

# 基于 H. 264/AVC 的帧层码率控制算法

周芦明 刘春生 罗亮

(南京航空航天大学自动化学院, 南京 210016)

**摘要** 在研究和分析了 H. 264/AVC 经典码率控制算法的基础上, 提出了一种新的帧层码率控制算法。该算法充分利用各帧的复杂度以及目标缓冲区优化比特分配, 同时采用了卡尔曼滤波器估计 MAD 的方法。与以往码率控制算法相比, 该算法能更加合理地分配各帧的目标比特数, MAD 预测也更加精确, 从而提高了量化参数 QP 预测的准确性。实验结果表明, 该算法能提高码率控制的精度, 并使图像总体质量有一定程度的提高。

**关键词** H. 264 码率控制 复杂度 卡尔曼滤波

中图分类号: TN919.8 文献标识码: A 文章编号: 1006-8961(2009)09-1813-06

## Frame Layer Rate Control Algorithm for H. 264 Video Encoder

ZHOU Lu-ming, LIU Chun-sheng, LUO Liang

(College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016)

**Abstract** Based on the research and analysis of the H. 264/AVC classical rate control algorithms, a new frame-rate control algorithm is proposed in this paper. The algorithm takes full advantage of the complexity of frame and the target buffer level to optimize the bit allocation, using a kalman filter method to estimate mean absolute difference (MAD). Comparing to the previous rate control algorithms, the algorithm can allocate the bits more rationally, and forecast MAD more accurately, thereby improve the accuracy of estimating the quantization parameter QP. The experimental results show that, compared to JM8.6 original algorithm, the proposed rate control algorithm can improve the precision of rate control, and heighten overall image quality at the same time.

**Keywords** H. 264, rate control, complexity, kalman filtering

## 1 引言

H. 264/AVC 是由 ISO/IEC MPEG 和 ITU-T VCEG 共同建立的联合视频工作组制定的新一代视频编码标准。与以前的标准相比, 它具有压缩效率高, 网络亲和性好等优点。H. 264/AVC 采用基于宏块和运动补偿的方案, 在保持原有编码体系一系列优点的同时, 引进了许多新的技术, 包括  $4 \times 4$  整数变换, 多模式的帧内预测,  $1/4$  像素精度的运动估计, 宏块的灵活分割, 环路滤波器的功能等<sup>[1]</sup>。

码率控制的目的是调整输出码流的码率, 充分

利用信道带宽, 以便尽可能地保证视频图像的总质量。码率控制作为视频编码的重要组成部分, 现有的视频压缩标准都对其进行了广泛的研究。目前经典的码率控制方法有 MPEG-2 的 TM5, H. 263 的 TMN8 以及 MPEG-4 的 VM8。

现有的 H. 264/AVC 参考代码采用的是由 Li 等人提出的 JVT-G012 提案<sup>[2]</sup>中的算法。该算法采用平均分配的方法分配帧层比特数和线性模型预测当前帧的 MAD (mean absolute difference)。由于各帧的复杂度不同以及线性模型的不准确性, 该算法输出的码率波动性比较大。针对这一问题, 本文提出了一种利用帧的复杂度和目标缓冲区来优化帧层的

比特分配以及采用卡尔曼滤波的方法提高了 MAD 预测准确性的算法。为简单起见,本文中 GOP (group of pictures) 的结构为 IPPP, 即编码时第 1 帧是 I 帧,其他帧是 P 帧。

## 2 H. 264 码率控制策略

### 2.1 R-Q (rate-quantization) 模型

H. 264/AVC 编码标准中的码率控制方法是在 MPEG-4 编码标准的基础上的发展,它的核心是模拟信息率失真特性的二次模型<sup>[3]</sup>,如下式:

$$T - T_H = x_1 \times \frac{MAD}{Q_{step}} + x_2 \times \frac{MAD}{Q_{step}^2} \quad (1)$$

式中, $T$  是预先分配的比特数, $T_H$  是编码头信息及运动矢量所占比特数, $MAD$  是平均绝对误差, $Q_{step}$  是量化步长, $x_1$  和  $x_2$  是模型参数,其值在每编完一帧后都要进行更新。

二次模型的引入提高了 H. 264/AVC 的 QP (quantization parameter) 预测的准确性。由式(1)可以看出,更合理的比特分配和更精确的 MAD 估计意味着更准确的量化参数 QP 预测。

### 2.2 MAD 预测的线性模型

JVT-G012 提案中,帧层和基本层均采用线性模型来预测当前图像的 MAD 值,即通过前一图像相同位置的 MAD 值,预测当前图像的 MAD 值。线性预测模型如下式:

$$MAD_c = \alpha_1 \times MAD_p + \alpha_2 \quad (2)$$

式中, $MAD_c$  和  $MAD_p$  分别表示当前图像 MAD 的预测值和前一图像 MAD 的实际值, $\alpha_1$  和  $\alpha_2$  是模型参数,其初始值分别设为 1 和 0。每个图像或基本单元编码完成后,模型参数需要更新。

### 2.3 帧层比特分配

H. 264/AVC 的码率控制算法中,P 帧的目标比特预分配主要受到目标缓冲区的 BufferFullness 以及 GOP 剩余可分配的比特数两个方面的影响。P 帧的目标比特数可表示为

$$f(n_{i,j}) = \beta \times \hat{f}(n_{i,j}) + (1 - \beta) \times \hat{f}(n_{i,j}) \quad (3)$$

式中, $f(n_{i,j})$  为第  $i$  个 GOP 中第  $j$  个 P 帧的目标比特数, $\hat{f}(n_{i,j})$  为根据剩余比特数计算出的各帧目标比特数, $\hat{f}(n_{i,j})$  为根据缓冲区状态计算出的目标比特数。 $\beta$  为权重系数。在没有 B 帧的情况下, $\beta$  值设为 0.5。 $\hat{f}(n_{i,j})$  和  $\hat{f}(n_{i,j})$  分别由下式计算:

$$\hat{f}(n_{i,j}) = \frac{T_r(n_{i,j})}{N_{p,r}(j-1)} \quad (4)$$

$$\hat{f}(n_{i,j}) = \frac{u(n_{i,j})}{F_r} + 0.5 \times (Tbl(n_{i,j}) - B_c(n_{i,j})) \quad (5)$$

式中, $T_r(n_{i,j})$  为当前 GOP 剩余的比特数, $N_{p,r}(j-1)$  为当前 GOP 中还未编码的 P 帧数, $u(n_{i,j})$  为目标码率, $F_r$  为帧频, $Tbl(n_{i,j})$  和  $B_c(n_{i,j})$  分别表示第  $i$  个 GOP 的第  $j$  个 P 帧编码完成后目标缓冲区和实际缓冲区的 BufferFullness。

## 3 比特分配的优化

### 3.1 基于帧复杂度的比特分配优化

由于在 JVT-G012 提案中,剩余比特实际上是平均分配给未编码的各帧,这样做并没有考虑到各帧的纹理复杂度和场景运动差异,各帧得不到与其复杂度相适应的目标比特数,导致编码后各帧的质量和码率输出波动性很大。在文献[4]中,Min 等人引入  $MAD_{ratio}$  与  $PSNR_{ratio}$  来衡量帧的复杂度,并利用两者一起作为帧目标比特数的加权系数。这种做法取得了比较好的码率控制效果。但经实验发现,当视频序列快速运动与慢速运动分布不均匀时,文献[4]中算法的控制效果欠佳,因此本文提出另一种方案。为了快速平缓两种运动切换时所引起的码率波动,定义:

$$MAD_{ratio,j} = \frac{MADP_j}{\frac{1}{20} \left( \sum_{i=1}^{20} MADA_{j-i} \right)} \quad (6)$$

式中, $MAD_{ratio,j}$  表示的是当前帧的纹理复杂度,其中  $MADP_j$  是根据前一帧的真实 MAD 预测出当前帧的 MAD。 $MADA_{j-i}$  为第  $j-i$  帧的真实 MAD。

$$PSNR_{drop,j} = PSNR_{j-1} - PSNR_{skip,j} \quad (7)$$

$$PSNR_{drop-ratio,j} = \frac{PSNR_{drop,j}}{\frac{1}{20} \left( \sum_{i=1}^{20} PSNR_{drop,j-i} \right)} \quad (8)$$

式中, $PSNR_{drop-ratio,j}$  表示的是当前帧的运动剧烈程度,其中  $PSNR_{j-1}$  是第  $j-1$  帧的真实 PSNR 值。 $PSNR_{skip,j}$  是假设第  $j$  帧被跳过,根据  $j-1$  帧的重建值和第  $j$  帧的原始值计算出的第  $j$  帧的 PSNR 值。定义一个复杂度因子如下:

$$FC = 0.7 \times MAD_{ratio,j} + 0.3 \times PSNR_{drop-ratio,j} \quad (9)$$

在这里  $MAD_{ratio,j}$  和  $PSNR_{drop-ratio,j}$  不再是以所有已编码的 P 帧数据做参考,而是以最近 20 个已编码 P 帧做参考,这样可以有效地利用相邻 P 帧的相关

性自适应调整当前帧的目标比特数。加权系数 0.7 和 0.3 是经反复实验而得到的最佳值。根据  $FC$ , 可以计算出当前帧的目标比特数为

$$\hat{T}_c = \begin{cases} 1.7 \times T_{ave,j} & FC \geq 2.0 \\ [1.1 + 0.8 \times (FC - 1.1)] \times T_{ave,j} & 1.1 \leq FC < 2.0 \\ 0.8 \times FC \times T_{ave,j} & FC < 1.1 \end{cases} \quad (10)$$

$$T_{ave,j} = \frac{R_j}{N_{np}} \quad (11)$$

式中,  $R_j$  为编完第  $j-1$  帧后当前 GOP 剩余的比特数,  $N_{np}$  为当前 GOP 中还未编码的 P 帧数。实验证明这样做能比较好地分配运动状态复杂的视频序列中各帧所需的比特数。

### 3.2 基于目标缓冲区的比特分配优化

在 JVT-G012 中, 各帧目标缓冲区的设置是根据第一个即时解码更新帧和第一个存储帧的编码比特数以及平均图像复杂度来进行的。在第  $i$  个 GOP 的第一个 P 帧编码完成后, 目标缓冲区 BufferFullness 的初始值依照下式设定为

$$T_{tbl}(n_{i,2}) = B_c(n_{i,2}) \quad (12)$$

其中,  $B_c(n_{i,2})$  是第  $i$  个 GOP 的第 2 个 P 帧编码完成后实际缓冲区的 BufferFullness。假如序列中没有 B 帧, 则随后的 P 帧的目标缓冲区 BufferFullness 为

$$T_{tbl}(n_{i,j+1}) = T_{tbl}(n_{i,j}) - \frac{T_{tbl}(n_{i,2})}{N_p - 1} \quad (13)$$

式中,  $N_p$  为 GOP 中 P 帧的总数。

由于 P 帧编码是要参考前面已编码的 P 帧, 一帧图像的编码效果可以直接影响到后续 P 帧的编码质量, 因此为 GOP 前一部分帧分配更多的比特对提高整个 GOP 的压缩质量有一定作用。为了达到这个目的, 可以考虑重新设置目标缓冲区的 BufferFullness, 在这里由下式设置:

$$\tilde{T}_{tbl}(n_{i,j+1}) = \begin{cases} \tilde{T}_{tbl}(n_{i,j}) - \frac{1}{2} \times \frac{T_{tbl}(n_{i,2})}{N_p - 1} & 1 \leq j \leq \frac{N_p}{3} \\ \tilde{T}_{tbl}(n_{i,j}) - \frac{3}{2} \times \frac{T_{tbl}(n_{i,\frac{N_p}{3}})}{N_p} & \frac{N_p}{3} < j \leq N_p - 1 \end{cases} \quad (14)$$

根据式(14), 当前帧的目标比特数为

$$\hat{T}_c = \frac{u(n_{i,j})}{F_r} + 0.5 \times (\tilde{T}_{tbl}(n_{i,j}) - B_c(n_{i,j})) \quad (15)$$

$u(n_{i,j})$  为目标码率,  $\tilde{T}_{tbl}(n_{i,j})$  为重新设置的目标缓冲区的 BufferFullness,  $B_c(n_{i,j})$  为实际缓冲区的 BufferFullness。从上式可以看出, GOP 的前三分之一的 P 帧将比后续 P 帧分配更多的比特。

## 4 基于卡尔曼滤波的 MAD 预测

JVT-G012 提案利用线性模型对 MAD 进行预测, 其模型参数采用线性衰落方法进行更新, 即在一次线性拟合之后, 去除其中误差最大的两个点再进行第二次线性拟合, 从而获得新的模型参数。当选择的拟合点较多时, 计算复杂度较大, 拟合点较少时, 预测结果不够精确。为了降低预测的计算复杂度, 同时提高 MAD 预测的精度, 本文提出用卡尔曼滤波器对 MAD 进行预测。

卡尔曼滤波是以最小均方误差作为预测的最佳准则, 来寻求一套递推估计的算法, 其基本思想是: 采用信号与噪声的状态空间模型, 利用前一时刻的估计值和当前时刻的观测值来更新对状态变量的估计, 求出当前时刻的估计值。它适合于实时处理和计算机运算。在文献[5]、[6]中, 作者将卡尔曼滤波分别应用到运动估计和头信息估计量中, 并获得了显著的效果, 这证明将卡尔曼滤波应用到视频编码中具有可行性和有效性, 为本文提供了理论依据。

卡尔曼滤波针对的是由线性随机差分方程描述的离散控制过程中状态量进行估计的。假设某系统  $k$  时刻的状态变量为  $x$ , 通常状态方程为

$$x_k = A_k x_{k-1} + B_k u_{k-1} + \omega_{k-1} \quad (16)$$

量测方程为

$$y_k = C_k x_k + v_k \quad (17)$$

式中,  $u_{k-1}$  表示前一时刻状态变量的控制输入,  $u_{k-1} \in \mathbf{R}^n$ , 当系统不存在控制输入时,  $u_{k-1}$  为零;  $x_k$  表示状态变量,  $x_k \in \mathbf{R}^n$ ;  $y_k$  表示当前的观察值;  $A_k$  为  $n \times n$  的状态变量增益矩阵;  $B_k$  为  $n \times l$  的控制矩阵;  $C_k$  为  $m \times n$  的量测矩阵; 随机向量  $\omega_{k-1}$  和  $v_k$  分别代表过程和测量噪声, 一般可以假定这两个噪声是相互独立的、均值为零的正态白噪声:

$$p(\omega) \sim N(0, Q_k)$$

$$p(v) \sim N(0, R_k)$$

在实际系统中, 过程噪声的方差  $Q_k$  和测量噪声

的方差  $R_k$  可以时变,这里假设其为常数。对于帧层 MAD 值的预测,将各帧的 MAD 实际值作为状态变量  $x_k$ ,将前一帧的实际值作为观察值  $y_k$ ,  $A_k$  和  $C_k$  都设定为常值 1,状态方程和量测方程如下:

$$x_k = x_{k-1} + \omega_{k-1} \quad (18)$$

$$y_k = x_k + v_k \quad (19)$$

则可得到卡尔曼递推方程公式为

$$\hat{x}'_k = \hat{x}'_{k-1} \quad (20)$$

$$\hat{x}_k = \hat{x}'_k + H_k (y_k - \hat{x}'_k) \quad (21)$$

$$H_k = P'_k (P'_k + R_k)^{-1} \quad (22)$$

$$P'_k = P_{k-1} + Q_{k-1} \quad (23)$$

$$P_k = (I - H_k) P'_k \quad (24)$$

各系数的物理意义:  $\hat{x}'_k$  为校正前估计值,即先验估计;  $\hat{x}_k$  为校正后的估计值,即后验估计;  $H_k$  为卡尔曼的增益矩阵;  $P'_k$  为校正前的 MAD 均方误差,即先验方差;  $P_k$  为校正后的误差均方差,即后验方差;  $Q_k$  为输入噪声方差,  $R_k$  为观测噪声方差。卡尔曼滤波器的输入参数为第一个已编码 P 帧的 MAD,输出则作为各帧的 MAD 预测值。经大量实验结果证明,当  $P_0 = 0.5$ ,  $Q_k = 20$  和  $R_k = 1$  时,预测效果比较好。为了比较预测准确性,这里定义 *error* 为总的预测误差,其由下式计算:

$$error = \sum_{j=0}^{N_p} (MADP_j - MADA_j)^2 \quad (25)$$

式中,  $MADP_j$  表示第  $j$  帧的预测 MAD 值,  $MADA_j$  表示第  $j$  帧的真实 MAD 值,  $N_p$  为 GOP 中 P 帧的总数。

表 1 是利用卡尔曼滤波器预测 MAD 与利用线性模型预测 MAD 的实验结果。测试模型采用 JM (Joint mode) 8.6,测试序列都为 QCIF 格式,长度为 150 帧,帧率为 30 fps,量化参数为 24,搜索范围为 16,参考帧为 10 帧,编码方式为 CAVLC。

表 1 卡尔曼滤波与线性模型的实验结果

Tab. 1 Results of the kalman filter and the linear model

| 序列      | 卡尔曼滤波 error | 线性模型 error |
|---------|-------------|------------|
| Claire  | 0.410 198   | 0.574 524  |
| Silent  | 1.991 773   | 2.553 014  |
| Grandma | 0.959 098   | 1.637 360  |
| Suzie   | 2.114 215   | 2.756 026  |
| Highway | 0.212 144   | 0.249 771  |

从表中数据可以看出卡尔曼滤波器预测 MAD 的总误差明显比线性模型预测的小,基本上后者的

误差比前者大出 20% 以上。这证明卡尔曼滤波器比线性模型预测 MAD 更加准确。

## 5 帧层码率控制算法

综合上述内容,确定 GOP 中 P 帧量化参数的具体步骤如下:

(1) GOP 的 I 帧和第 1 个 P 帧的量化参数采用 JVT-G012 中的方法确定;

(2) 根据式 (10) 和式 (15) 得到  $\hat{T}_c$  和  $\hat{T}_r$ , 后续 P 帧的目标比特数可由两者共同决定,其计算公式如下:

$$T = 0.5 \times \hat{T}_c + 0.5 \times \hat{T}_r \quad (26)$$

(3) 当实际缓冲区的 BufferFullness 超过缓冲区大小的 75% 时,就有发生上溢的危险。为了防止实际缓冲区发生上溢,将当前帧的目标比特数按下式进行调整:

$$T_c = \begin{cases} T & B_c(n_{i,j}) < 0.75 \times Buffersize \\ T - 0.5 \times (B_c(n_{i,j}) - 0.75 \times Buffersize) & \\ & B_c(n_{i,j}) \geq 0.75 \times Buffersize \end{cases} \quad (27)$$

其实 *Buffersize* 表示缓冲区的大小;

(4) 最终当前帧的目标比特数分配如下:

$$T_c = \max\left(T_c, \frac{u(n_{i,j})}{10 \times F_r}\right) \quad (28)$$

(5) 根据卡尔曼滤波器预测当前帧的 MAD 值;

(6) 调用二次 R-D 模型,根据上述步骤中得的目标比特数和 MAD 的预测值计算出当前帧的量化参数 QP,对当前帧进行编码;

(7) 更新参数,检查 GOP 是否已经编码完,编完进入步骤(1),否则进入步骤(2)。

## 6 实验结果

实验使用校验模型 JM8.6 作为测试视频编码软件。共取了 4 个 QCIF 格式的 Clair, Coastguard, Miss-america, Bridge-close 和一个 CIF 格式的 Pair 序列做测试,各视频序列都取 150 帧进行测试,测试条件与测试卡尔曼滤波器时的条件相同。为了比较算法的比特分配的准确性,这里将在文献[7]中定义的码率偏移 *bit\_rate\_deviation* 作为测试内容。观察

新算法的实验结果,并与 JM8.6 原算法进行比较, 其结果由表 2 列出。

表 2 两种算法的结果

Tab.2 Results of new rate control scheme and JM8.6 scheme

| 序列           | 目标码率(kbps) | PSNR(dB) |       | 实际码率(kbps) |        | bit_rate_deviation(%) |       |
|--------------|------------|----------|-------|------------|--------|-----------------------|-------|
|              |            | JM8.6    | NewRC | JM8.6      | NewRC  | JM8.6                 | NewRC |
| Clair        | 64         | 43.49    | 43.53 | 64.35      | 63.99  | 31.25                 | 28.96 |
|              | 48         | 42.18    | 42.19 | 48.39      | 48.05  | 28.05                 | 29.08 |
| Coastguard   | 64         | 29.32    | 29.29 | 63.89      | 64.15  | 33.10                 | 29.94 |
|              | 48         | 28.16    | 28.16 | 49.21      | 48.00  | 59.69                 | 33.42 |
| Miss-america | 64         | 42.77    | 42.79 | 63.87      | 64.03  | 28.61                 | 27.92 |
|              | 48         | 41.80    | 41.83 | 48.12      | 48.15  | 28.86                 | 26.35 |
| Bridge-close | 64         | 35.37    | 35.37 | 64.25      | 63.99  | 26.61                 | 25.49 |
|              | 48         | 34.82    | 34.82 | 48.18      | 48.10  | 18.32                 | 24.17 |
| Pair         | 192        | 31.33    | 31.36 | 191.82     | 192.12 | 22.46                 | 22.64 |
|              | 128        | 29.23    | 29.16 | 131.09     | 128.16 | 37.61                 | 17.12 |

从表 2 可以看出,本文算法的实际码率与目标码率的误差基本保持在 0.2 kbps 以内,这与 JM8.6 原有算法相比,码率控制的精度有着明显的提高。在图像恢复的客观质量方面,数据显示本文算法比原算法有所提高。在为每帧分配的比特准确性方面,本文算法所产生的误差比原算法的小,这说明本文算法能更加合理地分配各帧的比特数。

图 1 和图 2 是 Clair 序列在 64 kbps 下缓冲区 BufferFullness 图和 PSNR 波动图;图 3 和图 4 是 Bridge-close 序列 64 kbps 下缓冲区 BufferFullness 图和 PSNR 波动图。图 1 和图 3 显示了新算法的缓冲区 BufferFullness 比 JM8.6 原算法的在 GOP 前三分之一帧下降得慢,这意味着在新算法中,GOP 前三分之一的帧将获得更多的比特。在两个序列的最后

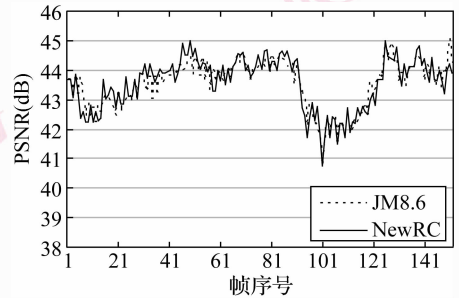


图 2 Clair 序列 PSNR 波动图  
Fig.2 PSNR curve of Clair sequence

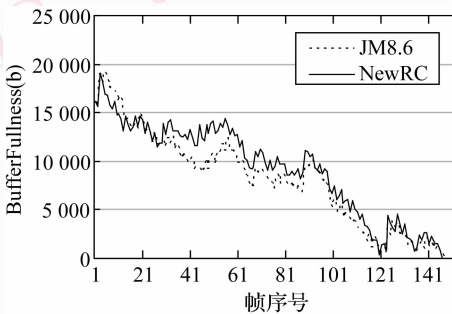


图 1 Clair 序列缓冲区 BufferFullness 图  
Fig.1 BufferFullness result of clair sequence

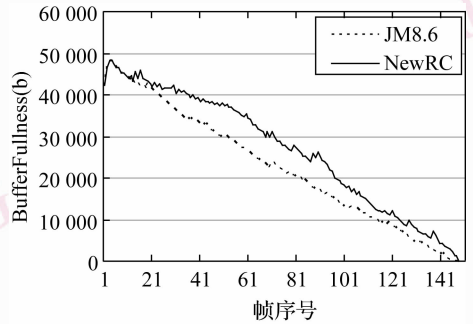


图 3 Bridge-close 序列缓冲区 BufferFullness 图  
Fig.3 Buffer Fullness result of bridge-close sequence

几帧进行编码时,由于编码产生的比特数过少,两种算法都出现了实际缓冲区为空的现象,这时缓冲区的 BufferFullness 为 0。图 2 和图 4 显示了新算法产

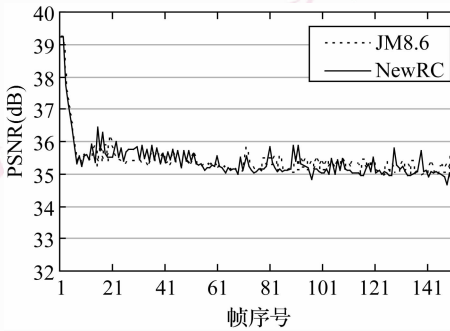


图 4 Bridge-close 序列 PSNR 波动图

Fig. 4 PSNR curve of bridge-close sequence

生的 PSNR 波动要比 JM8.6 原算法的略小,这使得图像序列的连续性得到了提高,有助于增加图像序列的主观质量。

## 7 结 论

本文首先充分利用帧复杂度和目标缓冲区优化了帧层的比特分配,然后验证了在预测帧层的 MAD 方面,卡尔曼滤波器比线性模型有着更高的准确性,并依此提出了一种新的帧层码率控制算法。该算法在码率控制精度,图像总体恢复质量等方面都比 JM8.6 原算法表现优秀。

## 参考文献 (References)

- 1 ISO/IEC 14996 - 10, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification [S].
- 2 Li Zheng-guo, Pan Feng, Lim Keng-pang, *et al.* Adaptive base unit layer rate control for JVT, JVT-G012 [A]. In: Proceedings of 7th Meeting, Pattay II [C], Thailand, 2003.
- 3 Chiang Ti-hao, Zhang Ya-qin. A new rate control scheme using quadratic rate distortion model [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1997, 7(1): 246-250.
- 4 Jiang Min-qiang, Ling Nam. Low-delay rate control for real-time H. 264/AVC video coding [J]. IEEE Transactions on Multimedia, 2006, 8(3): 467-477.
- 5 Kuo C M, Chung S C, Shih P Y. Kalman filtering based rate-constrained motion estimation for very low bit rate video coding [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2006, 16(1): 3-18.
- 6 Cheng Jie, Zhang Wen-jun, Yang Xiao-kang, *et al.* Estimation of the number of header bits using kalman filter and its application in frame layer rate control for H. 264 [J]. Journal of Image and Graphics, 2006, 11(11): 1640-1641. [陈杰, 张文军, 杨小康等. 基于卡尔曼滤波的 H. 264 头信息量估计及其在帧级码率控制中的应用 [J]. 中国图象图形学报, 2006, 11(11): 1640-1644.]
- 7 Zhu Guang-xi, Li Xiao, Lu Dong-xin, *et al.* Research about rate control on low bit rate [J]. Computer and Digital Engineering, 2006, 33(6): 1-5. [朱光喜, 李霄, 芦东昕等. 低码率下码率控制策略的研究 [J]. 计算机与数字工程, 2006, 33(6): 1-5.]