

# CABAC 熵编码机制下有效的变换域码率估计技术

孙书为, 陈书明, 刘祥远

(国防科技大学计算机学院, 湖南长沙 410073)

**摘 要:** CABAC 是 H. 264/AVC 视频压缩标准中采用的一种熵编码机制, 结合 RDO 模式选择技术, 可以降低 20% 的码率. 在 RDO 模式选择过程中, 需要对编码块的每一种候选模式进行熵编码以获得编码位数, 这在很大程度上增加了视频编码的计算复杂度. 为了避免复杂的 Lagrange 耗费计算, 本文第一次提出了一种 CABAC 熵编码机制下的变换域码率估计方法, 基于量化变换系数和运动向量差对熵编码位数进行估计. 在此基础上, 本文将码率估计技术应用在 RDO 模式选择中, 提出了基于码率估计的快速模式选择算法, 减少了模式选择的计算时间. 模拟结果显示, 本文提出的 CABAC 熵编码机制下的变换域码率估计技术在对编码性能影响不大的情况下, 减少了模式选择中 51% 的率失真耗费计算时间, 采用全搜索运动估计算法时, 节省 33% 的总编码时间.

**关键词:** CABAC; 熵编码; 码率估计; RDO 模式选择; Lagrange 耗费

**中图分类号:** TN919. 8      **文献标识码:** A      **文章编号:** 0372 2112 (2008) 08 1512 07

## Efficient Transform Domain Bit-Rate Estimation Technique for CABAC

SUN Shuwei, CHEN Shuming, LIU Xiangyuang

(School of Computer Science, National University of Defense Technology, Changsha, Hunan 410073, China)

**Abstract:** CABAC is the entropy coding method adopted in H. 264/AVC main profile, and the combination of RDO and CABAC can represent an improvement in bit rate of around 20%. During the RDO mode decision, entropy coding is performed for every candidate encoding mode, which increases the computation complexity of video encoding drastically. To avoid the expensive computation of Lagrange costs, this paper proposes a transform domain bit rate estimation method for CABAC for the first time, in which the bit rate is estimated based on the quantized transform coefficients and the motion vector differences. After that, a fast mode decision algorithm based on the bit rate estimation technique is presented, in which the bit rate estimation is performed instead of actual entropy coding in mode decision, thus reducing the computing time of H. 264 encoder. Simulation results show that the proposed transform domain bit rate estimation method for CABAC achieves about 51% computation time reduction of Lagrange cost for mode decision and saves about 33% total encoding time with FS(full search) motion estimation algorithm, while the losses of PSNR and bit rate are negligible.

**Key words:** CABAC; entropy coding; bit rate estimation; RDO mode decision; Lagrange cost

### 1 引言

H. 264/AVC 是 ITU-T 和 MPEG 联合制定的一个国际性视频编码标准, 它以传统的混合编码框架为基础, 提供了更多可供选择的编码模式和更加高效的熵编码机制—CABAC(上下文自适应二进制算术编码 Context-Based Adaptive Binary Arithmetic Coding)<sup>[1]</sup>, 从而获得了更高的视频压缩率.

为了在多种编码模式中选择出最好的一种, 人们提出了 RDO(率失真优化) 模式选择方法<sup>[2]</sup>, 以穷举方式计算编码块在每一种模式下的率失真性能, 选择具有最小

Lagrange 耗费的编码模式作为最佳模式:

$$J = D + \lambda \times R \quad (1)$$

此处,  $\lambda$  为 Lagrange 算子, 和量化参数 QP 直接相关;  $D$  是重构失真;  $R$  是熵编码后的码流位数.

文献[3]研究了 RDO 和 CABAC 的性能及其计算复杂度, 发现 RDO 和 CABAC 的结合可以降低 20% 的码率, 但是其代价是计算复杂度的明显增加, 算术编码和率失真模式选择的计算时间大约占总编码时间的 50%.

为了减少视频编码算法的计算复杂度, 人们在快速模式选择算法方面做了大量的研究工作<sup>[4~10]</sup>. 通过利

用图像的纹理特征, 宏块的平滑程度以及相邻宏块之间的相关性等信息, 预先排除一些不太可能的编码模式, 通过减少模式选择过程中候选模式数目的方式减少 RDO 计算时间. 但是排除一些编码模式后的 RDO 计算时间仍然很多, 而且对视频编码的率失真性能产生了较大的影响<sup>[3]</sup>.

与上述方法不同, 本文从另外一个角度研究降低 RDO 计算复杂度的方法: 以码率估计代替 RDO 模式选择过程中的熵编码, 从而加快 RDO 模式选择过程. 为了在不进行实际熵编码的情况下获得码率估计值, 人们基于不同的概率分布假设(广义高斯分布或者拉普拉斯分布)提出了不同的码率模型<sup>[11~15]</sup>. 模拟结果发现, 对于较高层次的统计(比如帧或者视频序列的层次), 这些假设和实际情况吻合较好, 但是当统计层次降低至宏块层甚至编码块层, 编码符号并不遵循如此良好的概率分布特点. 因此, 上述模型均只应用于宏观层次的码率控制和位分配机制中.

文献[16, 17]研究了 CAVLC(上下文自适应变长编码 Context-based Adaptive Variable Length Coding) 熵编码机制下的变换域码率估计技术, 以量化变换系数和运动向量差为基础对熵编码位数进行估计. 他们的方法大大加快了 RDO 模式选择过程, 但是其方法并不适用于 CABAC.

本文研究了 CABAC 熵编码机制下的变换域码率估计技术, 并将其应用于快速模式选择算法中. 模拟结果显示, 本文提出的码率估计方法在对编码性能影响不大的情况下, 可以减少 51% 的率失真耗计算时间, 在采用全搜索运动估计算法时, 可以节省 33% 的总编码时间.

## 2 H. 264 编码器的 CABAC 熵编码机制

算术编码(AC)可以看作是基于符号概率而对一段区间进行的迭代式分割, CABAC 是一种简化的算术编码方法<sup>[8]</sup>.

假设  $LPS$  (最小可能符号)的概率估计为  $P_{LPS} \in (0, 0.5]$ , 区间由其较小端的边界值  $low$  和宽度  $range$  表示. 给定区间进一步划分为 2 个子区间, 分别对应于  $LPS$  和  $MPS$ (最大可能符号), 如果待编码符号为  $LPS$ , 新的当前区间更新为

$$range_{new} = range \times P_{LPS} \quad (2)$$

$$low_{new} = low + range - range \times P_{LPS} \quad (3)$$

如果待编码符号为  $MPS$ , 新的当前区间更新为

$$range_{new} = range - range \times P_{LPS} \quad (4)$$

$$low_{new} = low \quad (5)$$

为了保证计算精度, 每次进行算术编码后对  $range$

和  $low$  进行归一化操作, 使得  $range$  值始终处于区间  $[0 \times 100, 0 \times 1 \text{ ff}]$  中. 为了达到该目的,  $range$  的值左移必需的位数,  $low$  的值左移相同的位数, 移出  $low$  的位构成了算术编码的结果.

在 CABAC 中, 大多数的符号都在一个上下文环境中进行编码, 那些具有相同统计特性的符号具有相同的上下文, 而不同统计特性的符号上下文也不同. 每个上下文环境中维护一种概率模型, 包含一对值: 6 位的概率状态索引值和 1 位的  $MPS$  值. 在 JM10.2 中, 共采用了 852 种不同的上下文环境<sup>[19]</sup>.

## 3 CABAC 熵编码机制下的码率估计技术

### 3.1 码率估计模型

根据 CABAC 熵编码机制, 假设符号  $LPS$  出现的概率为  $P_{LPS}$ , 并且编码  $x$  个  $LPS$  可以生成 1 位码流:

$$2^9 \times P_{LPS}^x = 2^8 \quad (6)$$

那么编码 1 位  $LPS$  生成的码流位数  $B_{LPS}$  为:

$$B_{LPS} = \ln P_{LPS} / \ln 0.5 \quad (7)$$

基于同样的道理, 编码 1 位  $MPS$  可以生成的码流位数  $B_{MPS}$  为:

$$B_{MPS} = \ln(1 - P_{LPS}) / \ln 0.5 \quad (8)$$

式(7)和(8)说明,  $B_{LPS}$  和  $B_{MPS}$  是  $P_{LPS}$  的对数函数. 在计算机上实现对数函数是非常复杂的工作, 本文采用简化的方法获得  $B_{LPS}$  和  $B_{MPS}$  值: 由于  $LPS$  的概率  $P_{LPS} (0, 0.5]$ , 可以将该区间划分为 10 个子区间, 对每个子区间取其中间值作为该子区间的典型概率, 并以此典型概率获得  $B_{LPS}$  和  $B_{MPS}$  的值应用于整个子区间, 表 1 给出了一种子区间划分的例子.

表 1 概率区间的划分及其  $B_{LPS}$  和  $B_{MPS}$  值

$P_{LPS}$	$B_{LPS}$	$B_{MPS}$	$P_{LPS}$	$B_{LPS}$	$B_{MPS}$
0.05	4.322	0.074	0.30	1.737	0.515
0.10	3.322	0.152	0.35	1.515	0.621
0.15	2.737	0.234	0.40	1.322	0.737
0.20	2.322	0.322	0.45	1.152	0.862
0.25	2.000	0.415	0.50	1.000	1.000

基于上面的分析, 我们给出 CABAC 熵编码机制下的码率估计方法: 假设编码某一语法单元时, 需要编码  $LPS$  的数目为  $N_{LPS}$ , 需要编码  $MPS$  的数目为  $N_{MPS}$ , 那么编码该语法单元生成的码流位数可以估计为

$$B_{ESTI} = B_{LPS} \times N_{LPS} + B_{MPS} \times N_{MPS} \quad (9)$$

文献[16]指出, 在第  $\xi h$  种模式下编码块的总位数  $B_T(QP, \xi)$  包括三部分: 量化变换系数的熵编码位数  $B_C(QP, \xi)$ , 运动向量的熵编码位数  $B_M(QP, \xi)$  和头信息

的熵编码位数  $B_H(QP, \xi)$ , 其中  $B_H(QP, \xi)$  包括编码模式, 参考帧和编码块类型 ( $CBP$ ) 等信息的编码位数. 总编码位数表示如下:

$$B_T(QP, \xi) = B_C(QP, \xi) + B_M(QP, \xi) + B_H(QP, \xi) \quad (10)$$

下面分别介绍每个部分进行码率估计的方法.

### 3.2 量化变换系数的码率估计方法

文献[18]对量化变换系数的 CABAC 熵编码进行了详细的描述(本文不再赘述其具体过程), 需要处理 6 种类型的符号, 分别为 *coded\_block\_flag*, *significant\_coeff\_flag*, *last\_significant\_coeff\_flag*, *coeff\_greater\_one\_flag*, *coeff\_abs\_level\_minus2* 和 *coeff\_sign\_flag*. 对于每种符号, H.264 均设置了多种上下文进行编码(比如 JM10.2 为 *significant\_coeff\_flag* 设置了 150 种上下文, 为 *coeff\_abs\_level\_minus2* 设置了 50 种上下文). 但是编码一种符号时并不会用到其对应的全部上下文, 而且一种符号在不同上下文环境中 0 和 1 的概率虽然不尽相同, 但是其差别并不是很大, 统计结果显示, 同一种符号在不同上下文环境中的概率趋于相同. 图 1 和图 2 分别给出了 CIF 格式的 Foreman 序列在第 1 帧(I 帧)和 10 帧(P 帧)编码第 198 个宏块时, *significant\_coeff\_flag* 和 *coeff\_abs\_level\_minus2* 在不同上下文中符号 0 的概率分布(仅统计了使用到的上下文). 其他符号也具有类似的统计结果.

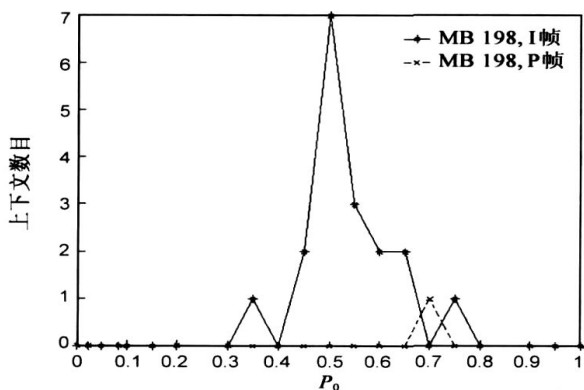


图 1 *significant\_coeff\_flag* 不同上下文中  $P_0$  分布因此, 本文在码率估计算法中以上述 6 种符号为单

位分别统计其已编码 0 和 1 的数目, 并据此得出每种符号的  $P_{LPS}$ , 然后以编码块为单位统计每种符号需要编码 0 和 1 的数目, 并据此估计量化变换系数部分的编码位数.

在进行码率估计前, 通过对量化变换系数块进行扫描, 可以得到如下信息:  $Num_{nonzero\_coeff}$ , 量化变换系数块中非 0 系数的数目;  $Num_{one\_coeff}$ , 绝对值等于 1 的系数数目;  $Pos_{last\_coeff}$ , 最后一个非 0 系数的位置;  $Sum_{absminus2\_coeff}$ , 所有绝对值大于 1 的系数绝对值减 2 的

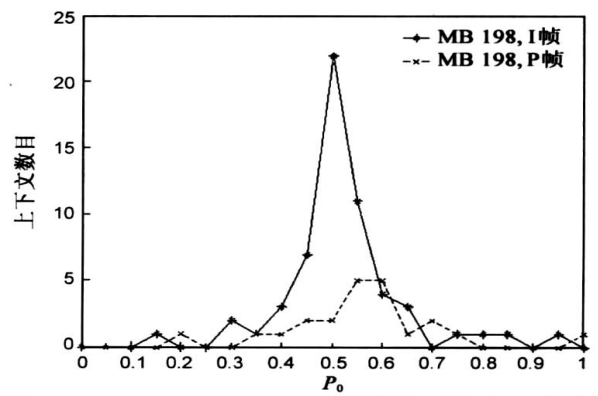


图 2 *coeff\_abs\_level\_minus2* 不同上下文中  $P_0$  分布和(如果系数绝对值大于 15, 统计绝对值时按照 13 累计该系数). 基于这些扫描信息, 可以对每种符号进行码率估计.

对 *coded\_block\_flag*, 由于每个块只有 1 位信息, 大部分情况下熵编码后的码流位数为 0, 因此以 0 作为该符号的估计值.

对 *significant\_coeff\_flag*, 每个块中会出现  $Num_{nonzero\_coeff}$  个 1,  $Pos_{last\_coeff} - Num_{nonzero\_coeff}$  个 0(对帧内  $16 \times 16$  预测模式的 AC 系数而言, 0 的数目在此基础上再减去 1).

对 *last\_significant\_coeff\_flag*, 每个块中会出现  $Num_{nonzero\_coeff} - 1$  个 0, 1 个 1.

对 *coeff\_greater\_one\_flag*, 每个块中会出现  $Num_{one\_coeff}$  个 0,  $Num_{nonzero\_coeff} - Num_{one\_coeff}$  个 1.

对 *coeff\_abs\_level\_minus2*, 每个块中会出现  $Num_{nonzero\_coeff} - Num_{one\_coeff}$  个 0,  $Sum_{absminus2\_coeff}$  个 1. 注意, 此时只统计了 *coeff\_abs\_level\_minus2* 进行指数哥伦布二进制化的前缀部分(对大量视频序列进行模拟统计发现大多数量化变换系数的绝对值小于 15, 也就是说对应的 *coeff\_abs\_level\_minus2* 在指数哥伦布二进制化后只存在前缀部分, 即使有很小一部分系数在指数哥伦布二进制化后存在后缀部分, 其位数和前缀部分相比也是非常小的). 实验结果显示, 本文对 *coeff\_abs\_level\_minus2* 的处理方式是可以接受的.

对 *coeff\_sign\_flag*, 由于统计中量化变换系数为正和负值的概率是相同的, H.264 编码器对 *coeff\_sign\_flag* 进行等概率二进制编码, 其熵编码位数和非 0 系数的数目是相同的, 即  $Num_{nonzero\_coeff}$ .

通过上述分析可见, 对 *coded\_block\_flag* 和 *coeff\_sign\_flag*, 分别采用 0 和  $Num_{nonzero\_coeff}$  作为其码率估计值, 对其他 4 种类型的符号, 分别得到了相应的  $N_{LPS}$  和  $N_{MPS}$  的值. 在码率估计过程中, 通过累加所有编码块中每种编码符号的  $N_{LPS}$  和  $N_{MPS}$  值可以获得该符号已编码

LPS 和 MPS 的总数目,也就获得了 LPS 的概率  $P_{LPS}$ , 通过表 1 就可以获得编码符号的  $B_{LPS}$  和  $B_{MPS}$ . 这样通过式 (9) 可以估计他们的码率值. 6 种符号的估计值都得到后, 将它们累加起来就得到了当前量化变换系数块的码率估计值.

### 3.3 运动向量的码率估计方法

文献[18]对运动向量的 CABAC 熵编码进行了详细的描述(本文不再赘述其具体过程), 需要处理 3 种类型的符号, 分别为 *significant\_mvd\_flag*, *mvd\_abs\_level\_minus1* 和 *mvd\_sign\_flag*. 基于和上一节相同的原因, 本文在码率估计算法中以上述 3 种符号为单位分别统计其已编码 0 和 1 的数目, 并据此得出每种符号的  $P_{LPS}$ . 然后以编码块为单位统计每种符号需要编码 0 和 1 的数目, 并据此估计运动向量部分的编码位数.

在进行码率估计前, 通过对当前编码模式的每一对运动向量值进行扫描, 可以得到如下信息:  $Num_{zero\_mvd}$ , 等于 0 的运动向量分量数目;  $Num_{nonzero\_mvd}$ , 非 0 运动向量分量数目;  $Sum_{absminus1\_mvd}$ , 所有非 0 运动向量分量的绝对值减 1 的和(如果运动向量分量绝对值大于 14, 统计绝对值时按照 13 累积该运动向量分量). 基于这些扫描信息, 可以对每种符号进行码率估计.

对 *significant\_mvd\_flag*, 每个帧间预测编码块会出现  $Num_{zero\_mvd}$  个 0,  $Num_{nonzero\_mvd}$  个 1.

对 *mvd\_abs\_level\_minus1*, 每个帧间预测编码块会出现  $Num_{nonzero\_mvd}$  个 0,  $Sum_{absminus1\_mvd}$  个 1. 注意, 基于前面处理 *coeff\_abs\_level\_minus2* 相同的原因, 此时只统计了 *mvd\_abs\_level\_minus1* 进行指数哥伦布二进制化后的前缀部分.

对 *mvd\_sign\_flag*, 由于统计上运动向量为正值和负值的概率是相同的, H. 264 中对 *mvd\_sign\_flag* 进行等概率二进制编码, 其编码位数和非 0 运动向量分量的数目是相同的, 即  $Num_{nonzero\_mvd}$ .

通过上述分析可见, 对 *mvd\_sign\_flag*, 本文采用  $Num_{nonzero\_mvd}$  作为其估计值, 对其他 2 种类型的符号, 分别得到了相应的  $N_{LPS}$  和  $N_{MPS}$  的值, 通过上一节相同的方法可以获得其对应的  $B_{LPS}$  和  $B_{MPS}$  值, 因此可以利用式 (9) 分别计算其相应的码率估计值. 最后将这些估计值累加起来就得到了当前帧间预测编码块运动向量的码率估计值.

### 3.4 其他部分的码率估计方法

理论上讲, 应该对式 (10) 中的三个编码部分都进行式 (9) 所描述的码率估计过程. 但是经过实验发现,  $B_C(QP, \xi)$  和  $B_M(QP, \xi)$  在总编码位数中的比重非常大, 而  $B_H(QP, \xi)$  的比重非常小. 由于  $B_H(QP, \xi)$  部分语法单元具有复杂多样的特点, 如果采用式 (9) 描述的方法

估计其码率, 那么新增加的计算负载足以抵消由码率估计替代熵编码所节省的计算负载. 因此本文采用较为简单的方法估计它们的码率, 比如对编码模式, 直接采用编码模式的索引值作为其码率估计值, 实验发现编码效果是可以接受的.

## 4 基于码率估计技术的快速模式选择算法

H. 264 编码算法是以宏块(MB)为基本单位进行处理的, 其基本流程如图 3 所示, RDO 模式选择存在于其中的 3 个步骤中:  $P8 \times 8$  帧间预测模式选择, I4MB 帧内预测模式选择和 MB 预测模式选择.

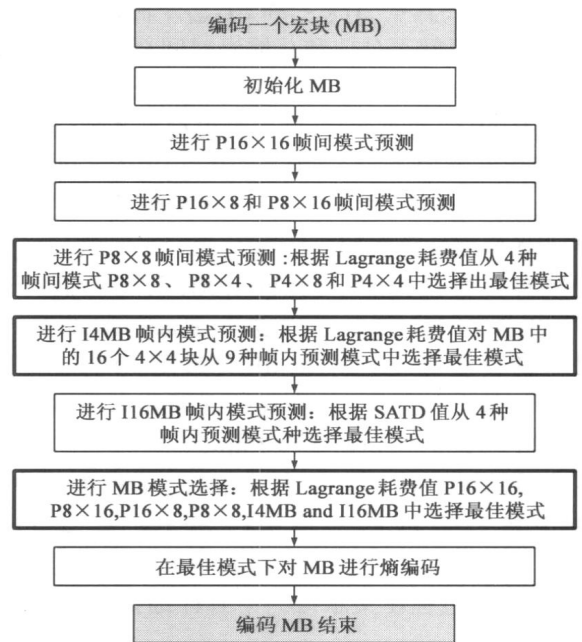


图 3 宏块编码基本流程

其中, 在  $P8 \times 8$  帧间预测模式选择过程中, Lagrange 耗费中的码率部分包括块头信息的码率, 运动向量的码率和亮度量化变换系数的码率三个部分; 在 I4MB 帧内预测模式选择过程中, Lagrange 耗费中的码率部分包括头信息的码率和亮度量化变换系数的码率两个部分; 在 MB 预测模式选择过程中, Lagrange 耗费中的码率部分除了头信息的码率, 运动向量的码率和亮度量化变换系数码率之外, 还包括色度量变换系数的码率.

下面给出基于码率估计的快速模式选择算法: 采用上一部分描述的变换域码率估计方法替换 RDO 模式选择中的 CABAC 熵编码, 根据码率估计值计算相应的 Lagrange 耗费, 并以此为标准进行模式选择; 宏块模式选择完成后, 在最佳模式下对宏块进行 CABAC 熵编码以生成码流; 在宏块编码过程中, 根据最佳编码模式累计各种语法单元已编码的 LPS 和 MPS 数目, 并根据该数值计算相应语法单元的  $P_{LPS}$  值.

## 5 实验结果与分析

为了验证本文提出的码率估计方法及其快速模式选择算法的有效性,本文基于参考软件 JM10.2<sup>[19]</sup>进行了实验,并和其他的快速模式选择算法在编码时间,输出码率和峰值信噪比 PSNR(peak signal to noise ratio)三个方面进行了对比。

实验中统计的 PSNR 值是亮度和色度的平均 PSNR 值,其计算过程见文献[4]。实验结果以编码时间差异  $\Delta T_{RD\text{Cost}}$  和  $\Delta T_{Total}$ , 码率差异  $\Delta Bit\text{-rate}$  和峰值信噪比差异  $\Delta PSNR$  的形式给出,所有的差异值均是相对于 JM10.2 的结果获得的。其中  $\Delta T_{RD\text{Cost}}$  和  $\Delta T_{Total}$  分别表示本文算法节省的 RDO 时间百分比和总编码时间百分比,  $\Delta Bit\text{-rate}$  表示本文算法获得的码率改变量的百分比,  $\Delta PSNR$  表示平均 PSNR 值的改变量,其计算过程见文献[16]。

实验环境如下: Intel Pentium4 2.0GHz CPU, 512MB DDR 内存, Windows 2000 Professional 操作系统。

### 5.1 IIII 序列编码实验

实验参数设置如下: 允许 RDO; 采用 CABAC 熵编码; GOP 采用 IIII 方式; 允许进行 Hadamard 变换; 量化参数分别设置为 28, 32, 36, 40; 共采用 10 种视频序列(300 帧), 其中 CIF 和 SIF 格式的视频帧率为 30fps(帧/秒), QCIF 格式的视频帧率为 15fps。

实验结果如表 2 所示, 其中  $\Delta T_{Total}$  取值为 4 种量化参数下统计值的平均值;  $\Delta Bit\text{-rate}$  和  $\Delta PSNR$  为 JM10.2 和本文提出算法所获得的率失真曲线差异的平均值, 其计算过程在文献[20]中有详细描述, 此处不再赘述。

表 2 IIII 序列编码实验结果

视频序列	格式	$\Delta T_{Total}$	$\Delta PSNR$	$\Delta Bit\text{-rate}$
News	CIF	- 47.658%	- 0.0751dB	+ 1.097%
Foreman	CIF	- 47.128%	- 0.0761dB	+ 1.521%
Coastguard	CIF	- 49.168%	- 0.1481dB	+ 2.966%
Mobile	SIF	- 53.057%	- 0.0901dB	+ 1.068%
Stefan	SIF	- 51.492%	- 0.1251dB	+ 1.605%
NewsQ	QCIF	- 50.767%	- 0.071dB	+ 0.932%
CarphoneQ	QCIF	- 48.459%	- 0.091dB	+ 1.421%
ForemanQ	QCIF	- 49.763%	- 0.0941dB	+ 1.575%
CoastguardQ	QCIF	- 51.288%	- 0.1371dB	+ 2.598%
MobileQ	QCIF	- 53.542%	- 0.1031dB	+ 1.292%
平均		- 50.232%	- 0.101dB	+ 1.607%

表 3 给出了本文算法和其他快速帧内编码算法的比较, 在全 1 帧编码设置下, 本文提出的码率估计技术节省的编码时间和文献[4, 5]具有可比性, 但是对失真

和码率造成的损失却更小。值得说明的是, 由于本文算法和文献[4, 5]中的算法分别针对帧内模式选择的不同部分进行了加速, 因此可以结合起来进一步降低帧内编码的计算复杂度。

图 4 给出了 CIF 格式的 Foreman 序列通过 JM10.2 和本文算法进行 IIII 序列编码后的率失真曲线, 该图说明本文算法和 JM10.2 具有近似的率失真性能, 从另外一个方面证实了码率估计方法的有效性。

表 3 本文算法和其他快速帧内编码算法的比较

编码序列	算法	$\Delta T_{Total}$	$\Delta PSNR$	$\Delta Bit\text{-rate}$
III 序列	文献[4]	- 59.825%	- 0.244dB	+ 3.712%
	文献[5]	- 48.332%	- 0.238dB	+ 3.414%
	本文	- 50.232%	- 0.101dB	+ 1.607%

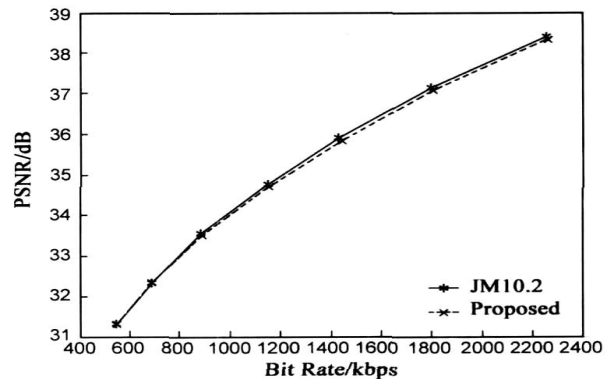


图 4 Foreman 序列 IIII 编码的率失真曲线

### 5.2 IPPP 序列编码实验

实验参数设置如下: 允许 RDO; 采用 CABAC 熵编码; GOP 采用 IPPP 方式; 参考帧数设置为 1; 采用全搜索 FS 算法, 搜索窗口定义为 16; 允许进行 Hadamard 变换; 量化参数分别设置为 22, 28, 34, 40; 共采用了 10 种视频序列(300 帧), 其中 CIF 和 SIF 格式视频的帧率为 30fps, QCIF 格式视频的帧率为 15fps。

实验结果如表 4 所示, 其中  $\Delta T_{Total}$  和  $\Delta T_{RD\text{Cost}}$  取值为 4 种量化参数下统计值的平均值;  $\Delta Bit\text{-rate}$  和  $\Delta PSNR$  为 JM10.2 和本文提出算法所获得的率失真曲线差异的平均值<sup>[20]</sup>。

表 5 给出了本文算法和其他快速帧间编码算法的比较, 本文提出的码率估计技术对 PSNR 和码率的影响和文献[16]具有可比性, 而节省的总编码时间更多。其原因在于, 文献[16]中的码率估计是基于 CAVLC 熵编码机制进行的, 相比 CAVLC, CABAC 熵编码机制在带来更高压缩比的同时, 也大大增加了编码器的计算复杂度, RDO 模式选择的计算时间在总编码时间中占的比例更大。因此, 在采用码率估计技术后, 本文获得的加速比较大, 相对于文献[9, 10]中的快速算法, 本文算法在

编码时间方面具有可比性,而且具有更好的率失真性能.同样,本文算法和文献[9,10]中的算法可以结合起来进一步降低视频编码的计算复杂度.

表 4 IPPP 序列编码实验结果

视频序列	格式	$\Delta T_{Total}(\Delta T_{RDCost})$	$\Delta PSNR$	$\Delta Bitrate$
News	CIF	-30.06% (-48.21%)	-0.0521dB	+1.065%
Foreman	CIF	-30.36% (-50.45%)	-0.0591dB	+1.460%
Coastguard	CIF	-32.60% (-50.41%)	-0.0661dB	+1.733%
Mobile	SIF	-38.23% (-54.67%)	-0.037dB	+0.589%
Stefan	SIF	-35.69% (-53.12%)	+0.0651dB	-1.236%
NewsQ	QCIF	-33.65% (-51.63%)	-0.0691dB	+1.187%
CarphoneQ	QCIF	-32.34% (-50.57%)	-0.0481dB	+0.998%
ForemanQ	QCIF	-32.57% (-50.70%)	-0.0521dB	+1.051%
CoastguardQ	QCIF	-33.99% (-51.95%)	-0.0491dB	+1.306%
MobileQ	QCIF	-39.05% (-55.50%)	-0.0261dB	+0.561%
平均		-33.85% (-51.72%)	-0.0391dB	+0.871%

表 5 本文算法和其他快速帧间编码算法的比较

编码序列	算法	$\Delta T_{Total}$	$\Delta PSNR$	$\Delta Bitrate$
IPPP 序列	文献[9]	-30.821%	-0.0791dB	+1.654%
	文献[10]	-28.710%	-0.0781dB	+2.780%
	文献[16]	-16.96%	-0.0491dB	+1.031%
	本文	-33.85%	-0.0391dB	+0.871%

图 5 给出了 CIF 格式的 Foreman 序列通过 JM10.2 和本文算法进行 IPPP 序列编码后的率失真曲线,该图说明本文算法和 JM10.2 具有近似的率失真性能.

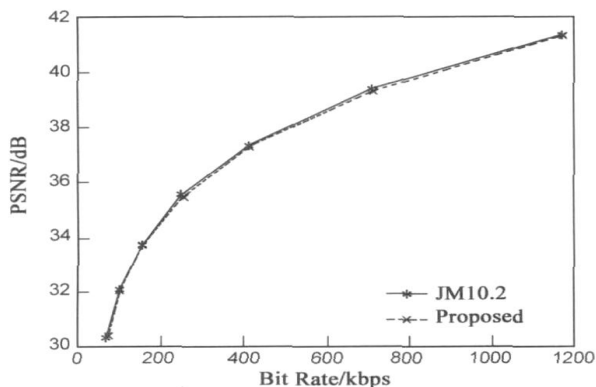


图 5 Foreman 序列 IPPP 编码的率失真曲线

## 6 结论

RDO 模式选择在先进视频编码机制中具有重要的作用,结合 CABAC 熵编码机制,可以明显提高视频压缩率,但同时也增加了编码器的计算复杂度.本文第一次提出了一种 CABAC 熵编码机制下的变换域码率估计技术,以量化变换系数和运动向量差为基础对编码块的熵

编码位数进行有效估计;以此为基础,本文给出了基于码率估计的快速模式选择算法,采用码率估计代替 RDO 模式选择中的熵编码过程,减小了模式选择的计算时间.模拟结果显示,提出的 CABAC 熵编码机制下的变换域码率估计技术在对编码性能影响不大的情况下,减少了模式选择中 51% 的率失真计算时间,采用全搜索运动估计算法时,节省了 33% 的总编码时间.

进一步的工作包括在以下几个方面对算法进行优化研究:细化码率估计模型,寻找码率估计准确度和计算复杂度的最佳折衷;变换域码率估计技术和其他文献提出的快速算法具有互补性,开发结合两种技术的快速模式选择算法进一步提高编码器的性能.

## 参考文献:

- [1] ISO/IEC FDIS 14 496-10, Information Technology - Coding of Audio-Visual Objects - Part 10: Advanced Video Coding, Final Draft International Standard[S]. 2003.
- [2] G Sullivan, T Wiegand, K P Lim. Joint model reference encoding methods and decoding concealment methods[A]. 9th JVT Meeting (JVT I049d0) [C]. San Diego, 2003.
- [3] J L Nunez Yanez, V A Chouliaras. Hardware assisted rate distortion optimization with embedded CABAC accelerator for the H.264 advanced video codec[J]. IEEE Transactions on Consumer Electronics, 2006, 52(2): 590-597.
- [4] F Pan, X Lin. Fast mode decision algorithm for intra prediction in H.264 video coding[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2005, 15(7): 813-822.
- [5] R Su, G Z Liu, T Y Zhang. Fast mode decision algorithm for intra prediction in H.264/AVC[A]. Proceedings of the 31st International Conference on Acoustics, Speech, and Signal Processing[C]. Toulouse: IEEE Signal Processing Society, 2006. 921-924.
- [6] 李世平, 蒋刚毅, 郁梅. 快速帧内预测模式选择新方法[J]. 电子学报, 2006, 34(1): 141-146.  
Li Shiping, Jiang Gangyi, Yu Mei. New fast mode selection for intra prediction[J]. Acta Electronica Sinica, 2006, 34(1): 141-146. (in Chinese)
- [7] 贾克斌, 谢晶, 方湘. 一种基于自相关法的 H.264 高效帧内预测算法[J]. 电子学报, 2006, 34(1): 152-154.  
Jia Kebin, Xie Jing, Fang Xiang. An efficient intra frame prediction algorithm based on autocorrelation for H.264/AVC[J]. Acta Electronica Sinica, 2006, 34(1): 152-154. (in Chinese)
- [8] 宋彬, 常义林, 周宁兆. 基于 H.264 帧内预测的快速算法[J]. 电子学报, 2006, 34(1): 31-34.  
Song Bin, Chang Yilin, Zhou Ningzhao. Fast intra prediction method based H.264[J]. Acta Electronica Sinica, 2006, 34(1): 31-34. (in Chinese)
- [9] Z P Lin, H T Yu, F Pan. A scalable fast mode decision algo-

- rithm for H. 264 [A]. Proceedings of the 2006 International Symposium on Circuits and Systems [C]. Island of Kos: IEEE Circuits and Systems Society, 2006. 3494– 3497.
- [10] J Bu, S Y Lou. A predictive block size mode selection for inter frame in H. 264 [A]. Proceedings of the 31st International Conference on Acoustics, Speech, and Signal Processing [C]. Toulouse: IEEE Signal Processing Society, 2006. 917– 920.
- [11] T Chiang, Y Q Zhang. A new rate control scheme using quadratic rate distortion model [J]. IEEE Transaction on Circuits and Systems for Video Technology, 1997, 7(1): 246– 250.
- [12] J Ribas Corbera, S Lei. Rate control in DCT video coding for low delay communications [J]. IEEE Transaction on Circuits and Systems for Video Technology, 1999, 9(1): 172– 185.
- [13] Z He, Y K Kim, S K Mitra. Low delay rate control for DCT video coding via  $\rho$  domain source modeling [J]. IEEE Transaction on Circuits and Systems for Video Technology, 2001, 11(8): 928– 940.
- [14] Z He, S K Mitra. Optimum bit allocation and accurate rate control for video coding via  $\rho$  domain source modeling [J]. IEEE Transaction on Circuits and Systems for Video Technology, 2002, 12(10): 840– 849.
- [15] Z He, S K Mitra. A linear source model and a unified rate control algorithm for DCT video coding [J]. IEEE Transaction on Circuits and Systems for Video Technology, 2002, 12(11): 970– 982.
- [16] Y K Tu, J F Yang. Efficient rate distortion estimation for H. 264 coders [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2002, 12(11): 970– 982.
- [17] M G Sarwer, L M Po. Fast bit rate estimation for mode decision of H. 264/AVC [J]. Transactions on Circuits and Systems for Video Technology, 2007, 17(10): 1402– 1407.
- [18] D Marpe, H Schwarz, T Wiegand. Context based adaptive binary arithmetic coding in the H. 264/AVC video compression standard [J]. IEEE Transaction on Circuits and Systems for Video Technology, 2003, 13(7): 620– 636.
- [19] JM10. 2. Reference Software of JVT. <http://bs.hhi.de/suehring/tm/download/jm10.2.zip>, 2006– 01– 06
- [20] G Bjontegaard. Calculation of average PSNR differences between RD curves (VCEG-M33) [A]. 13th VCEG Meeting [C]. Austin, 2001.

#### 作者简介:



孙书为 男, 1979 年生于山东临清, 2003 年毕业于国防科技大学计算机学院, 获工学硕士学位, 现为国防科技大学计算机学院博士生, 主要研究方向为视频编解码算法和高性能微处理器设计. E-mail: sws97@163.com

陈书明 男, 1961 年生于安徽六安, 1993 年毕业于国防科技大学计算机系, 获工学博士学位, 现为国防科技大学计算机学院教授, 博士生导师. 主要研究方向为高性能微处理器设计和超深亚微米 VLSI 设计理论与技术. E-mail: smchen@nudt.edu.cn

刘祥远 男, 1977 年生于江西会昌, 2007 年毕业于国防科技大学计算机学院, 获工学博士学位, 主要研究方向为超深亚微米 VLSI 设计理论与技术.