保持在区间 $[0.75, 1.5]$ 之间。这意味着,如果四舍五入的话,编码间隔总是约等于 1。因此, $(Q_s \times A) \approx Q_s$, 算术编码的迭代公式进行相应的简化。

目前高性能的 DSP、ARM 等处理器均有完备的乘法运算指令, FPGA 等逻辑器件亦具有乘法操作能力,为硬件实现而进行的简化操作不再需要。实际上,在 TMS320C6000 系列 DSP 的指令中,乘法指令和其他加法、逻辑操作指令占用不同的功能单元,不同功能单元的指令可以并行,程序中希望各种功能单元的指令平衡分配。因此,保留乘法操作有时反而可以提高运算速度。编码依然按照如下规则进行迭代,其中 C 为码字:

如果输入符号是 MPS: TRUE, 则

$$C = C + (Q_s \times A), A = A - (Q_s \times A)$$

如果输入符号是 MPS: FALSE, 则

$$C = C, A = A \times Q_s$$

3.2 概率估计

在位平面扫描后独立的编码过程中,不需要进

行信源符号的互换,始终约定 MPS 为 1, LPS 为 0。也不需要查表方式得到 Q_s 的值,在位平面的扫描的过程中,记录了每种上下文模型的二进制数据 0、1 的个数。扫描完成后,通过归一化可以准确得到各个上下文模型中 0、1 的概率。即:

$$p(0|s) = n(0,s)/n(s)$$

$$p(1|s) = n(1,s)/n(s)$$

其中, $n(0,s)$ 和 $n(1,s)$ 分别表示二进制输入数据流 s 中 0 和 1 的个数,而 $n(s) = n(0,s) + n(1,s)$ 。如果是定点处理器,可以将概率选择统一的比例因子进行定标,转化为定点数,用定点运算指令完成小数运算。

3.3 有限精度

随着算术编码的进行, C 和 A 中的小数的位数将会越来越多,有限的寄存器位数使编码难以向下进行。为了解决这个问题,采用了移位重整算法。在 JPEG2000 中,十六进制整数 0x8000 代表小数 0.75。当编码间隔 A 小于 0x8000 时,通过 A 的左移可以使 A 保持在区间 $[0.75, 1.5]$ 。这样就可以用固定位数的寄存器来存储 A,解决了有限精度的问题。

为了保持 A 和 C 有统一的尺度,每次 A 左移一位, C 也左移一位。因为 C 中存储的是编码的结果,需要有一个缓冲区来保存从 C 中移出的数据位。使用寄存器 B 指向 C 寄存器的高位,每当 C 寄存器满,就从 B 寄存器把一个字节的数据输出。在 JPEG2000 中,初始 $A = 0x8000$ 、 $C = 0$ 、移位计数器 $CT = 12$, B 寄存器每次输出一个字节的数据。

为了充分利用硬件资源和提高运算速度,考虑减少码字进位处理的次数,让 B 寄存器每次输出 16 位的数据,利用 C 寄存器的全部 32 位,有限精度处理如图 3 所示,图中 a 表示寄存器 A 中的非零数据位。对应的 0x4000 代表小数 0.75,初始值做相应的变动: $A = 0x4000$ 、 $C = 0$ 、移位计数器 $CT = 15$ 。

3.4 码字进位传递的处理

每当编码一个“1”时,有 $C = C + (Q_s \times A)$,这就可能在 C 中产生一个进位。如果这个进位传递到已经输出的压缩数据流中,需要改变以前输出的数据,这相当困难。为解决这个问题,用到位填充技术。从寄存器 C 中移出的 16 位,在送入数据流之前,保存在寄存器 B 中。当 CT 记数减到 0 的时候,存储在寄存器 B 中的数据输出到压缩数据流中,然后从寄存器 C 的高位移出 16 位到 B,如图 3。x 代

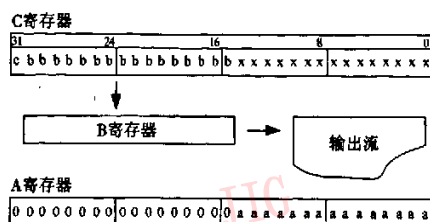


图 3 改进 MQ 编码中的寄存器和输出流

Fig. 3 Register and output of modified MQ-coder

表 C 中因为 Q_c 的加入而增加的小数位, b 是要从寄存器 C 转移到寄存器 B 的数据位, c 代表进位位。对寄存器 B 的操作遵循位填充机制:

(1) 如果 $B \neq 0\text{xfff}$, 并且进位位 $c = 0$, 则把寄存器 B 中的数据位转移到输出数据流, 把寄存器 C 中的第 15 到 30 位的数据转移到寄存器 B, 置 $CT = 16$ 使下一次的输出也是 16 位数据。

(2) 如果 $B = 0\text{xfff}$, 则把寄存器 B 中的数据位转移到输出数据流, 把寄存器 C 中包括进位的第 16 到 31 位的数据转移到寄存器 B, 置 $CT = 15$ 使下一次输出包含本次留下的第 15 位。

(3) 如果 $B \neq 0\text{xfff}$, 并且进位位 $c = 1$, 则寄存器 B 中的数据加 1, 置进位位 $c = 0$ 。检测寄存器 B, 如果 $B = 0\text{xfff}$, 遵循规则 1, 否则, 遵循规则 2。

按照上面的位填充机制, 在解码的时候, 解码器检测所有的数据为 0xfff 的字节后面的第 1 位, 如果这一位是 1, 则解码器知道这里产生了一个进位。

4 算法的 DSP 实现

4.1 C6000 系列 CPU 结构与流水线

C6000 系列 CPU 采用哈佛结构, 取指与执行指令可以并行运行。程序总线宽度为 256bits, 每一次取指操作都是取 8 条指令, 成为一个取指包。取指、指令分配和指令译码单元都具备每周周期读取并传递 8 条 32 位指令的能力。

TMS320C6000 的特殊结构使得其可以采用“流水线”(pipeline)的工作方式, 多个指令包(每包最多 8 条指令)交迭地不同部件内处理, 大大提高了微处理器的吞吐量。

4.2 MQ 编码的并行处理

位平面扫描完成后, 得到 32 位 int 型 2 维数组 $\text{probability}[N][2]$ 和 16 位 short 型数组 $\text{typebuf}[]$ 。

probability 存放 N 种概率模型的 0、1 符号的个数, typebuf 依次存放各符号 sym 和符号对应的概率模型 n , 元素定义为

$$\text{typebuf}[n] = (\text{sym} < 7) + n, \text{sym} \in \{0, 1\}, 0 \leq n \leq N-1$$

由 3.2 节可知, 根据 probability 可以准确得到各种概率模型中 0、1 的概率, 定点表示为 Q15 格式数, 求 C、A 时右移 15 位转换。由于集合中只有 $\{0, 1\}$ 两个符号, 只需各概率模型中符号 0 的概率 $p[n][0]$ 。

以线性汇编语言进行 MQ 编码的程序编写, 循环读取 $\text{typebuf}[]$ 中的符号 sym 和概率模型 n , 由 n 得到 $p[n][0]$, sym 和 $p[n][0]$ 输入进行 MQ 编码, 循环执行直到所有符号编码完毕, 循环结束后将 C 寄存器中残余的 bytes 输出。

C6000 中进行代码优化的一个重要的技术是软件流水, 通过编排循环指令, 使循环的多次迭代能够并行执行的技术。各个符号及其编码的计算不相关, 每次循环从 typebuf 中读取 4 个数组元素并行处理, 相应的寄存器 C、A、B、CT 和码流存储空间 pbuf 都设置 4 个, 编码循环结束后 4 组码流进行合并。C6000 有两个数据通路 A 和 B, 每个通路有 4 个完全相同的功能单元, 两个通路间的运算可以并行。因此, 1、2 两对符号及其概率的编码安排到 A 侧, 3、4 两对安排到 B 侧。取 $\text{typebuf}[]$ 中的元素时使用数据打包处理, 用 LDW(读入字)一次取两个 16 位数组元素。

软件流水的建立可以大大提高效率, 但是其建立是有条件限制的, 如一个循环内的代码长度不能太长(最长约为 200 条), 循环内不能有跳转等。而 MQ 编码对码字进位传递的处理用到位填充机制, 如 3.3 节所示, 需要相当的判断和跳转操作。而同时对 4 个符号进行编码时, 其中的跳转增加 4 倍, 这样会频繁打断流水线, 使软件流水无法建立。实现过程中采用条件操作代替跳转, 所有的 C6000 指令都是可以条件执行的, 在书写汇编程序时, 以方括号对条件操作进行描述, 方括号内是条件寄存器的名称, 如下面所示的执行包。

```
[B0] ADD .L1 A1, A2, A3
```

```
||[! B0] ADD .L2 B1, B2, B3
```

经过反复的设想和调试, 把位填充机制中的跳转指令全部改为条件操作, 并行效率大大提高。

按并行要求编写线性汇编程序以后, 设定优化选项, 由汇编优化器自动优化。-o3 选项表示可得

到的最大程度的优化,使用时尽量联合使用-pm 选项。-pm 是程序级优化,所有源文件都被编译到一个中间文件里,编译器从整个程序的角度来观察。-g选项使能符号调试和汇编源语句调试,会导致代码尺寸较大,执行变慢。调试完毕正常运行时,舍弃此选项。-op2、-mt 和-mh(n)选项对性能优化相对较小,在保证结果正确的情况下仍推荐尽量选用。

5 实验结果与结论

C6000 编译器提供一些默认的反馈,作为一种工具用于具体的解释关于循环的软件流水所有问题及最后得到的结果。编译器的反馈信息表明,MQ 编码循环的迭代间隔为 20,即完成 4 个符号的编码需要 20 个时钟周期(clock),则平均编码 1 个符号只需要 5 个 clock。

以 $512 \times 512 \times 8\text{bits}$ 的 Lena、Barbara 和 Goldhill 为标准图像,变换和上下文编码算法采用改进的双正交叠式变换^[3] + 算术编码,和文献[3]中算法类似,在工程样机硬件平台上进行测试。8:1 压缩,基于 C6000 的硬件 Timer,表 1 给出了改进前后 MQ 编码所需要的时钟周期(clock)。4:1、8:1 和 16:1 压缩,以峰值信噪比(PSNR)为评价标准,表 2 给出了改进前后的图像客观质量评价。

表 1 编码速度比较

Tab.1 Comparison of coding speed

单位:clock

	改进前	改进后	加速比
Lena	18 217 631	2 221 662	8.20
Barbara	19 753 681	2 429 727	8.13
Goldhill	17 976 220	2 211 097	8.13

表 2 PSNR 比较

Tab.2 Comparison of PSNR

单位:dB

		0.5bps	1bps	2bps
Lena	改进前	36.30	39.27	43.12
	改进后	36.42	39.52	43.71
Barbara	改进前	32.46	37.13	42.49
	改进后	32.51	37.36	43.13
Goldhill	改进前	32.80	35.87	40.55
	改进后	31.81	36.02	41.03

位平面算术编码的编码分离和 MQ 编码器改进方法,由于寄存器精度的提高,重建图像压缩质量稍有提高。由于编码分离后的并行处理和 MQ 编码器改进带来的良好计算性能,使 MQ 编码的效率有了很大幅度的提高。同时,编码分离方法和 MQ 编码器改进还可以无缝地连接到许多位平面算术编码中,具有良好的适应性。

本文算法已应用于某卫星图像数据压缩机中。该卫星星载多个线阵 CCD 相机,相机时钟频率为几十 MHz,每帧数据为 $n \times k$ 字节,总码速率近 1Gbps,要求 4:1 和 8:1 压缩。

参考文献(References)

- 1 Acetelma K R. Digital Image Processing[M]. Translated by Zhu Zhi-gang, Lin Xue-yan, Beijing: Publishing House of Electronics Industry, 2002: 377 ~ 380[Acetelma K R. 数字图像处理[M]. 朱志刚,林学颜译. 北京:电子工业出版社,2002:377 ~ 380.]
- 2 Glen G, Langdon Jr. An introduction to arithmetic coding[J]. IBM Journal of Research and Development, 1984, 28(2): 135 ~ 149.
- 3 Tu C, Tran T D. Context-based entropy coding of lock transform coefficients for image compression[J]. IEEE Transactions on Image Processing, 2002, 11(11): 1271 ~ 1283.
- 4 ISO/IEC FCD 15444-1, International Technology-JPEG2000 Image Coding System-Part1. Core Coding System[S].
- 5 Chen Bo, Cheng Li-zhi, Wang Hong-xia. LBT based low complexity image compression method[A]. In: 18th International Conference on Pattern Recognition[C], Hongkong, China, 2006: 941 ~ 944.