并行 MPS 算术编码的性能分析

王　前[①②]　　吕东强[③]　　葛宝珊[①]
[①](北京航空航天大学计算机学院　北京　100083)
[②](解放军 61081 部队　北京　100094)
[③](第二炮兵装备研究院四所　北京　100085)

摘　要：利用多维二进制码流的统计规律，基于 MPS 并行的算术编码不但避免了传统并行算术编码的复杂运算，且不会影响其基本概率估计规律。该文运用全概率定理和统计平均思想从理论上分析了并行度与加速比、编码效率之间的关系，指出并行度为 2 的 MPS 并行编码方案在编码速度和效率方面较其它并行度占有很大优势，并行度 3 和 4 的编码方案在编码效率方面基本持平，并加以试验证明。

关键词：算术编码；并行；大概率符号；状态转移
中图分类号：TN919.81　　**文献标识码**：A　　**文章编号**：1009-5896(2009)12-2907-05

Performance Analysis of Arithmetic Code on Parallelized MPS

Wang Qian[①②]　　Lü Dong-qiang[③]　　Ge Bao-shan[①]
[①](*School of Computer Science and Engineering*, *Beihang University*, *Beijing* 100083, *China*)
[②](61081 *army of PLA*, *Beijing* 100094, *China*)
[③](*The Fourth Institute of the Second Artillery Equipment Academe*, *Beijing* 100085, *China*)

Abstract: Arithmetic code on parallelized MPS(Most Probable Symbol) not only avoids complex operation of classical parallelized arithmetic code, but also does not inflect its basic probability estimation rule since utilizing statistic law of multidimensional binary coding. The relation between parallel degree, speedup ratio and coding efficiency is theoretically analyzed based on the theorem of complete probability and statistic average. It is pointed out the algorithm with 2 parallel degree is superior to others on the coding efficiency and speed, the algorithm of 3 parallel degree is equal to the one of 4 parallel degree on the coding efficiency. The result is verified by the experiment.

Key words: Arithmetic code; Parallelization; MPS(Most Probable Symbol); State transition

1　引言

算术编码算法能够灵活地适应数据的概率变化，压缩效率明显高于变长编码，但实现复杂度高，处理速度慢，难以满足大数据量处理时的实时性要求，因此在实际应用中受到很大的限制。针对算术编码的特点，流水线和并行处理是其中最主要的加速措施。

并行可分为等价并行和近似并行两类。如文献[1–4]提出的并行算法都属于等价并行，但文献[2]系统的吞吐率提高幅度较低，只有 24%，属于部分并行范畴；文献[3]设计出 5 种不同形式的算术编码并行结构，吞吐率提高幅度从 10%到 120%不等。文献[4]采用完全并行算法，吞吐率比单输入系统提高 100%，但系统复杂度较高，不利于硬件的快速实现。文献[5,6]的算法都属于近似并行，即改变标准算术编码的流程，所产生的压缩码流不能与标准算术解码兼容。如文献[5]运用线性近似法和并行概率模型实现并行编码，文献[6]运用 MPS 并行来提高并行度。近似并行吞吐率高，且实现难度小，但引起一个值得考虑的问题：即编码效率。本文针对 MPS 并行算术编码算法，分析其给编码过程带来的影响，利用数值解析法确定其编码效率的变化范围，为今后优化编码设计提供依据。

2　算法思想

二值算术编码的主要思想是根据输入信源符号的概率分布计算序列所对应的区间，然后进行区间截断，并把截断结果作为压缩码流的输出。考虑信源序列 $U=(u_1,u_2,\cdots u_L)$，二值化的特点使得编码序列分为大概率符号(MPS)和小概率符号(Less

2008-12-01 收到，2009-05-11 改回
国家 863 计划项目(2006AA701121)，教育部博士点基金和新世纪优秀人才支持计划资助课题

Probable Symbol，LPS)，即 $u_l \in A=\{0,1\}$ 的分布函数。考虑到并行化的需要，不妨设置扫描的窗口尺寸 $W_{\text{size}}=2$。加窗后的序列为 $u_l' \in B=\{00,10,01,11\}$ 其概率分布函数为 $P\{u_{i+1} \mid u_i\}$，若能够知道 u_l' 对应的区间分布，那么并行的问题就可以彻底解决。遗憾的是，算术编码的区间划分属于典型的递归反馈类型，前一符号的编码区间直接影响后一符号的区间归属，有如下递推关系：$F(U_{i+1})=F(U_i)+P(U_i)F(u_{i+1})$ 和 $P(U_{i+1})=P(U_i)P(u_{i+1})$，其中 $P(\bullet)$ 为对应编码区间，$F(\bullet)$ 为输出码流，因此直接找到 u_l' 的概率分布函数以及对应的区间划分是非常困难的事情。

在输入 MPS 的情况下，尽量减少编码的运算，这样减少复杂度，而把复杂的操作放置在针对 LPS 的编码过程中，是一种比较理想的并行策略选择。文献[6]采取了一种更为直接的策略。即在连续的大概率编码过程中，只对开头的 MPS 进行编码，舍掉后续的大概率编码，从而提高编码的并行度。从信息论的角度说，重点针对 LPS 进行编码，因为 LPS 符号包含更多的信息量，也有利保持信息量的完整，降低其冗余度。

在具体的实现过程中，对输入源引进加窗机制，其滑动方向受地址指针控制。设窗口尺寸为 2 bit 位宽，若窗口内两个是 MPS，则舍掉第 2 个 MPS，窗口内的其余比特流情况依照原先规则编码。这种舍掉 MPS 的机制其实质是 MPS 并行编码。

对任何一个有限长度的信源字母序列，如果编码得到的码字母序列不与其他任何信源字母对应的码字母序列相同，则称为唯一可译码。显然 MPS 并行编码法违反了译码唯一的原则，因为一旦在解码端遇到 MPS 码流，无法正确辨认是单 MPS 比特还是双 MPS 比特。为了能正确译码，需要在 LPS 后面加入 flag 标记。在解码端，遇到 MPS，默认为[MPS|MPS]，但不马上解码；若下一码是 MPS，则输出。若下一码是 LPS，则看 flag 的情况。可设 flag 为 1 时，码流为[MPS|MPS]；反之，为 MPS。

Flag 作为特殊的标志嵌入到压缩码流之中，为了区别与其它正常的数据码流，可分配一个单独的上下文符号与之匹配。由于它的加入，对编码器的压缩性能会产生一定的影响，影响的程度由其概率分布的特点决定。

关于窗口尺寸 W_{size} 的设定问题，是一个值得探讨的问题。总的原则是尽最大可能创造 MPS 并行的机会，但是在获得编码速度提高的同时，是否会用压缩性能的损失作为补偿？因此，W_{size} 实际上是调节性能和速度的参数因子。可以直观的分析一下，W_{size} 越大，包含的比特位数越多，窗口中首位比特是 MPS 的概率越低，因为相当多的 MPS 比特转移到窗口中的其他位置，也就是说 MPS 并行的概率就越低。所以 W_{size} 的选择对编码的速度和性能都会产生很重要的影响。本文将在第 3 节对这一问题作出完整的定量分析。

表 1 反映了输入序列在不同窗口尺度作用下的对应输出关系。符号“0”和“1”代表添加的 flag 标志，“0”表示未省略 MPS 符号，“1”的意义正好相反。“X”代表原先的输入码流符号。为明显表示对应关系，Win_2 窗口间用间隔符“-”连接。

表 1 不同端口系统下的输入输出关系表

输入	输出(Win_2)	输出(Win_4)	输出(Win_3)
00-00	0-0	000	00X
00-01	0-01	001	
00-10	0-10	0110	011X
00-11	0-11	0111	
01-00	01-0	0100	0100X
01-01	01-01	0101	
01-10	01-10	0110	0101X
01-11	01-11	0111	
10-00	10-0	1000	100X
10-01	10-01	1001	
10-10	10-10	1010	101X
10-11	10-11	1011	
11-00	11-0	1100	110X
11-01	11-01	1101	
11-10	11-10	1110	111X
11-11	11-11	1111	

3 压缩性能分析

并行 MPS 编码省略了大量的 MPS 编码，显然会提高编码效率，而在占少数的 LPS 后添加辅助标志，则会降低编码效率。两者相抵消后，究竟对编码效率产生太大的影响？接下来本文从理论推导和实验论证两方面说明该算法对压缩性能的影响。

3.1 状态转移与码流输出

算术编码通过建立 Markov 状态转移模型自适应地跟踪信源中的码字真实发生概率。概率估计由最大可能符号(MPS)和该符号的最小近似概率分布(Qe)表示。在算术编码中，由于是二进制编码，概率近似的值域为 46 个离散值，概率估计过程可近似为一阶离散 Markov 过程，概率更新就是从当前状态转移到下一个状态。

概率估计过程在任意 k 时刻状态记为 $X^{(k)}$，$X^{(k)}$ 为一随机变量，$X^{(k)}$ 的概率分布 $P^{(k)}=[x_1^{(k)},x_2^{(k)},\cdots x_{92}^{(k)}]$，根据概率分布的特性可知 $\sum_{n=1}^{92}x_n^{(k)}=1$，假设一阶转移矩阵为 $\boldsymbol{T}$，其中每个元素 t_{ij} 表示从状态 i 转移到状态 j 的概率，则下一时刻的概率分布可表示为 $P^{(k+1)}=P^{(k)}\times\boldsymbol{T}\sum_i x_i^{(k)}t_{ij}=x_j^{(k+1)}i,j\in[1,92]$；极限 $\lim_{n\to\infty}P^{(k)}\times T^n=\prod$。

概率更新的过程就是状态转移的过程，其间伴随着码流的输出。则系统在 k 状态的码率表达式为

$$R_k=q.\left(B_{\mathrm{lps},k}+\sum_{t=1}^{\infty}B_{\mathrm{mps},k}^{(t)}(1-q)^t\right)\tag{1}$$

其中 $B_{\mathrm{lps},k}$，$B_{\mathrm{mps},k}$ 分别为 LPS，MPS 归一化产生的比特数目，q 是 LPS 的发生概率。算术编码器为提高概率预测的准确性，把多值符号进行二进制表示，根据不同的统计特点，对其进行合理的归类，每一类码字用各自匹配的上下文模型记录其状态变化，则总的码率等于上下文概率空间内的码字之和，可表示为

$$R=\sum_{m=1}^{l}\sum_{k=1}^{h}R_{mk}\tag{2}$$

高阶模型通常比低阶模型有更好的压缩效果，但是随着阶数的增加，提升的空间逐步减少。显然，跟踪模型越完善，越能准确地预测码流变化规律。文献[7]对跟踪模型进行了更为深入的研究，依据不同的量化分层进行对比实验。更细致的分层使状态之间的跳转步长更小，会得到更精确的概率估计值，5 bit Qe index 的效率损失约 5%，6 bit Qe index 的效率损失在 2%。预测准确的同时带来实现复杂度的增加，不利于算法的快速实现。概率预测的准确度与收敛速度成反比关系，MPS 编码引起的状态迁移变化率小，更有利于精确预测，因此省略 MPS 符号而产生的并行编码只会略微影响算法的概率估计与状态转移。另一方面，从概率建模的角度来提高算术编码的压缩性能是很困难的事情，可从合理分类的角度来提高算术编码的压缩性能。位平面的扫描过程为数据的有效分类以及去除信息冗余提供了重要的保障。

要使 R_k 变小，直观的想法是增加 MPS 事件的发生概率，即提高式(1)中 $(1-q)^l$ 的数值，这样可减少归一化的次数，从而降低码流输出的长度。这种直观想法是与位平面扫描的实质相吻合的。位平面扫描的作用是把具有关联关系的码流更有效地结合在一起，因此就得到更长的 MPS 字符串，进一步奠定了 MPS 全并行编码器的实现基础。接下来本文将重点分析不同窗口尺度下的算法的编码效率。

3.2　2、4 输入系统的性能分析

为方便公式推导，不妨从复杂度最小的窗口尺寸 $W_{\mathrm{size}}=2$ 开始，考虑在概率分布函数为 $P\{u_{i+1}\,|\,u_i\}$ 中，$u_l\in A=\{0,1\}$ 的分布函数其中 $\eta_2=P\{\mathrm{MPS}\,|\,\mathrm{LPS}\}$；$\eta_1=P\{\mathrm{MPS}\,|\,\mathrm{MPS}\}$；$\eta_3=P\{\mathrm{LPS}\,|\,\mathrm{MPS}\}\approx\eta_2$；$\eta_4=P\{\mathrm{LPS}\,|\,\mathrm{LPS}\}$。标准算术的编码压缩率为 $\beta=1-\alpha$，设编码前的原始数据量为 M。

运用统计平均思想，每个 MPS 符号减少的码流量为

$$\frac{M\beta}{(M/2)(2\eta_1+\eta_2+\eta_3)}=\frac{\beta}{\eta_1+\eta_2}\tag{3}$$

则总的略掉 MPS 符号实际码流减少量为

$$\frac{M}{2}\eta_1\cdot\left[1-\frac{\beta}{\eta_1+\eta_2}\right]\tag{4}$$

显然在式(1)中，当 $l=1$ 时，码率达到上限 R_{Max}，也就是说当 MPS、LPS 两事件处于等概率状态时，得到的压缩码流长度最长，编码压缩率最低，约等于 1.034。flag 标志字是为正确解码专门设计的，其上下文模型没有经过细致的建模分类，因此不会有太高的编码效率，但总能低于上限 R_{Max}。设 flag 标志字的发生概率为 η_f，则输出码流增加量为 $M\cdot\eta_f\cdot1.034$。(其中 $\eta_f=\frac{\eta_2+\delta\eta_3}{2}\approx\frac{1+\delta}{2}\eta_2\approx(3/4)\eta_2$)

因此编码长度的减少量为

$$\Delta l\approx M\left[\frac{(\eta_1+\eta_2-\beta)\eta_1}{2(\eta_1+\eta_2)}-\frac{3}{4}\eta_2\right]=\frac{M}{4(\eta_1+\eta_2)}(2\eta_1^2-2\beta\eta_1-\eta_1\eta_2-3\eta_2^2)$$ 记为 $f_{\mathrm{Win2}}(\eta_2)$

当 $\Delta l=0$ 时，$\eta_2^{\mathrm{Win2}}=\frac{\sqrt{25\eta_1^2-24\beta\eta_1}-\eta_1}{6}$(舍去负值)。

当 W_{size} 增大后，为简化推导过程，不妨设 $W_{\mathrm{size}}=4$，可以理解成一个 Win4 窗口等位置地包含两个 Win2 窗口，由于{MPS | MPS}在 Win4 中的后半部分不参与并行编码，因此并行发生率 η_1 也随之减小，按统计规律，大概为原来的 1/2；η_f 的情况分为两类：{MPS | LPS}在 Win4 中的后半部分不参与 flag 编码，而{MPS | LPS}在 Win4 中的前部分参与 flag 编码的概率则不变，因此在 Win4 中 flag 编码略小于 Win2 中的 2/3。

依此类推，$\Delta l\approx\frac{M}{4(\eta_1+\eta_2)}(\eta_1^2-\beta\eta_1-\eta_1\eta_2-2\eta_2^2)$，记为 $f_{\mathrm{Win4}}(\eta_2)$。

$$\eta_2^{\mathrm{Win4}}=\frac{\sqrt{9\eta_1^2-8\beta\eta_1}-\eta_1}{4}\text{(舍去负值)}$$

可以证明 $\eta_2^{\text{Win2}} > \eta_2^{\text{Win4}}$，其物理意义为：在同等输入情况下，Win2 算法较 Win4 算法更容易获得编码正增益($\Delta l > 0$)。因为 $\max\left(f_{\text{Win2}}(\eta_2)\right) = f_{\text{Win2}}(0) > \max\left(f_{\text{Win4}}(\eta_2)\right) = f_{\text{Win4}}(0)$ ， $f_{\text{Win2}}(x)$ 和 $f_{\text{Win4}}(x)$ 在 $x \geq 0$ 时均为单调函数。当 $f_{\text{Win2}}(x) \geq f_{\text{Win4}}(x)$ 时，可推得 $\eta_1^2 - \beta\eta_1 \geq \eta_2^2$ 普遍成立。因此曲线 $f_{\text{Win2}}(\eta_2)$ 始终在曲线 $f_{\text{Win4}}(\eta_2)$ 的上方。也就是说 Win2 算法的编码效率高于 Win4 算法的编码效率。

3.3 3 输入系统的性能分析

当 W_{size} =3 时，与 Win_2 相比，不是等位置地进行窗口划分，直接推导 Δl_3 有些繁琐，但由于 $\text{Win}_3 \in [\text{Win}_2, \text{Win}_4]$ 的区间，是否可以推导出其编码压缩性能 $\Delta l_3 \in [\Delta l_2, \Delta l_4]$ 的区间？本文将进行如下分析。

设 Win_2 或 Win_4 可采用尺度为 4 的滑动窗口，Win_3 的滑动窗口尺度为 3，3 和 4 的最小公倍数为 12，即窗口划分为 12 时，不破坏 Win_2 和 Win_3 的排列关系， η_f 的变化情况可等价为 {MPS|LPS} 或 {LPS|MPS} 在窗口内随机出现的条件下，运用全概率定理，添加 flag 标志的概率，可用下式表示：

$$\frac{\sum_{k=1}^{12} P\left(\text{flag}_{\eta_3} + \text{flag}_{\eta_2} \mid \text{win}_3\right)_k}{\sum_{k=1}^{12} P\left(\text{flag}_{\eta_3} + \text{flag}_{\eta_2} \mid \text{win}_2\right)_k} = \frac{4*(1+\phi)+8\phi}{3*(1+2\phi)+3*3\phi} = \frac{4+12\phi}{3+15\phi}(\phi \to 1) \quad (5)$$

其导数在取值范围内恒小于 0，可估算出上式的数值略大于 8/9。

依据统计规律，并行发生率 η_1 为原 Win_2 的 2/3，因此

$$\Delta l \approx \frac{M}{4(\eta_1+\eta_2)}\left(\frac{4}{3}\eta_1^2 - \frac{4}{3}\beta\eta_1 - \frac{67}{51}\eta_1\eta_2 - \frac{45}{17}\eta_2^2\right),$$

记为 $f_{\text{Win3}}(\eta_2)$。

$$\eta_2^{\text{Win3}} = \frac{\sqrt{4579\eta_1^2 - 4080\beta\eta_1} - 22.3\eta_1}{90} \text{(舍去负值)}$$

直接比较 η_2^{Win3} 和 η_2^{Win4} 之间的关系较为复杂，可借助图解法来解决此类问题。图 1 反映在一定的 MM 串概率条件下，不同窗口尺寸下零点曲线的走向关系。所谓“零点”，其物理意义是 MPS 并行算术算法编码效率与标准算术编码效率之差值。判断依据为：零点曲线值越大，编码效率越高。标准算法的实际值位于 win2 曲线和 win3，win4 曲线之间，表明 Win2 算法能够获得比标准算术算法更高的编码效率，而 Win3 和 win4 算法能够只能获得比标准算术算法更低的编码效率，且 win4 效率略高于 Win3，最终得到编码效率的不等式关系为：$E_{\text{win2}} \gg E_{\text{win4}} \geq E_{\text{win3}}$。同时，随着 MM 串比例的增加，即压缩倍率的提高，win 曲线也表现出缓慢的增长关系，说明此时的编码效率也略有提高。

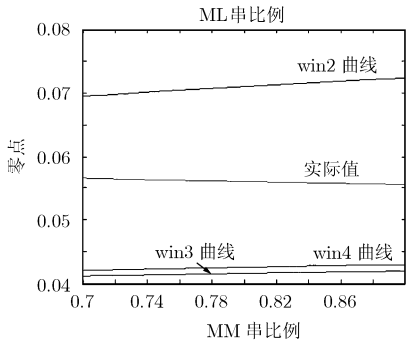

图 1 零点曲线走向图

4 实验验证

给定不同压缩倍率的标准信源[8]，经过不同窗口参数的算法处理，得到的结果如表 2。表中编码效率系数 α 的定义为 $\frac{L_{W_{\text{size}}} - L_0}{L_0}$，其中 $L_{W_{\text{size}}}$ 为文中算法产生的压缩码流长度，L_0 为标准算术编码产生的压缩码流长度。因此 α 越小，编码效率越高。

表 2 不同窗口尺度下的编码效率表

序列	W_{size}	64 倍	32 倍	4 倍
City	2	−0.0423	−0.0366	−0.0297
	3	0.0144	0.0191	0.0253
	4	0.0144	0.0156	0.0196
factory	2	−0.0358	−0.0338	−0.0319
	3	0.0162	0.0201	0.0237
	4	0.0142	0.0184	0.0191
couple	2	−0.0291	−0.0291	−0.0285
	3	0.0234	0.0249	0.0264
	4	0.0208	0.0219	0.0216
cman	2	−0.0391	−0.0337	−0.0251
	3	0.0158	0.0176	0.0255
	4	0.0053	0.0118	0.0197
lena	2	−0.0303	−0.0293	−0.0278
	3	0.0188	0.0199	0.0237
	4	0.0178	0.0188	0.0219
Barbara	2	−0.0326	−0.0314	−0.0265
	3	0.0178	0.0181	0.0256
	4	0.0147	0.0173	0.0228

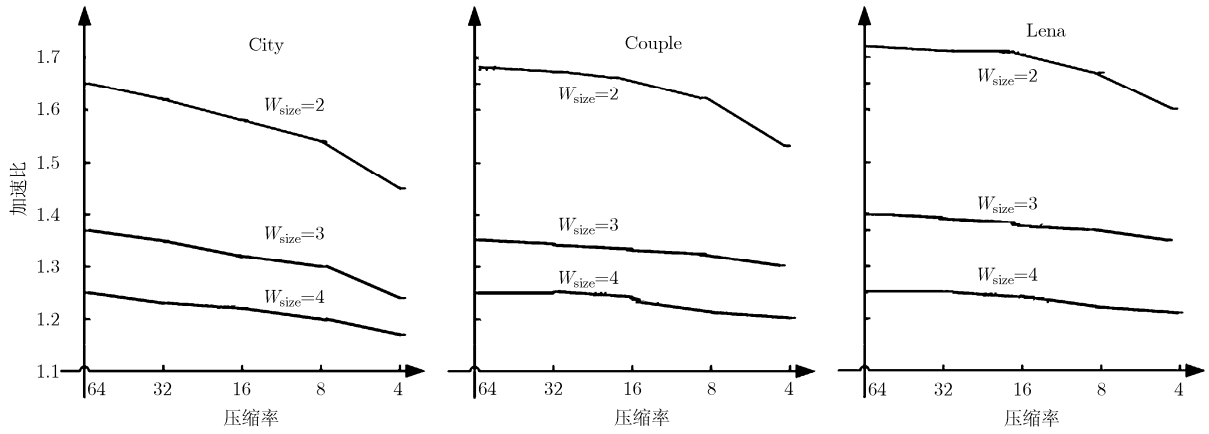

图 2 MPS 并行算术算法的并行度表现图

由于压缩效率与倍率的变化不是很明显，为便于说明效果，倍率之间的跨度较大。各类图像在 $W_{\text{size}}=2$ 时的压缩效率比均为负值，即此时编码效率略高于标准算术编码，而在 $W_{\text{size}}>2$ 时的编码效率都低于标准算术编码，$W_{\text{size}}=3$ 时的编码效率略低于 $W_{\text{size}}=4$ 时的编码，与先前的理论推导基本一致。由表 2 还可得到这样的规律：压缩效率与倍率在通常情况下成正比，即倍率越高，压缩效率越好。但这种变化趋势的变化率很平缓，一般在 1%到 2% 之间。因此，该算法在低倍压缩中仍适用。

系统的吞吐率与窗口尺寸成反比，这是由该算法的基本性质决定的。但是并行的衰减幅度并不是均匀的，具体来说，当 W_{size} 由 2 到 3 时，衰减幅度最大，而 W_{size} 由 3 到 4 时，幅度次小，照此推理，幅度逐渐减小。图像的复杂度对并行会产生一定的影响，图像越简单，对应的加速比也较大。这主要是简单图像中 MPS 符号的比例较多造成的。图像的压缩倍率与加速比成正比关系。图 2 反映了不同复杂度图像加速比随压缩率变化的曲线。加速比 β 的具体定义为：$\beta=\dfrac{T_0}{T_{W_{\text{size}}}-T_0}$，其中 $T_{W_{\text{size}}}$ 为文中算法需要的时长，T_0 为标准算术编码需要的时长。β 越大，系统的吞吐率越高。由图可知，窗口尺度越小，倍率与加速比之间的变化曲率越明显。因为此时加速比的绝对值较大，对倍率的变化也较为明感。

5　结论

并行算术编码由于吞吐率高，正逐步得到广泛应用。本文针对 MPS 并行的算术编码方案，指出其不会改变算术编码的基本概率估计规律，运用全概率定理和统计平均思想从理论上分析了并行度与加速比、编码效率之间的关系，并在试验中得到证明。在位平面扫描提供较好统计分类的前提下，并行度为 2 的 MPS 并行编码在编码速度和效率方面较其它并行度占有很大优势，具有良好的应用前景。

参 考 文 献

[1] Zhang Yi-zhen, Xu Chao, and Wang Wen-tao. Performance analysis and architecture design for parallel EBCOT encoder of JPEG2000[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2007, 17(10): 1336–1347.

[2] Li Y J, Elgamel M, and Bayoumi M. A partial parallel algorithm and architecture for arithmetic encoder in JPEG2000[C]. IEEE International Symposium on Circuits and Systems, Kobe Japan, 2005, 5: 5198–5201.

[3] Pastuszak G. A high-performance architecture of double-mode binary coder for H. 264.AVC[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2008, 18(7): 949–960.

[4] Gupta A K, Nooshabadi S, and Taubman D. Realizing low-cost high-throughput general-purpose block encoder for JPEG2000[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2006, 16(7): 843–858.

[5] Lin J H and Keshab K P. Parallelization of context-based adaptive binary arithmetic coders[J]. *IEEE Transactions on Signal Processing*, 2006, 54(10): 3702–3711.

[6] Andra K. Wavelet and entropy coding accelerator for JPEG 2000. [Ph.D. dissertation]. USA: Arizona State University, 2001.

[7] Pennebaker W B and Mitchell J L. Probability estimation for the Q-Coder[J]. *IBM Journal of Research and Development*, 1988, 32(6): 737–752.

[8] 焦润海. 图像压缩中的高效预测编码及其优化实现技术. [博士论文], 北京: 北京航空航天大学计算机学院, 2007.
Jiao R H. High efficient prediction coding and its optimization in image compression. [Ph.D.dissertation], Beijing: School of Computer Science and Engineering, Beihang University, 2007.

王　前：　男，1978 年生，工程师，研究方向为图像压缩技术及其硬件实现.

吕东强：　男，1978 年生，工程师，研究方向为图像压缩及融合技术.

葛宝珊：　男，1967 年生，副教授，研究方向为数字视频处理、计算机网络和嵌入式系统.