

H.264 视频标准中码率控制算法的改进

程 波, 陈真勇, 熊 璋

CHENG Bo, CHEN Zhen-yong, XIONG Zhang

北京航空航天大学 计算机学院, 北京 100083

School of Computer Science and Technology, Beijing University of Aeronautics and Astronautics, Beijing 100083, China

E-mail: cb@cse.buaa.edu.cn

CHENG Bo, CHEN Zhen-yong, XIONG Zhang. Improvement of rate control algorithm in H.264 standard. Computer Engineering and Applications, 2007, 43(24): 94-96.

Abstract: Rate control is a very important part of the video compressing code. In many applications, video sequence is required to transmit in constant rate channel, but the bit rate of the compressed video sequence is variable generally. It is necessary to solve this problem using rate control technology. Therefore, this paper discusses the rate control technology in JVT-G012 in detail, analyzes the disadvantage of this algorithm, gives the corresponding improvement and verifies via the experiments.

Key words: video coding; H.264; rate control; JVT-G012

摘 要: 码率控制是视频压缩编码技术的一个重要部分。在许多应用中, 视频序列都要求在恒定比特率的信道中传输, 而压缩后的视频序列中的数据量通常是变化的, 因此, 需要通过码率控制技术来解决。对 JVT-G012 提案中的码率控制技术进行了深入详细的探讨, 分析了此算法的不足, 在此基础上进行了相应的改进, 并通过实验进行了验证。

关键词: 视频编码; H.264; 码率控制; JVT-G012

文章编号: 1002-8331(2007)24-0094-03 **文献标识码:** A **中图分类号:** TN919.8

1 引言

JVT (Joint Video Team, 视频联合工作组) 由 ITU-T 的 VCEG (Video Coding Experts Group) 和 ISO/EIC 的 MPEG (Motion Picture Experts Group) 联合成立于 2001 年 12 月。H.264 标准是 JVT 组织制定的新建议, 在 2003 年 5 月被正式确定为国际标准^[1]。H.264 标准集中了以往标准的优点。与 H.263+ 或者 MPEG-4 简单类相比, H.264 在大多数码率下最多可以节省 50% 的码率。H.264 能持续提供较高的视频质量, 在实时通信应用中能很好地工作在低延时模式下, 同时也能非常好地工作在视频存储等没有延迟限制的模式下^[2]。H.264 在压缩率和图像质量方面、网络适应性方面、抗丢包和抗误码方面都有很好的表现, 因此, H.264 逐渐成为多媒体通信中首选的视频编码标准。

视频编码后的信息量往往是变化的, 为了使编码后的码流速率能与通信信道的速率相匹配, 在编码器和信道之间设计一个缓冲区非常必要。缓冲区容量越大, 码流就越好控制, 能更好地匹配信道, 但同时也给解码端带来更大的时延。要做到时延小, 则要求缓冲区的容量减小, 这便形成一对矛盾。码率控制的目的就是为了解决这个矛盾, 在保证视频传输过程失真尽可能小的情况下, 满足时延的要求并使得缓冲区既不发生上溢也不发生下溢。

目前, 针对码率控制已经有相当多的研究。He 等^[3]提出线性率失真函数, 通过变换量化后零值在变换系数中的比例 (认

为这对编码码率的影响最大) 来选取量化参数, 可避免蛋鸡悖论; 陈等^[4]提出联合编码模式选择、信源的码率控制算法; Xue^[5]等提出基于块活动性和缓冲状态的算法; Ma 等^[6]提出结合 HRD 的控制算法, 并被 H.264 采用。有些算法采用多个量化参数进行量化, 然后选取最优的结果。此外, 也有一些算法采用各种模型来计算量化参数, 如模糊逻辑、动态规划、二次模型和指数模型等。

本文针对 JVT-G012 提案中的码率控制技术进行探讨, 对该提案中所采用的算法运行效率不高的原因进行分析, 并进行改进。

2 视频码率控制系统

视频码率控制系统可以用图 1^[7]来描述。原始视频输入编码器进行编码, 当第 i 帧编码结束后, 相应的压缩码流 R_i 被输入到缓存器, 用户网络接口选择适当的信道码率, 然后将经编码器平滑后的码流送到网络。压缩后的视频码流经过网络到达解码缓冲器, 经过 L 帧的系统延时后, 第 i 帧的压缩流到达解码器进行还原。

视频编码时, 用户网络接口会先选择信道码率。信道一般可以分为两大类: 变化速率信道 (VBR) 和恒定速率信道 (CBR)。VBR 常用于包交换网络, 如 ATM, 而 CBR 常用于电路交换网。研究表明^[8], 经包交换网络传输的 VBR 码流和经电路

基金项目: 航空支撑科技基金。

作者简介: 程波 (1981-), 男, 硕士生, 研究方向: 图像通信、多媒体通信; 熊璋 (1956-), 男, 教授, 博士生导师, 研究方向: 多媒体技术和分布式信息处理。

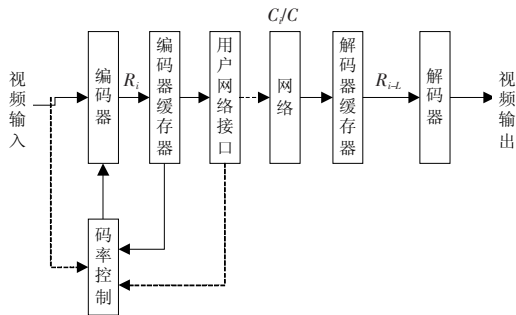


图1 码率控制流程

交换网络传输的 CBR 码流相比, VBR 码流具有更好的图像质量、更高的 SMG 和更短的系统延时, 但是前提是必须有良好的视频信源和信道码率控制来满足包交换网络对 VBR 的约束。CBR 码流输出速率是恒定的, 一般只需要根据给定的信道码率进行码率控制, 算法比较简单。如果要想把 VBR 码流转化成 CBR 码流, 需要很大的缓冲器, 这样很难将系统延时控制在较小的范围内。视频传输中的码率控制即是根据网络吞吐量的约束条件, 在视频序列的编码单位, 如图像组(GOP)、帧、条(Slice)、宏块和块之间进行适当的码率分配, 以确保解码端获得较好的图像质量和较短的系统延时。

3 JVT-G012 码率控制算法

JVT-G012 是 JVT 于 2003 年 3 月提出的关于码率控制算法的提案。该提案中的码率控制算法采用了分层控制的方法^[9], 共分为三层: GOP(图像组)层码率控制, 帧层码率控制和基本单元(BU)层码率控制。GOP 层的目标比特采用固定比特; 帧层的目标比特由网络带宽、缓存大小、缓存占用量以及剩余比特共同决定; 基本单元层的目标比特基于平均绝对差值(MAD) 进行分配。

3.1 帧层码率控制

帧层码率控制分为两个阶段:

- (1) 为每一个 P 帧计算一个目标比特数;
- (2) 循环地更新率失真模型的各个参数并进行跳帧控制。

假设只有一组图像(GOP), 并且图像序列的编码顺序是第一帧为 I 帧, 为连续的 P 帧。N 表示该图像组中帧的总数目, $B_c(j)$ 表示编码第 j 帧图像后缓冲器实际的占用量, $A(j)$ 表示对第 j 帧图像编码后缓冲器实际产生的比特数, B_s 表示缓冲器大小, $Tbl(j)$ 为第 j 帧编码完成后缓冲器的目标值, R_r 为缓冲器的剩余比特数。那么, 编码第 j+1 帧图像后缓冲器的实际占用量 $B_c(j+1)$ 可以表示如下^[10]:

$$B_c(j+1) = B_c(j) + A(j+1) - \frac{R_b}{R_f} \quad (1)$$

$$B_c(1) = \frac{B_s}{8} \quad (2)$$

其中 R_b 为信道的目标比特率, R_f 为帧率, 即每秒传输的帧数。假设第一个 P 帧的量化参数等于 GOP 的量化参数, 可以得到:

$$Tbl(2) = B_c(2) \quad (3)$$

其它 P 帧编码后缓冲器的目标值可以定义如下:

$$Tbl(j+1) = Tbl(j) - \frac{Tbl(j) - B_s/8}{N-1} \quad (4)$$

因此, 考虑缓冲器占用量而得到的目标比特数可以表示为:

$$T_{buf} = \frac{R_b}{R_f} + [Tbl(j) - B_c(j)] \quad (5)$$

考虑剩余比特数而得到的目标比特数可以表示为:

$$T_r = \frac{R_r}{N_r} \quad (6)$$

其中, N_r 表示 GOP 中的剩余帧数。最后的目标比特可以通过 T_{buf} 和 T_r 的加权和得到:

$$T = \beta \times T_r + (1 - \beta) \times T_{buf} \quad (7)$$

其中 β 取值为 0.5。

3.2 基本单元层码率控制

JVT-G012 中提出了基本单元的概念。基本单元可以是宏块、条、场或者帧。一般, 选择的基本单元越大, 获得的 PSNR 越大, 但是比特输出的波动也会越大; 选择的基本单元越小, 比特输出相对平稳, 但是获得的 PSNR 将会变小。

从图 2 可以看出^[9], 基本单元层码流控制算法的主要步骤为预测当前基本单元的 MAD, 分配当前基本单元的目标比特和计算量化参数。

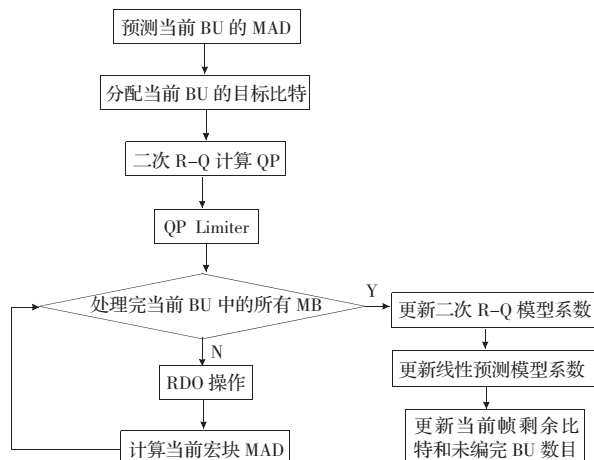


图2 基本单元层码率控制流程

JVT-G012 中对当前基本单元 MAD 值的计算是通过一个线性模型来进行预测的。具体做法是根据前一帧相应位置的基本单元预测当前基本单元的 MAD。假设 MAD_{cb} 和 MAD_{pb} 分别表示当前基本单元 MAD 预测值和前一帧相同位置基本单元 MAD 的实际值。线性预测模型如下:

$$MAD_{cb} = a_1 \times MAD_{pb} + a_2 \quad (8)$$

其中 a_1 和 a_2 是预测参数, 初始值分别设为 1 和 0, 每次编码完一个基本单元后要对这两个预测参数进行更新。

当前基本单元目标比特是根据当前基本单元的 MAD 预测值进行分配的。假设 T_i 为当前基本单元的目标比特, T_r 为当前帧中剩余的比特, N 为当前帧中基本单元数目, 则当前基本单元的目标比特可以定义为:

$$T_i = T_r \times \frac{MAD_i^2}{\sum_{j=i}^{N-1} (MAD_j^2)} \quad (9)$$

另外, 基本单元层的量化参数 QP 通过解下面的方程得到:

$$T_i = MAD \times (C_1 \times \frac{1}{Q} + C_2 \times \frac{1}{Q^2}) \quad (10)$$

4 码率控制算法的改进

4.1 帧层码率控制

帧层码率的分配主要取决于两个因素:一个是缓冲器的占用量,另外一个为是剩余比特。

计算目标缓冲器的占用量 B_t 的公式如下:

$$B_t = B_c - \frac{B_c - \frac{B_s}{8}}{N_r} \quad (11)$$

式中 B_c 是当前缓冲器占用量, B_s 是缓冲器大小, N_r 是剩余帧数, B_t 是当前帧被编码后缓冲器的目标比特, B_c 的初始值为 $B_s/8$ 。

在此定义一个变量 B'_{frame} :

$$B'_{frame} = \frac{R_B}{R_F} - 0.7 \times (B_c - B_t) \quad (12)$$

式中 R_B 是目标比特率, R_F 是帧率, 常量 0.7 是从大量实验中总结出来的实验数据。定义另外一个变量 B''_{frame} :

$$B''_{frame} = \begin{cases} 0.75 \times a \times \frac{R}{N_r} + 0.25 \times R_l & H_1 < a < H_2 \\ \sqrt{a} \times \frac{R}{N_r} & \text{其他} \end{cases} \quad (13)$$

式中 R 是剩余比特, R_l 是最后一帧的比特数。经由实验, H_1 、 H_2 一般分别取 0.9 和 1.2。 a 的定义为:

$$a = \frac{MAD_{pre}}{MAD_{avg}} \quad (14)$$

定义好了 B'_{frame} 和 B''_{frame} , 就可以得到帧层码率的计算公式:

$$B_{frame} = \beta \times B'_{frame} + (1 - \beta) \times B''_{frame} \quad (15)$$

式中 β 取常量值 0.1。

帧的量化参数 QP 可以通过下面二次模型计算:

$$W = m_1 \times \frac{\sigma}{Q} + m_2 \times \frac{\sigma^2}{Q} + W_{hdr} \quad (16)$$

其中 m_1 和 m_2 是一次和二次系数, W 是总比特数, 用来编码一帧, σ 可以由 MAD 简化代替, Q 是量化参数, W_{hdr} 是帧头比特。

4.2 基本单元层的码率控制算法

在改进的帧层码率控制中, 仍然沿用线性模型去预测帧的 MAD 值。通过大量的实验, 发现 MAD 值具有很强的相关性。基本单元码率控制中 MAD 值的预测最为关键, 它既影响目标比特的分配, 又是计算 QP 的一个输入。

JVT-G012 提案中的线性预测模型在每次编码完一个基本单元后要模型中的参数 a_1 和 a_2 进行刷新。提案中对 MAD 的预测是通过线性拟合得到的, 这样预测的精度由线性拟合的输入点数决定, 一般要输入坐标平面上 20 个以上的点进行一次拟合, 一次拟合完成后, 去掉与拟合直线误差较大的点进行第二次拟合。可以看出这种处理过程运算开销非常大。

在实际中, MAD 值之间的时间相关性和空间相关性都是非常大的。因此, 基于这种相关性, 本文提出一种新的模型来对 MAD 值进行线性预测, 避免了线性拟合。当前基本单元的 MAD 值, 跟前一帧相同位置的基本单元的 MAD 值、前一帧相同位置的基本单元的相邻基本单元的 MAD 值以及当前基本单元的相邻基本单元的 MAD 值具有较强的线性关系。假设一帧图像的基本单元有 M 行、 N 列, 则一帧图像的总基本单元数就有 $M \times N$ 个, 则新的预测模型公式如下:

$$MAD_{C(i,j)} = (1 - 12\beta) \times MAD_{P(i,j)} + \sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} \beta_{(m,n)} \times MAD_{P(m,n)} + \sum_{n=j-1}^{j+1} \beta_n \times MAD_{C(i-1,n)} + \beta \times MAD_{C(i,j-1)} \quad (17)$$

式中参数取值如下:

$$\begin{cases} \beta_{(m,n)} = 0 & m=0 \text{ 或者 } n=0 \\ \beta_{(m,n)} = 0.02 & m \neq 0 \text{ 或者 } n \neq 0 \\ \beta_n = 0 & n=0 \\ \beta_n = 0.02 & n \neq 0 \\ \beta = 0.02 \end{cases}$$

其中, $MAD_{C(i,j)}$ 为当前基本单元的 MAD 值, $MAD_{P(i,j)}$ 为前一帧相同位置基本单元的 MAD 值, $MAD_{C(i,j-1)}$ 为当前基本单元左边的基本单元的 MAD 值, 依此类推。

在上面的预测模型公式中, 参数都为常量, 这样大大简化了 MAD 值预测的算法复杂度。

5 实验结果与分析

实验采用 JM10.1 的参考源码。将改进的算法应用到 JM10.1 软件模型上, 并与软件的原始算法进行了比较。选取的测试对象是 Foreman 序列和 Trevor 序列, Foreman 序列的特点是镜头运动较大, Trevor 序列的特点则是场景的变化较大。测试的其它条件为: 每帧图像大小为 QCIF (176×144); 序列中 I 帧量化级数为 31, P 帧量化级数为 34。

MAD 值预测的实验结果如表 1 所示, 可以看出改进后的算法和原算法的误差平方和的差值与编码帧数目成正比, 说明改进后的算法的预测精确度普遍优于原算法。

表 1 改进后算法与原算法的预测误差

| 测试序列 | 编码数 目/帧 | 原算法预测 | 改进算法预测 | 误差平方 和差值 Δ |
|---------|------------|--------|--------|----------------------|
| | | 误差平方和 | 误差平方和 | |
| Foreman | 10 | 1.768 | 1.634 | 0.134 |
| | 50 | 8.412 | 7.131 | 1.281 |
| | 100 | 38.458 | 36.127 | 2.331 |
| Trevor | 10 | 1.984 | 1.156 | 0.828 |
| | 50 | 15.775 | 13.026 | 2.749 |
| | 100 | 32.538 | 28.485 | 4.053 |

为了验证每帧的编码效率是否真的有所提高, 在目标比特率为 64 Kb/s、48 Kb/s、32 Kb/s、20 Kb/s 时候采用 IPPP 序列格式条件下, 对 JM10.1 算法和改进算法分别进行测试。因为不同的码率控制机制产生的实际比特不同, 为了公平的比较, 定义:

$$\text{平均 PSNR} = 10 \log_{10} \times \frac{PSNR}{B} \quad (18)$$

其中 B 是每一帧实际产生的比特数目。

表 2 全面比较了两种算法的性能差异。对于不同的测试序列和目标比特率, 改进后的算法平均 PSNR 高于 JM10.1, 并且相应的比特率更接近于目标比特率。这就证明了改进后的算法的精度更高并且视频质量更好。

6 结论

本文在研究现有视频编码标准 H.264 中的码率控制算法的基础上, 针对 JVT-G012 提案中的码率控制算法进行了改进, 主要贡献体现在以下几个方面: 详细讨论了视频编码标准