

# 基于 $\rho$ 域源模型的 H.264 宏块行层码率控制算法

徐平 朱善安

(浙江大学电气工程学院, 杭州 310027)

**摘要** 在 H.264 中, 对于一个宏块行  $\rho$  与比特率仍然满足线性关系。基于此, 提出了基于  $\rho$  域源模型的 H.264 宏块行层码率控制算法。在宏块行间, 根据第一次 RDO 时各个宏块行所耗用比特数的百分比来分配比特数。实验结果表明, 与 JM8.4 中采用的码率控制算法相比, 本文提出的宏块行层码率控制算法在保证基本相同的编码效率情况下, 能显著降低比特数估计偏差, 更好地跟踪目标码率。

**关键词** H.264  $\rho$  域源模型 码率控制 宏块行 比特分配

中图法分类号: TP391 文献标识码: A 文章编号: 1006-8961(2006)12-1806-07

## Macroblock Slice Layer Rate Control Algorithm for H.264 via $\rho$ -Domain Source Modeling

XU Ping, ZHU Shan-an

(College of Electrical Engineering, Zhejiang University, Hangzhou 310027)

**Abstract** Extensive experimental results show that  $\rho$  is still linear with the bit rate of a macroblock slice in H.264. Based on the above conclusion, we have successfully developed a macroblock slice layer rate control algorithm for H.264 via  $\rho$ -domain source modeling. The bit allocation of the proposed method is achieved according to the percentage of bit count of each macroblock slice in the first RDO. Experimental results show that the proposed rate control algorithm can guarantee the same coding efficiency, achieve significantly smaller bit rate estimation error and be better matched to the target bit rate than JM8.4 rate control algorithm.

**Keywords** H.264,  $\rho$ -domain source modeling, rate control, macroblock slice, bit allocation

### 1 引言

H.264 是由视频编码专家组(Video Coding Experts Group, VCEG)和运动图像专家组(Moving Picture Experts Group, MPEG)建立的联合视频组(Joint Video Team, JVT)近年来致力开发研究的最新视频压缩国际标准。开发 H.264 的目的是显著地提高压缩性能以及其他一些特性,如容错性和网络友好性等。H.264 通过一系列复杂的技术来实现高压压缩性能,如多模式的运动估计和运动补偿、多参考帧、帧内预测、 $4 \times 4$  的整数变换、内容自适应二进制算术编码、环路滤波以及高效的比特流切换技术等等。

在视频编码和通讯中,两个重要的性能指标是压缩后的码流大小和图像质量。码流大小决定了需

要传输视频数据的带宽,现有的码率控制大多是通过选择适当的量化参数来分配比特,使得图像失真尽可能小。由于 H.264 采用了许多复杂的技术来提高压缩性能,使得 H.264 的构架比以前的标准,如 H.263 和 MPEG2 等更复杂,因而在进行码率控制时,对现有的 H.264 码率控制的分析也更复杂。在 H.264 的编码器中,码率控制和模式选择都需涉及量化参数,在进行码率控制时存在如下蛋鸡悖论:对当前帧中某个宏块进行基于 RDO(rate distortion optimization)的编码,需要通过该宏块的 MAD(mean absolute difference)来决定其量化参数,而该宏块的 MAD 却只能在基于 RDO 的模式选择以后才能得到。目前,最新的 H.264 参考代码采用的码率控制算法是由 Li 等人提出<sup>[1~3]</sup>,该算法将一帧分成若干个基本单元,在模式选择之前,通过采用线性 MAD

模型,用前一帧某一位置基本单元的 MAD 来预测当前帧同一位置基本单元的 MAD 来避免蛋鸡悖论,然后采用二次模型决定其量化参数,再用该量化参数进行基于 RDO 的模式选择和编码。

$\rho$  域的码率控制算法是由 He 等人提出的<sup>[4,5]</sup>,其中  $\rho$  表示变换系数量化后零系数的个数占所有系数的百分数。大量的实验及理论证明得出以下结论<sup>[4,5]</sup>对于视频信号  $\rho$  与纹理部分编码比特率成线性关系。在文献 [4] 中  $\rho$  域的码率控制算法已经在 H. 263 中表现出了极好的性能。通过获得变换系数的分布,可得  $\rho$  与量化参数的一一对应关系。因此,以  $\rho$  为中介,由目标码率可唯一地确定量化参数。

尽管  $\rho$  域的码率控制在 H. 263 等标准中已经取得了极好的压缩性能,但是,由于 H. 264 中引入了一系列复杂的编码技术,使得较难将其直接应用到 H. 264 进行码率控制。Shin 等人<sup>[6]</sup>提出了一种基于 H. 264 JM3.9 平台的帧层  $\rho$  域码率控制算法,并取得了较好的实验结果。本文的基本想法是将 H. 264 的帧层  $\rho$  域码率控制算法进一步扩展到宏块行层,并进行更精确的宏块行层比特分配,以获得更小的编码码流的波动,更精确地跟踪目标码率。

## 2 $\rho$ 域 H. 264 宏块行线性码率模型

在图像压缩国际标准 JPEG, MPEG2, H. 263 和 MPEG4 中,比特率和  $\rho$  存在线性关系,且可用以下线性模型表示<sup>[4,5]</sup>:

$$R(\rho) = \alpha(1 - \rho) \quad (1)$$

其中  $R$  为比特率,  $\rho$  为斜率常量。尽管 H. 264 与以前的视频压缩标准在算法上有很大不同,文献 [6] 通过验证在 H. 264 的一帧中比特率和  $\rho$  仍然满足式 (1) 的线性关系,成功地将其基于  $\rho$  域源模型的方法引入到 H. 264 帧层的码率控制中。在单元层码率控制<sup>[6,7]</sup>之前是帧层码率控制和宏块层码率控制。帧层码率控制可获得较高的编码效率但是造成较大的码流波动,而宏块层的码率控制尽管可最大限度地减小码流的波动,却带来编码效率的损失。因此, Li 等人提出了基本单元的概念,目的是在编码效率和码流的波动之间找到折衷。一帧图像可分为若干基本单元,而每个基本单元由一组宏块构成,每个基本单元共用一个量化参数。一个基本单元可为一帧图像,可为一个宏块,也可为一个宏块行等等。在实时视频通讯中,一个基本单元的大小为一个宏块行<sup>[1,2]</sup>。本文是

将基于  $\rho$  域源模型的方法扩展到 H. 264 宏块行层码率控制,以实现更为精确的码率控制和较小的码流波动。下面通过实验来验证在 H. 264 中,对于单独的宏块行比特率和  $\rho$  也存在线性关系。图 1 给出了“news”QCIF 视频序列第 2 帧中每个宏块行的  $R(\rho)$  曲线,帧率为 15fps,量化参数  $QP$  的取值范围为 0 ~ 51。从图中可以明显看出  $R(\rho)$  近似为一条经过 (1, 0) 点的直线,线性关系明显。从本序列的其他帧及其他序列的大量实验结果来看,也同样表明有此结论。

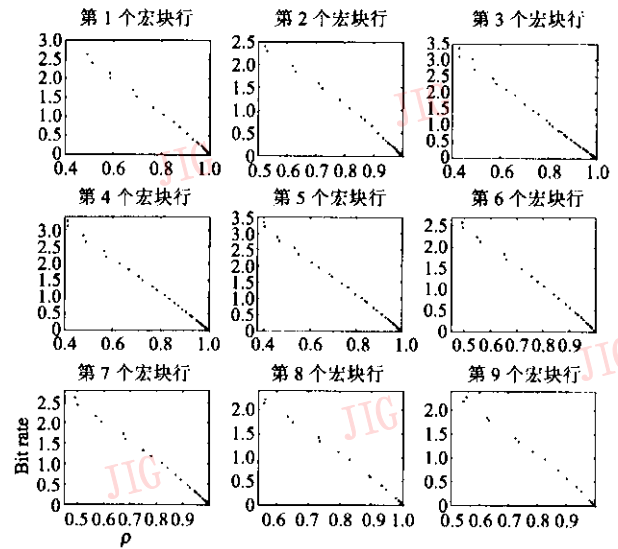


图 1 H. 264 编码中宏块行的比特率和  $\rho$  线性关系

Fig. 1 Linear relationship between bitrate and  $\rho$  of macroblock row in H. 264

为了更好地验证宏块行层的线性特性,研究各宏块行比特率  $R$  和  $\rho$  之间的相关系数  $C$ <sup>[7]</sup>:

$$C = \frac{\frac{1}{K} \sum_{i=0}^{K-1} (\rho_i - \bar{\rho})(R_i - \bar{R})}{\sqrt{\frac{1}{K} \sum_{i=0}^{K-1} (\rho_i - \bar{\rho})^2} \cdot \sqrt{\frac{1}{K} \sum_{i=0}^{K-1} (R_i - \bar{R})^2}} \quad (2)$$

其中,

$$\bar{\rho} = \frac{1}{K} \sum_{i=0}^{K-1} \rho_i, \bar{R} = \frac{1}{K} \sum_{i=0}^{K-1} R_i \quad (3)$$

当  $C$  等于  $\pm 1$  时,表明  $R$  和  $\rho$  之间完全成线性关系;当  $C$  越接近  $\pm 1$  时,表明  $R$  和  $\rho$  之间越接近线性关系。采用“foreman”,“carphone”,“silent”和“coastguard”4 个 QCIF 标准测试序列进行宏块行线性性能测试,其中,目标码率为 64kbps,帧率为 15fps,编码的帧数为 100f,量化参数的范围为 0 ~  $K$  (即  $K=51$ )。图 2 给出了上述 4 个序列各个宏块行  $-C$  值。从实验的结果来看,  $-C$  总是大于 0.988,因而认

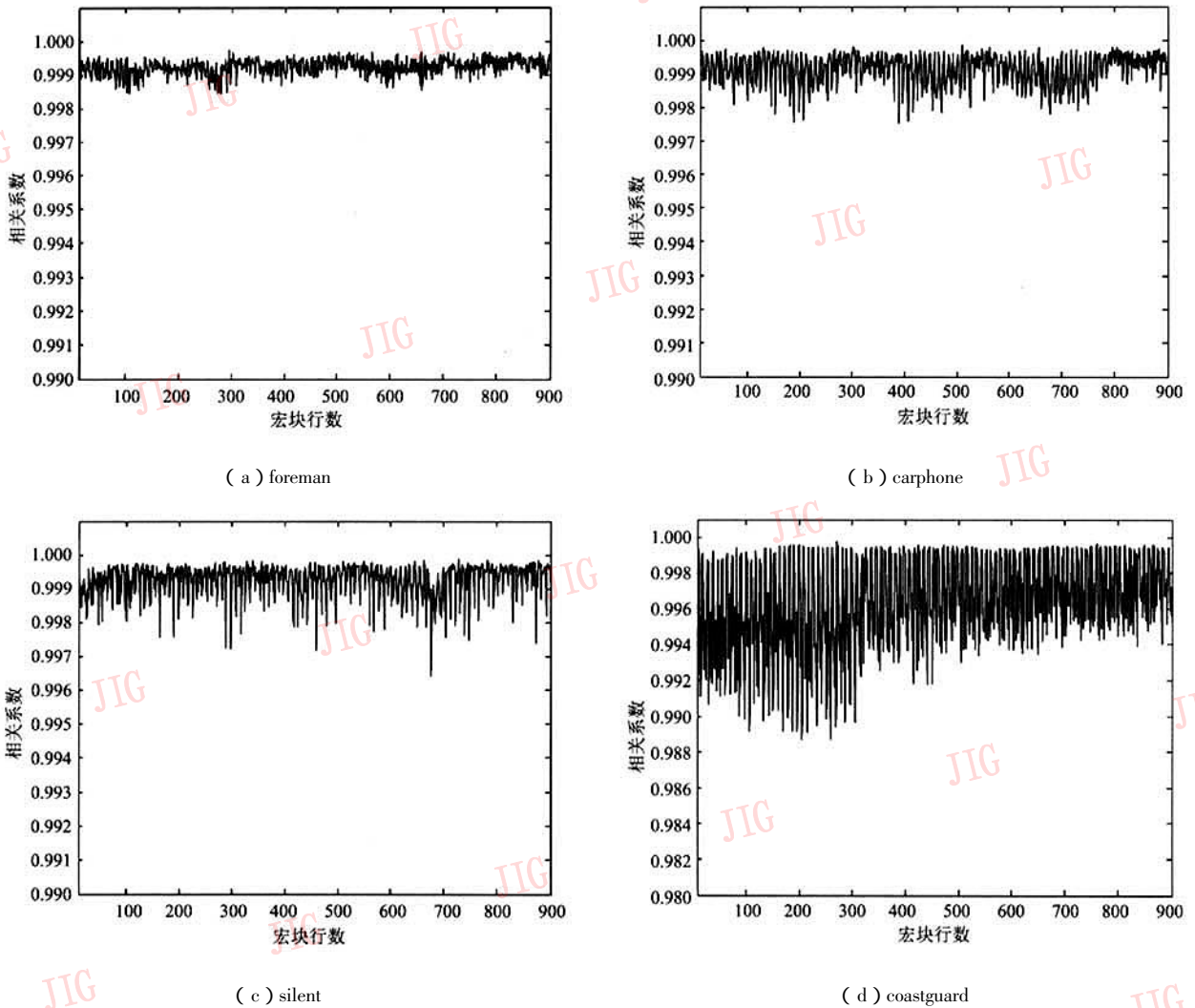


图 2 每个宏块行的比特率  $R$  和  $\rho$  之间的相关系数 ( $-C$ )

Fig. 2 The correlation coefficient ( $-C$ ) between  $R$  and  $\rho$  of each macroblock row

为  $R$  和  $\rho$  之间在宏块行层存在显著的线性特性。

### 3 基于 $\rho$ 域源模型的宏块行层码率控制算法

基于  $\rho$  域源模型的宏块行层码率控制算法可概括为：

(1) 确定当前帧的目标比特数；

(2) 采用前一帧相同位置宏块层的量化参数作为当前帧当前宏块行的初始量化参数, 并对其进行第 1 次基于 RDO 的编码, 获得各宏块行变换系数的分布、编码比特数和初始  $\rho$  值；

(3) 根据当前帧可用编码比特数和在第 1 次 RDO 过程中得到的当前宏块行的编码比特数所占

百分数来确定当前宏块行的目标比特数；

(4) 根据  $\rho$  域源模型, 确定各宏块行的目标  $\rho$  值, 并确定相应目标量化参数；

(5) 当目标量化参数不等于初始量化参数时, 采用目标量化参数对当前宏块行进行第 2 次基于 RDO 的编码；

(6) 更新当前帧剩余可用的编码比特数。

#### 3.1 帧层比特分配策略

帧层的码率控制分为预编码和后编码两个阶段<sup>[2,3]</sup>。预编码的目的是用来确定帧层目标比特数和帧层的量化参数, 并对该帧中的所有宏块进行 RDO；而后编码则是用来更新模型参数和决定跳帧数目。目标码流大小采用流量往返模型 (fluid traffic model) 来估计。为简化起见, 这里仅考虑 IPPP 形式的 GOP

(group of picture) 即第 1 帧为 I 帧,其余的为 P 帧。

初始分配给第  $i$  个 GOP 的比特数为

$$B_i(1) = \frac{R_i(1)}{f} N_i + B_{i-1}(N_{i-1}) \quad (4)$$

虚拟缓存的占用由下式一帧一帧地更新:

$$B_i(j) = B_i(j-1) - b_i(j-1) + \frac{R_i(j) - R_i(j-1)}{f} (N_i - j + 1) \quad (5)$$

其中  $B_i(j)$  表示编码完第  $i$  个 GOP 中第  $j$  帧以后虚拟缓存所占的大小,  $N_i$  表示第  $i$  个 GOP 的长度,  $b_i(j)$  表示第  $i$  个 GOP 中第  $j$  帧的压缩码流大小,  $R_i(j)$  表示第  $i$  个 GOP 中第  $j$  帧的目标码率,  $f$  表示帧率。

分配给当前 P 帧的目标比特数由下述步骤得到:

(1) 定义编码每个 P 帧后占用的目标缓冲大小

文献 [2][3] 需要采用首先按照初始的量化参数来编码第 1 个 P 帧来建立 MAD 的预测模型,而本文采用的  $\rho$  域的码率控制算法可对第 1 个 P 帧的量化参数进行调整来实现码率控制。因此,初始占用的目标缓冲大小为

$$S_i(1) = V_i(1) \quad (6)$$

其中,  $V_i(1)$  表示 1 帧编码完以后实际占用的缓存大小,  $S_i(j)$  表示第  $j$  帧占用的目标缓存大小。第  $j+1$  帧占用的目标缓存大小为

$$S_i(j+1) = S_i(j) - \frac{S_i(1) - V_i/8}{N_p} \quad (7)$$

其中,  $V_i$  表示缓存总的大小,  $N_p$  表示第  $i$  个 GOP 中剩余未编码的 P 帧数。

(2) 计算当前 P 帧的目标比特数

根据线性跟踪理论,分配给当前 P 帧的目标比特数由该帧占用的目标缓存大小、帧率、可用的带宽大小和实际的缓存占用大小决定:

$$\tilde{T}_i(j+1) = \frac{R_i(j+1)}{f} + \gamma (S_i(j+1) - V_i(j+1)) \quad (8)$$

其中,  $\gamma$  为一常量,典型值为 0.125。同时,在该 GOP 中剩余的比特数也需考虑:

$$\hat{T}_i(j+1) = \frac{B_i(j)}{N_p} \quad (9)$$

则当前 P 帧的目标比特数为

$$T_i(j+1) = \beta \hat{T}_i(j+1) + (1 - \beta) \tilde{T}_i(j+1) \quad (10)$$

其中  $\beta$  为一常量,典型值为 0.875。

为了保持编码帧的质量,  $T_i(j+1)$  需加如下约束条件:

$$T_i(j+1) = \max \left\{ T_i(j+1) m_{\text{hdr},i}(j) + \frac{R_i(j+1)}{4f} \right\} \quad (11)$$

其中  $m_{\text{hdr},i}(j)$  表示前一个 P 帧的头信息和运动矢量占用的比特数。

考虑到虚拟参考解码器 (hypothetical reference decoder, HRD), 当前 P 帧目标比特数可进一步表示为

$$T_i(j+1) = \min \{ \max \{ Z_i(j+1), T_i(j+1) \}, U_i(j+1) \} \quad (12)$$

其中  $Z_i(j+1)$  和  $U_i(j+1)$  分别表示当前帧的分配比特数的下界和上界<sup>[2,3]</sup>。

### 3.2 宏块行层比特分配策略

在第 1 次基于 RDO 的编码选择最优模式和获得变换系数分布的过程中,将每个宏块的最优模式下的比特数等信息暂存下来。宏块层的比特分配依据各个宏块行耗用的比特数占总的编码比特数的百分数来分配:

$$\bar{b}_{l,i}(j+1) = T_{r,i}(j+1) \frac{\hat{b}_{l,i}(j+1)}{\sum_{k=1}^N \hat{b}_{k,i}(j+1)} \quad (13)$$

其中  $\bar{b}_{l,i}(j+1)$  表示分配给当前宏块行  $l$  的目标比特数,  $T_{r,i}(j+1)$  表示当前帧中剩余未编码宏块行可用的比特数,初始值为  $T_i(j+1)$ ,总的宏块行数为  $N$ ,  $\hat{b}_{l,i}(j+1)$  表示第 1 次 RDO 编码过程中该宏块行耗用的比特数。

除去头部信息后,宏块行纹理部分的目标比特数为

$$b_{l,i}(j+1) = \begin{cases} \text{当 } \sum_{k=1}^N \hat{b}_{k,i}(j+1) > 0 \\ \bar{b}_{l,i}(j+1) - \hat{m}_{l,i}(j+1) \frac{T_{r,i}(j+1)}{\sum_{k=1}^N \hat{b}_{k,i}(j+1)} \\ \text{当 } \sum_{k=1}^N \hat{b}_{k,i}(j+1) = 0 \\ \bar{b}_{l,i}(j+1) - \hat{m}_{l,i}(j+1) \end{cases} \quad (14)$$

其中  $b_{l,i}(j+1)$  表示宏块行纹理部分的目标比特数,  $\hat{m}_{l,i}(j+1)$  表示第 1 次 RDO 过程中该宏块行头部分耗用的比特数。

### 3.3 $\rho$ 值的计算

在 H.264 中,对  $4 \times 4$  的整数到整数变换后的系数采用非线性标量量化。量化参数  $QP$  的范围为 0~51,对于亮度和色度略有不同,但可由亮度的量

化参数  $QP_{luma}$  推出色度的量化参数  $QP_{chroma}$ 。  $QP$  每增加 6, 量化步长  $Q_{step}$  翻一倍。

对  $(i, j) 0 \leq i, j < 4$  位置的变换系数  $x_{i,j}$  的量化可用下式表示:

$$\hat{x}_{i,j}(QP) = Round\left(\frac{x_{i,j}M(i, j) + \lambda 2^{qbits}}{2^{qbits}}\right) \quad (15)$$

$$qbits = 15 + Floor(QP/6) \quad (16)$$

其中  $Round$  表示四舍五入取整运算,  $Floor$  表示求不大于本身的最大整数运算,  $M(i, j)$  依赖于  $QP$  值,  $QP$  每增加 6,  $M(i, j)$  值不变, 如表 1<sup>[8]</sup> 所示;  $\lambda$  为死区阈值, 当宏块为帧内编码模式时取  $1/3$ , 当宏块为帧间编码模式时取  $1/6$ 。

表 1  $M(i, j)$  的值  
Tab. 1 Value of  $M(i, j)$

QP	位置 (0 0)(0 2), (2 0)(2 2)	位置 (1 1)(1 3), (3 1)(3 3)	其他位置
0	13 107	5 243	8 066
1	11 916	4 660	7 490
2	10 082	4 194	6 554
3	9 362	3 647	5 825
4	8 192	3 355	5 243

用  $D(x)$  表示宏块行中变换系数的分布, 则  $\rho$  值的计算可用下式表示:

$$\rho(QP) = \frac{1}{L} \sum_{|x_{i,j}| \leq (1-\lambda) 2^{qbits}/M(i, j)} D(x) \quad (17)$$

其中  $L$  为宏块行中总的变换系数的个数。

### 3.4 量化参数的确定

#### 3.4.1 I 帧量化参数的确定

第 1 个 GOP 中 I 帧的初始量化参数由通道带宽和 GOP 的帧数确定:

$$QP_0(0) = \begin{cases} 35 & b \leq l_1 \\ 25 & l_1 < b \leq l_2 \\ 20 & l_2 < b \leq l_3 \\ 10 & b > l_3 \end{cases} \quad (18)$$

$$b = R_0(0) / f N_{pixel}$$

其中  $b$  为每像素点的平均比特数,  $N_{pixel}$  为一帧中总的像素点数。推荐值为: 对于 QCIF 格式视频  $l_1 = 0.1$   $l_2 = 0.3$   $l_3 = 0.6$  对于 CIF 格式视频  $l_1 = 0.2$   $l_2 = 0.6$   $l_3 = 1.2$  其他视频格式  $l_1 = 0.6$   $l_2 = 1.4$   $l_3 = 2.4$ 。

其他的 GOP 中 I 帧的量化参数  $QP_l(0)$  可由下式计算:

$$QP_l(0) = \max \left\{ \min \left\{ \frac{\sum_{j=1}^{N_p} \overline{QP}_{i-1}(j)}{N_p} - \min \left\{ 2 \frac{N_{i-1}}{15} \right\}, \right. \right. \\ \left. \left. QP_{i-1}(0) + 2 \right\}, QP_{i-1}(0) - 2 \right\} \quad (19)$$

#### 3.4.2 P 帧各个宏块行量化参数的确定

首先, 用前一帧相同位置的宏块行的量化参数  $QP_{l,i}(j)$  对当前编码帧的所有宏块进行第 1 次基于 RDO 的编码, 统计各宏块行变换系数的分布  $D(x)$ , 根据式 (17) 得各个宏块行的初始  $\hat{\rho}_{l,i}(j+1)$  统计最优模式下比特数  $\hat{b}_{l,i}(j+1)$  及头部分比特数  $\hat{m}_{l,i}(j+1)$ 。

量化参数的确定可分为以下 3 种情况:

(1) 当为第 1 个宏块行时, 量化参数取为前一帧各宏块行的平均量化参数:

$$QP_{0,i}(j+1) = \overline{QP_l(j)} \quad (20)$$

(2) 当  $T_{r,i}(j+1) < 0$  时, 当前宏块行的量化参数应大于前一个宏块行的量化参数, 即

$$QP_{l,i}(j+1) = QP_{l,i-1}(j+1) + 1 \quad (21)$$

为了保持视觉质量的连续性, 量化参数可进一步约束为

$$QP_{l,i}(j+1) = \max \left\{ \overline{QP_l(j)} - 2, \min \{ QP_{l,i}(j+1), \overline{QP_l(j)} + 2 \} \right\} \quad (22)$$

(3) 其他的情况下, 分为以下几步来确定各个宏块行的量化参数:

① 由 3.1 和 3.2 节帧层及宏块行层比特分配策略确定各个宏块行纹理部分的目标比特数  $b_{l,i}(j+1)$ ;

② 由  $R-p$  线性模型式 (1) 可得各宏块行的目标  $\rho$ :

$$\rho_{l,i}(j+1) = \frac{\hat{b}_{l,i}(j+1) - \hat{m}_{l,i}(j+1) - b_{l,i}(j+1)(1 - \hat{\rho}_{l,i}(j+1))}{\hat{b}_{l,i}(j+1) - \hat{m}_{l,i}(j+1)} \quad (23)$$

由对所有宏块进行第 1 次基于 RDO 编码过程中得到的各宏块行的变换系数的分布信息和式 (17), 可得目标量化参数  $QP_{l,i}(j+1)$ ;

③ 为了减小图像的块效应, 量化参数需引入如下约束:

$$QP_{l,i}(j+1) = \max \{ QP_{l,i-1}(j+1) - 1, \min \{ QP_{l,i}(j+1), QP_{l,i-1}(j+1) + 1 \} \} \quad (24)$$

为了保持图像的视觉质量, 需进一步加入式 (22) 的约束条件;

④ 如果  $QP_{l,i}(j+1) \neq QP_{l,i}(j)$ , 采用量化参数

$QP_{i,j}(j+1)$  对当前宏块行的所有宏块进行第 2 次基于 RDO 的编码, 否则  $QP_{i,j}(j+1) = QP_{i,j}(j)$ ;

⑤ 统计该宏块行耗用的编码比特数, 并更新前帧编码剩余的可用比特数  $T_{r,i}(j+1)$ 。

### 4 实验结果

本文算法的实现是在 H. 264 JM8.4 参考代码上进行修改的。实验采用了 news, foreman, carphone, trevor, claire 和 silent 等 QCIF 标准测试序列, 它们的纹理和运动复杂度各不相同。实验条件如下: 采用 Exp-Golomb 编码, 除 trevor 序列的 GOP 长度为 75f 以外, 其余序列的 GOP 长度为 100f, 帧率为 15fps, 参考帧为 5f, -32 到 32 的搜索范围。除 news 和 silent 序列的目标码率为 64kbps 以外, 其余序列的目标码率为 128kbps。图 3 分别为此 6 个序列在上述实验条件下, 本文算法与 JM8.4 采用的码率控制算法每帧的编码比特数比较。其中, JM8.4 采用的码率控制算法中的一个单元设为 11 个宏块大小, 即一个宏块行。从上述不同测试序列的实验结果可以明显地看出, 本文算法每帧编码比特数的波动比 JM8.4 小。为了定量地分析码率控制算法跟踪目标比特数的精

度, 采用平均比特数估计偏差  $MBEE^{[6]}$  (mean bitrate estimation error) 对其进行性能分析:

$$MBEE = \frac{1}{N_F} \sum_{i=0}^{N_F-1} \frac{|R_{i,target} - R_{i,encoded}|}{R_{i,target}} \quad (25)$$

其中,  $N_F$  和  $i$  分别表示总的帧数和编码帧的索引号,  $R_{i,target}$  和  $R_{i,encoded}$  分别表示第  $i$  帧的目标比特数和实际编码比特数。表 2 给出了 6 个标准测试序列的平均比特数估计偏差和平均信噪比 (PSNR) 的实验结果比较。可以看出, 在与 JM8.4 基本相同的平均信噪比的情况下, 本文算法编码比特数的波动显著减小。因此, 本文算法的编码比特数波动小, 跟踪目标码率的性能明显优于 JM8.4。

表 2 平均比特率估计偏差和平均信噪比的比较

Tab. 2 Comparison of mean bitrate estimation error and mean PSNR

测试序列	平均比特率估计偏差		PSNR (dB)	
	JM8.4	本文算法	JM8.4	本文算法
news	0.146	0.044	37.863	37.953
foreman	0.074	0.041	37.860	37.897
carphone	0.071	0.036	40.653	40.692
trevor	0.138	0.077	39.314	39.249
claire	0.062	0.031	48.205	48.203
silent	0.214	0.071	37.411	37.502

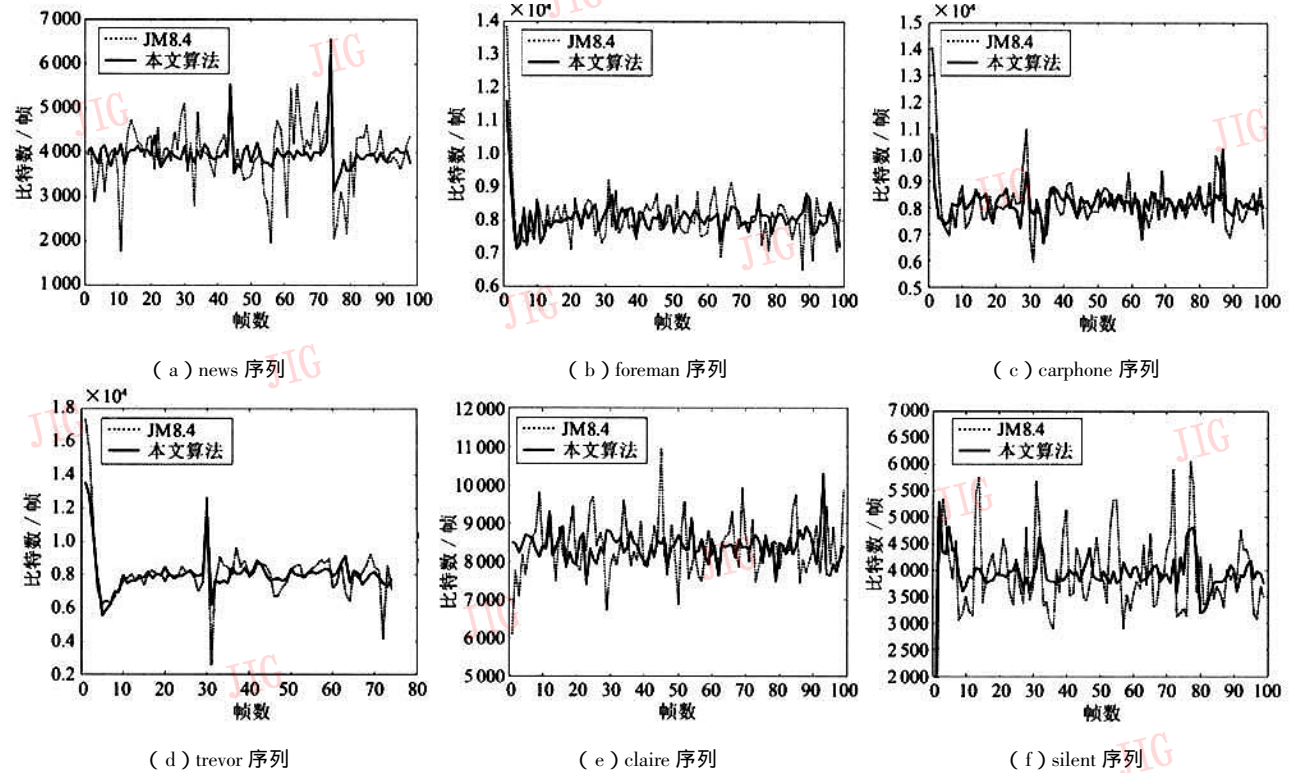


图 3 本文算法与 JM8.4 编码比特数的比较

Fig. 3 Comparison of coding bit count between the proposed method and JM8.4

## 5 结 论

利用在每个宏块行中  $\rho$  与比特率之间仍然成线性关系这一特性,提出了基于  $\rho$  域源模型 H. 264 的宏块行层码率控制算法,并提出了基于第 1 次 RDO 编码过程中的各个宏块行耗用的比特数所占百分数来分配各宏块行的目标比特数的宏块行层比特分配策略。实验结果表明,与 JM8. 4 参考软件采用的码率控制算法相比,本文算法具有在基本保证编码效率的同时,使得编码比特数波动小,能更好地跟踪目标比特数。

致 谢 感谢 Dr. Li Zhengguo 无私地为作者提供 H. 264 码率控制的相关资料。

### 参考文献(References)

- 1 Li Z G, Pan F, Lim K P, *et al.* Adaptive basic unit layer rate control for JVT [ A ]. In : JVT-G012, 7th Meeting [ C ], Pattaya II, Thailand, 2003.
- 2 Li Z G, Gao W, Pan F, *et al.* Adaptive rate control with HRD consideration [ A ]. In : JVT-H014, 8th Meeting [ C ], Geneva, Switzerland, 2003.
- 3 Ma S W, Li Z G, Wu F. Proposed draft of adaptive rate control [ A ]. In : JVT-H017, 8th Meeting [ C ], Geneva, Switzerland, 2003.
- 4 He Z, Kim Y K, Mitra S K. Low-delay rate control for DCT video coding via  $\rho$ -domain source modeling [ J ]. IEEE Transactions on Circuits and Systems for Video Technology, 2001, 11( 8 ): 928 ~ 940.
- 5 He Z, Mitra S K. A unified rate-distortion analysis framework for transform coding [ J ]. IEEE Transactions on Circuits and Systems for Video Technology, 2001, 11( 12 ): 1221 ~ 1236.
- 6 Shin I H, Lee Y L, Park H W. Rate control using linear rate- $\rho$  model for H. 264 [ J ]. Signal Process : Image Communication, 2004, 19( 4 ) 341 ~ 352.
- 7 Zhuang Chu-qiang, Wu Ya-sen. Basis of mathematical statistics with applications [ M ]. Guangzhou : Publishing House of South China University of Technology, 1994. [ 庄楚强, 吴亚森著. 应用数理统计基础 [ M ]. 广州 : 华南理工大学出版社, 1994. ]
- 8 Malvar H S, Hallapuro A, Karczewicz M, *et al.* Low-complexity transform and quantization in H. 264/AVC [ J ]. IEEE Transaction on Circuits and Systems for Video Technology, 2003, 13 ( 7 ): 598 ~ 603.