

# 一种基于运动复杂度的 H.264 JVT-H017 码率控制跳帧算法

张 英,魏忠义,石 娟

(西安工程大学 电子信息学院, 陕西 西安 710048)

**摘 要:** 针对 JVT-H017 码率控制算法中跳帧策略没有很好地考虑图像的运动复杂程度, 仍采用类似于 MPEG-4 Q2 中的简单方法的问题, 提出了一种基于运动复杂度的跳帧算法。此方法充分利用了序列的运动程度, 较好地缓解了由于物体剧烈运动所带来的缓存上溢现象, 优化了图像的质量。

**关键词:** JVT-H017; 码率控制; 运动复杂度; 跳帧

中图分类号: TN919.81

文献标识码: A

文章编号: 1674-7720(2010)21-0031-03

## A frame skipping strategy in H.264 JVT-H017 rate control scheme based on the motion complexity

ZHANG Ying, WEI Zhong Yi, SHI Juan

(Electronic Information School, Xi'an Polytechnic University, Xi'an 710048, China)

**Abstract:** In order to emend the deficiency of the frame skipping strategy used in JVT-H017 rate control method, in which a number of pictures may be skipped by using the method similar to MPEG-4 Q2 and the motion complexity of the frame to be not considered. This paper present a frame skipping strategy based on the JVT-H017 rate control method. The motion complexity of the frame as the criterion is used in this method. It can prevent the overflowing of the buffer caused by the strong movement of object and improve the quality of the picture.

**Key words:** JVT-H017; rate control; motion complexity; frame skip

H.264<sup>[1]</sup>是最新的国际视频编码标准,它不仅使视频压缩比以往标准有明显提高,而且具有良好的网络亲和性,特别是对 IP 互联网、无线移动网等易误码、易阻塞、QoS 不易保证的网络视频传输性能有明显的改善。因此, H.264 已成为现今图像压缩研究的一个热点。同时,码率控制作为视频压缩算法的一个重要方面,也引起了研究者的很大重视。

在 H.264 码率控制经典算法里, JVT-H017<sup>[2]</sup>因对实际码率和 PSNR 的控制更加准确,并能更好地逼近限制的码率,而与 JVT-G012<sup>[3]</sup>一起成为目前最常用的 H.264 码率控制算法。但 JVT-H017 算法仍然存在明显的不足<sup>[4]</sup>: 同时引用率失真(R-D)模型和 MAD 线性模型会增加编码的计算量; 算法中图像组 GOP (Group of Pictures) 码率的分配方法和 HRD 机制限制了前后质量平缓度的进一步提高; 编码后的跳帧策略

仍然采用类似于 MPEG-4 Q2 中的简单方法,严重影响了运动剧烈图像的质量。本文在介绍 JVT-H017 码率控制算法的基础上,提出了一种利用运动复杂度作为跳帧判别的算法,合理地分配缓存空间给运动剧烈的帧,较好地保留了图像的细节,进一步提高了图像的质量。

### 1 JVT-H017 码率控制算法

JVT-H017 码率控制策略包含 3 个层次: GOP 层、帧层和基本单元层。其中基本单元层指的是同一帧中一组连续的宏块,每个基本单元层至少包含一个宏块。

#### 1.1 GOP 层码率控制

GOP 层码率控制主要任务是计算当前 GOP 中剩余图像帧的比特数,初始化即时解码刷新(IDR)和第一个存储帧的量化参数。当第  $i$  个 GOP 的第  $j$  帧被编码时,当前 GOP 中剩余帧的比特数计算如下:

$$B_i(j) = \begin{cases} \frac{R_i(j)}{f} \times N_i - V_i(j) & j=1 \\ B_i(j-1) + \frac{R_i(j) - R_i(j-1)}{f} \times (N_i - j + 1) - b_i(j-1) & j=2, 3, \dots, N_i \end{cases} \quad (1)$$

式中,  $f$  是预定义的编码帧率;  $N_i$  是第  $i$  个 GOP 的总帧数;  $R_i(j)$ 、 $V_i(j)$  分别是当第  $i$  个 GOP 的第  $j$  帧被编码时, 当前可用的比特率和虚拟缓冲(buffer)的占用率;  $b_i(j-1)$  是第  $j-1$  帧的比特数。

当前帧为 IDR 帧或第  $i$  个 GOP 的第 1 个存储帧时, 则需初始量化参数  $QP_i(1)$ :

$$QP_i(1) = \max \left\{ QP_{i-1}(1) - 2, \min \left[ QP_{i-1}(1) + 2, \frac{\text{Sum}PQP(i-1)}{N_p(i-1)} - \min \left\{ 2, \frac{N_{i-1}}{15} \right\} \right] \right\} \quad (2)$$

式中,  $N_p(i-1)$  是第  $i-1$  个 GOP 的已存储帧总数,  $\text{Sum}PQP(i-1)$  是所有已存储帧的平均帧量化参数和。

## 1.2 帧层码率控制

帧层的码率控制包括编码前和编码后两个阶段。

### 1.2.1 编码前阶段

编码阶段是计算每一帧的量化参数。对于存储帧和非存储帧, 方法是不一样的。本文介绍的是只有存储帧的情况。这一过程分两个步骤完成:

#### (1) 确定每个帧的目标码率

##### ① 确定当前 GOP 中每个存储帧的目标码率

每一个存储帧的目标缓冲由第一个 IDR 或存储帧的编码比特和平均复杂度得到。在编码完第  $i$  个 GOP 的第 1 个存储帧后, 目标缓冲的初始值为:

$$S_i(2) = V_i(2) \quad (3)$$

后续存储帧的目标缓冲计算为:

$$S_i(j+1) = S_i(j) - \frac{S_i(2)}{N_p(i)-1} + \frac{\overline{W}_{p,d}(j) \times (L+1) \times R_i(j)}{f \times (\overline{W}_{p,d}(j) + \overline{W}_{b,d}(j) \times L)} - \frac{R_i(j)}{f} \quad (4)$$

式中,  $\overline{W}_{b,d}(j)$  表示 B 帧的平均复杂度,  $\overline{W}_{p,d}(j)$  表示 P 帧的平均复杂度,  $L$  为 B 帧的个数。

#### ② 计算当前帧的目标比特

第  $i$  个 GOP 中第  $j$  帧的目标比特的分配是根据目标缓存、帧率、可用信道带宽以及实际的缓存占用率进行计算的:

$$T_i(j) = \beta \times \hat{T}_i(j) + (1-\beta) \times \tilde{T}_i(j) \quad (5)$$

其中:

$$\tilde{T}_i(j) = \frac{R_i(j)}{f} + \gamma \times (S_i(j) - V_i(j)) \quad (6)$$

$$\hat{T}_i(j) = \frac{W_{p,d}(j-1) \times B_i(j)}{W_{p,d}(j-1) \times N_{p,r} + W_{b,d}(j-1) \times N_{b,r}} \quad (7)$$

为了符合假定解码器 HRD 的要求, 目标比特的定义为:

$$T_i(j) = \min\{U_i(j), T_i(j)\} \quad (8)$$

$$T_i(j) = \max\{Z_i(j), T_i(j)\}$$

式中,  $Z_i(j)$ 、 $U_i(j)$  分别是 HRD 的上下限。在没有 B 帧的情况下,  $\beta=0.5$ ; 否则,  $\beta=0.9$ 。

### (2) 计算量化参数和执行率失真优化(RDO)

首先采用式(9)计算与目标比特关联的量化步长, 相应的量化参数  $QP_i(j)$  由量化步长和高级视频编码(AVC)的量化参数计算得到:

$$T_i(j) = c_1 \times \frac{\tilde{\sigma}(j)}{Q_{\text{step},d}(j)} + c_2 \times \frac{\tilde{\sigma}(j)}{Q_{\text{step},f}(j)} - m_{h,i}(j) \quad (9)$$

式中,  $m_{h,i}(j)$  是所有头信息和运动矢量的编码比特和,  $c_1$ 、 $c_2$  是系数。为了保证连续帧之间的平滑性, 量化参数  $QP_i(j)$  调整为:

$$QP_i(j) = \min\{QP_i(j-L-1)+2, \max\{QP_i(j-L-1)-2, QP_i(j)\}\} \quad (10)$$

### 1.2.2 编码后阶段

编码一帧以后, 系数  $a_1$ 、 $a_2$  及  $c_1$ 、 $c_2$  采用类似 MPEG-4 Q2 部分的线性回归模型进行更新。在此过程中, 实际编码产生的比特加入到缓存中。为保证更新后的缓存占用率不要过高, 采用类似 MPEG-4 Q2 部分的跳帧策略进行调整。

### 1.3 基本单元码率控制

基本单元码率控制和帧层码率控制的方法类似, 为每个基本单元预测 MAD, 计算目标比特, 用二次 R-D 模型计算基本单元层的  $QP$  时需考虑: (1) 对当前帧的第一个基本单元,  $QP$  需设为前一帧所有基本单元量化参数的平均值; (2) 当前帧中还有未编码的基本单元, 而分配给它们的剩余比特数却小于零。这时, 当前基本单元的  $QP$  应相应地增大; (3) 其他情况下, 通过二次 R-D 模型来获得  $QP$ 。

## 2 改进的跳帧算法

在实际的编码中, 如果每个 GOP 都超过了预先分配给它的比特数, 这时缓冲就会溢出, 从而导致跳帧现象, 使得图像的质量下降<sup>[4]</sup>。JVT-H017 码率控制算法采用的跳帧方法仍是 MPEG-4 Q2 中的简单方法, 即只是简单地设置缓存区占有度上限达到 80% 时, 就执行跳帧。(同样在以往的跳帧控制算法中<sup>[5]</sup>, 一般都只是将缓存的上限定为 80% 来作为跳帧的依据, 当缓存容量超过 80% 时, 将此帧定为跳帧)。相反, 当缓存容量低于 25% 时, 也将此帧定为跳帧, 但是这样处理没有充分考虑到简单的跳帧会影响运动剧烈的图像的细节表现, 导致图像质量严重下降。于是参考文献[6]根据高运动复杂度对应高比特数、低运动复杂度对应低比特数这一比特分配法则, 利用过滤掉运动平稳的帧而保存运动剧烈的帧, 并为运动剧烈的帧分配更多缓存空间的思想, 提出了用前一帧预测得到的比特数与前面已编码帧的实际比特数的平均值的比值作为衡量跳帧的标准。但参考文献[5]并未考虑到出现比特数较多和比特数较少的帧交错的帧序列这一情况。若前一帧比特数较少, 当前帧比特数较多, 则得到的比值偏小; 反之, 得到的比值偏大。可见用此方法作为跳帧的判断方法, 准确率并不高。所以本文算法从图像复杂度算法本身入手, 并考虑到相邻帧之间具有很

强的相关性,重新定义一个跳帧判别方法,力求得到的跳帧判别方法能较准确地优化图像质量。

为了更好地适应 AVS 编码器的实时应用,减少 B 帧延时带来的影响,本文只考虑有 I 帧、P 帧的图像格式。由于 I 帧编码需要较多的比特数,它将占据多个帧间间隔传输,因此,本文为了减小缓冲延时,紧邻 I 帧的前两个 P 帧仍按原算法进行跳帧。即从第 3 个 P 帧开始定义当前 P 帧的运动复杂度因子  $c_i$ :

$$c_i = \frac{a_1 \times W_{p,i}(j-1) - a_2 \times W_{p,i}(j-2)}{\frac{1}{j-1} \times \sum_{k=3}^{j-1} W_{p,i}(k)} \quad k=3, 4, \dots, j-1 \quad (11)$$

式中,  $W_{p,i}(j-1)$  表示当前帧的前 1 个 P 帧的实际复杂度,  $W_{p,i}(j-2)$  表示当前帧前面第 2 个 P 帧的实际复杂度,  $W_{p,i}(k)$  为前面已编码帧的实际复杂度,  $W_{p,i}(j) = b_{p,i}(j) \times QP_{p,i}(j)$ ,  $a_1$ 、 $a_2$  是经验值。

GOP 结构为 IPPP, 即第一帧为 I 帧, 后面的为 P 帧。设定缓存容量的百分比为  $w_1$ 、 $w_2$  ( $w_1 < w_2$ ), 同时设定图像的运动复杂度为  $a$ 、 $b$  ( $a < b$ ),  $a$ 、 $b$  取固定值,  $w_i$  是当前帧的复杂度。如果  $c_i < a$ , 当  $w_i > w_1$  时, 就将此帧判为跳帧; 如果  $a < c_i < b$ , 当  $w_i > w_2$  时, 也判定为跳帧; 当  $c_i > b$  时, 则按照以前的跳帧算法判断; 当缓存的容量超过 80% 以后, 无论  $c_i$  的值为多少都将此帧判定为跳帧。

### 3 实验结果与分析

为了对算法进行分析, 在 JM8.6 上对本跳帧算法进行仿真。实验中分别使用 QCIF 序列 foreman 和 CIF 序列 mobile, 共编码 100 帧, GOP 的结构是 IPPP... IPPP... 帧内编码刷新周期为 20 帧, 第一帧的量化步长为 28,  $a=0.6$ ,  $b=0.8$ ,  $w_1=60\%$ ,  $w_2=70\%$ ,  $a_1=3/2$ ,  $a_2=1/2$ 。在 JM8.6 现有的码率控制算法的基础上, 分别对跳帧策略与参考文献[2]和参考文献[5]中的跳帧策略进行测试, 截取其中的 20~40 帧进行显示。比较 3 种算法的 PSNR。在相同的条件下, 本文提出的跳帧算法相对其他两个跳帧算法有所改进, 更加准确地做到了跳过平稳运动的图像, 保留运动剧烈的图像这准则。从表 1、表 2 可以看出, 本文提出的跳帧算法, 使信噪比有所提高, 优化了图像的质量。

表 1 针对 foreman 本算法与其他两种算法的信噪比的比较

foreman	PSNR_Y/dB	PSNR_Cr/dB	PSNR_Cb/dB
JVT-H017	35.83	37.61	37.65
参考文献[5]	35.92	37.63	37.75
文本算法	36.05	37.71	37.84

表 2 针对 mobile 本算法与其他两种算法的信噪比的比较

mobile	PSNR_Y/dB	PSNR_Cr/dB	PSNR_Cb/dB
JVT-H017	35.19	36.91	36.88
参考文献[5]	35.21	36.93	36.90
文本算法	35.38	36.99	36.94

本文针对 JVT-H017 码率控制算法中跳帧策略仍然采用类似 MPEG-4 Q2 的简单方法这一缺点, 提出的跳帧算法合理地利用了图像运动复杂度这一尺度来作为判别标准, 与参考文献[2]的(JVT-H017)相比, 使得图像信噪比有所提升, 优化了图像质量。但是, 本算法并没有考虑到出现场景切换和 B 帧存在的图像序列的跳帧问题, 所以将来的工作将是在此算法的基础上, 提出一种更好的优化算法, 使其能提高不同图像序列的图像质量。

### 参考文献

- [1] 余兆明, 查日勇, 黄磊, 等. 图像编码标准 H.264 技术[M]. 北京: 人民邮电出版社, 2005.
- [2] MA Si Wei, LI Zheng Guo, WU Feng, et al. JVT-H017-2003 proposed draft of adaptive rate control[S]. 2003.
- [3] MA Si Wei, LI Zheng Guo, WU Feng, et al. JVT-G012-2003 proposed draft of adaptive rate control[S]. 2003.
- [4] 李记苗, 刘彦明, 高峰. H.264 码率控制及其改进策略[J]. 计算机与数学工程, 2006(6): 52-54.
- [5] 郑从卓, 李均利, 陈刚, 等. AVS 实时编码的 VBR 码率控制[J]. 计算机工程, 2009(35): 240-242.
- [6] 周骏华, 石旭刚, 宋玮. 一种基于运动复杂度的新的 H.264 码率控制算法[J]. 浙江工业大学学报, 2006(6): 672-675.

(收稿日期: 2010-06-07)

### 作者简介:

张英, 女, 1984 年生, 硕士研究生, 主要研究方向: 无线通信与个人通信基本理论研究。

魏忠义, 男, 1946 年生, 研究生导师, 主要研究方向: 无线通信与个人通信。

石娟, 女, 1984 年生, 硕士研究生, 主要研究方向: 无线通信与个人通信基本理论研究。