

# MPEG2-AVS 转码的码率控制算法

陈学梁

(重庆邮电大学下一代网络应用技术研究所, 重庆 400065)

**摘要:** AVS 是中国自主制定的数字视频编解码标准, 为推动中国数字音视频技术进入实用阶段, 有必要实现 MPEG2-AVS 的转码。该文先直接向 MPEG2-AVS 转码器中加入 AVS 编码标准中的码率控制算法, 并在此码率控制算法的基础上进行改进。通过改进的算法获得的码流比原算法码流平稳, 且更能接近目标比特率。

**关键词:** 码率控制; MAD; 视频编码; AVS; MPEG2

## Rate Control Algorithm in MPEG2-AVS Transcoding

CHEN Xue-liang

(Next Generation Network Application Technology Institute, Chongqing University of Posts and Telecommunications, Chongqing 400065)

**【Abstract】** AVS is digital video encoding and decoding standard constituted by China. In order to accelerate technology of digital audio and video into using phase, it is necessary to produce the transcoder from MPEG-2 to AVS. This paper directly adds the AVS algorithm of rate control to the transcoder, on which improvements are achieved. So the bit rate get closer to the target bit rate, and stream of frame becomes smoother than before.

**【Key words】** rate control; MAD; video coding; AVS; MPEG2

MPEG2-AVS 转码后的 AVS 码流由于图像内容的不同以及编码参数选择的不同, 其码流会产生波动。为了保证输出码流的稳定性, 有必要对输出码流进行控制, 使其维持在一定的范围内, 码率控制是 MPEG2-AVS 转码中不可缺少的一部分。

目前的码率控制算法主要有: 基于缓冲区和信道速率调整量化步长算法, 信息调整量化步长算法以及两者相结合的算法, 但这些算法都存在一些缺点<sup>[1-2]</sup>。本文在 MPEG2-AVS 转码器中使用 AVS 码率控制算法, 并在此算法的基础上进行了改进, 最后对改进算法的性能进行了分析。

本文的算法改进主要是从已编码 MPEG-2 码流中的 DCT 系数中获得编码后的残差系数, 然后根据这些残差系数算出宏块的 MAD 值。通过将该 MAD 值同原有算法中前一帧相同位置宏块 MAD 值对比, 反映宏块的编码复杂度。仿真结果显示: 此方法获得的码流比原算法码流平稳, 且更能接近目标比特率。

### 1 MPEG2-AVS 转码器中的码率控制算法改进

#### 1.1 帧层码率控制算法的改进

P 帧的目标比特数是根据当前图像的编码复杂度和当前缓冲区大小的加权和来分配的, 而当前图像复杂度是根据统计前面已编码帧的复杂度而预测出来的。在运动剧烈或场景切换的情况下, 当前帧的编码复杂度无法根据前面帧的复杂度得到, 因而会使预测值与实际值相差较大, 从而使编码码率与目标码率相差甚远。

由于 MPEG-2 编码后的码流在 MPEG2-AVS 转码器中可得到每帧的编码长度, 编码长度能反映图像的复杂度, 因此这里可用每帧的编码长度来取代原来的编码复杂度, 以便为每个 P 帧分配比特数。

设每个 GOP 中未分配的比特数为  $R$ , 则后续每个 P 帧分配的比特数可以根据下式来计算:

$$R_p = R \times \frac{L_k}{\sum_{i=k}^N L_i} \quad (1)$$

其中,  $k$  为 GOP 中的第  $k$  帧。

通过此方法得到的 P 帧比特数与通过缓冲区获得的 P 帧比特数加权平均, 这样就得到最终的 P 帧预分配比特数。

#### 1.2 宏块层码率控制算法的改进

在宏块层分配比特数时, 当前基本单元的 MAD 值是通过前一帧中相同位置基本单元的 MAD 值, 采用线性公式预测得到的。这是因为通常认为帧与帧之间存在很强的时间相关性, 这样预测得到的值与实际值相差不大。然而当视频序列出现运动剧烈的场景或场景切换时, 帧与帧之间的时间相关性就变得脆弱或完全不相关, 在这种情况下继续使用线性预测方法来获得当前基本单元的 MAD 值, 会使预测得到的 MAD 值同实际的 MAD 值不相符, 进而导致码率控制效果剧减以致失效。因此必须采取有效的方法来避免这种情况发生。由于使用场景切换检测和处理都会带来计算量的增加, 因此对于场景切换检测的几种方法, 它们都不适用于实时性较高的场合, 都有一定的应用局限性。

分析 MPEG-2 码流, 可以知道其中包括 DCT 系数等编码信息。在转码器中 DCT 系数经反 VLC、反量化、反 DCT 变换之后就得到 MPEG-2 中每一宏块编码后的残差, 而 AVS 码率控制算法中的 MAD 值就是 AVS 编码后每一宏块残差的绝对值之和取平均, 因此可以用 MPEG-2 中解码得到的 MAD

**基金项目:** 重庆市科委自然科学基金资助项目(CSTC2005BB2054); 重庆市教委科学技术研究项目(KJ050512, KJ050513, KJ060514); 重庆邮电大学青年教师基金资助项目(A2005-26)

**作者简介:** 陈学梁(1982-), 男, 硕士研究生, 主研方向: 下一代网络(NGN)技术, 3G 通信

**收稿日期:** 2006-11-25 **E-mail:** chenxueliang.master@gmail.com

值来预测码率控制所需要的 MAD 值。

不管采用何种编码标准，都是对复杂度较高的图像分配较多的比特数，复杂度较低的图像分配较少的比特数。因此，从 MPEG-2 中获得的 MAD 值的大小实际上也反映了 AVS 中宏块的编码复杂度。

值得注意的是：MPEG-2 的帧内编码模式是直接对帧内宏块的像素值进行 DCT 变换，因此解码 MPEG-2 码流得到的 DCT 系数是像素值，而不是残差，所以无法得到当前宏块的 MAD 值。为了获得帧内宏块的 MAD 值，并且同 AVS 的帧内编码模式相对应，减小编码模式的不匹配而造成获得的 MAD 值同 AVS 编码后巨大差别，可以对 MPEG-2 中的帧内宏块采取同 AVS 中的帧内 DC 预测一样的预测方法。

假设每个要处理的 8×8 块上面和左边的像素值如图 1 所示。对于第 1 个宏块 AVS 中采用前一帧的平均量化参数(在这里可不用处理)；对于每帧的第 1 行宏块，块 1 和块 2 采用左边 8 个像素点的均值预测，块 3 和块 4 采用上边和左边共 16 个像素点的均值预测。

	A	B	C	D	E	F	G	H
I								
M								
N								
O								
P								
Q								
R								
S								

图 1 8×8 块周围的像素值

参照图 1，具体的算法描述如下：

块 1 和块 2 中的像素预测值：

$$pixel[x][y] = (I + M + N + O + P + Q + R + S + 4) / 8 \quad (2)$$

其中， $x, y \in (0, 7)$

块 3 和块 4 中的像素预测值：

$$pixel[x][y] = (A + B + C + D + E + F + G + H + I + M + N + O + P + Q + R + S + 8) / 16 \quad (3)$$

其中， $x, y \in (0, 7)$ 。

对于每帧的第 1 列宏块，块 1 和块 3 采样上面的 8 个像素点预测，块 2 和块 4 采用上边和左边各 8 个像素点的平均值预测；

块 1 和块 3 中的像素预测值：

$$pixel[x][y] = (I + M + N + O + P + Q + R + S + 4) / 8 \quad (4)$$

其中， $x, y \in (0, 7)$ 。

块 2 和块 4 中的像素预测值：

$$pixel[x][y] = (A + B + C + D + E + F + G + H + I + M + N + O + P + Q + R + S + 8) / 16 \quad (5)$$

其中， $x, y \in (0, 7)$ 。

对于其他宏块，均采用上边和左边共 16 个像素点的平均值预测。

$$pixel[x][y] = (A + B + C + D + E + F + G + H + I + M + N + O + P + Q + R + S + 8) / 16 \quad (6)$$

其中， $x, y \in (0, 7)$ 。

通过以上算法，就可得到 MPEG-2 中帧内帧间编码宏块的 DCT 系数，通过这些 DCT 系数就可获得 MPEG-2 中帧内帧间宏块的 MAD 值。然后就可预测当前编码宏块和未编码宏块的 MAD 值，根据当前宏块的 MAD 值在未编码宏块中所占的比例为其分配比特数，最后利用信源模型计算当前宏块的量化参数，并在编码完成后对 MAD 线性预测和信源模

型的参数进行更新。

## 2 仿真结果及分析

在测试中，先用 MPEG-2 编码器对原始 YUV 视频序列编码，然后用 MPEG2-AVS 转码器实现转换编码。测试序列的长度均为 90 帧，帧率均为 25fps。由于改进后的算法只是针对 P 帧，因此在实验时编码序列中不包含 B 帧，即编码类型为 IPPP...。另外不同测试序列的大小以及复杂度的不同，对测试序列设定的目标码率也不相同。

图 2、图 3 表示 MPEG2-AVS 转码器在加入 AVS 码率控制算法和改进码率控制算法后，CIF 序列的编码比特数及图像客观质量 PSNR 值的对比。目标码率设定为 1 400Kb/s。此测试序列在 36~41 帧时图像运动较为剧烈。在原算法获得的实际比特率是 1 390.33Kb/s，图像亮度平均 PSNR 为 36.93db，而改进算法的实际比特率为 1 400.41Kb/s，图像亮度平均 PSNR 为 36.94db。

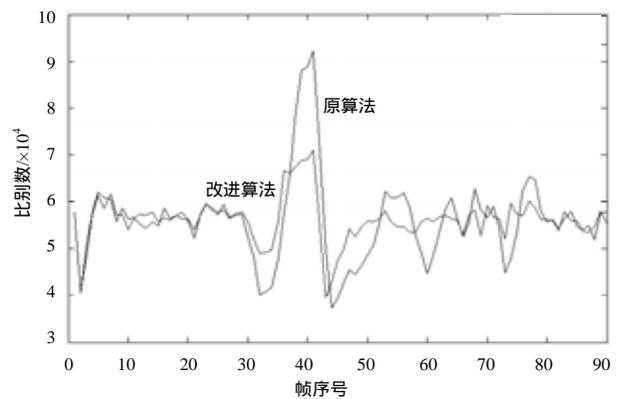


图 2 Football 序列原算法与改进算法的编码比特数对比

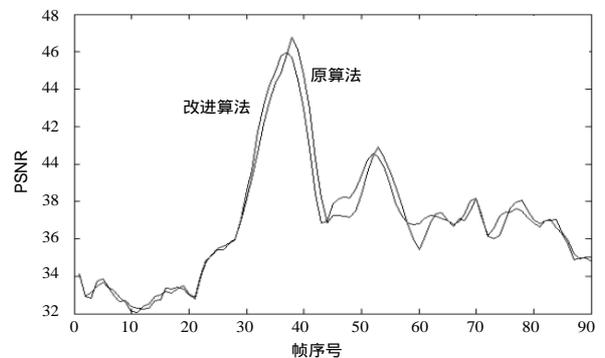


图 3 Football 序列原算法与改进算法的 PSNR 对比

图 4、图 5 表示 MPEG2-AVS 转码器中在加入 AVS 码率控制算法和改进码率控制算法后，ermr 序列(720×576)的图像编码质量(PSNR)和编码比特数的对比。在测试中，目标码率设定为 4Mb/s。原算法的实际比特率为 3 990.93Kb/s，图像亮度平均 PSNR 为 22.16db；改进后算法获得的实际比特率是 4 000.19Kb/s，图像亮度平均 PSNR 为 22.18db。

从仿真结果可以看出，改进后的码率控制算法和原算法相比，码率的准确度要好于原算法。另外，从每一帧的编码比特数对比图来看，改进后的码率控制算法能使每帧的码率趋于平稳，这样缓冲区不容易出现上溢或下溢的情况。这主要是因为改进算法中的 MAD 值更能反映当前宏块的编码复杂度，从而使预测得到的 MAD 值更加准确，准确的 MAD 值通过信源模型能获得更符合实际情况的量化参数，以利于

对码率的控制。

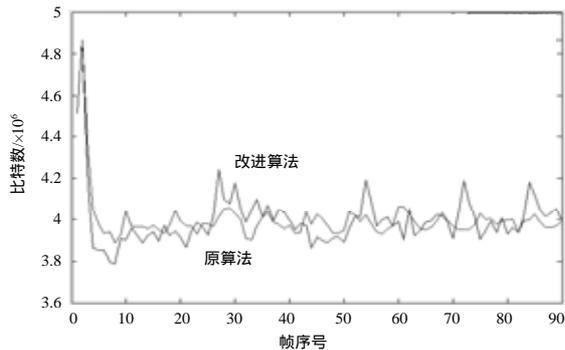


图4 ermr 序列原算法与改进算法的编码比特数对比

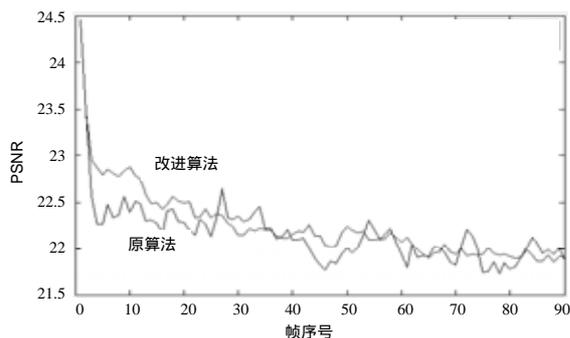


图5 ermr 序列原算法与改进算法的 PSNR 对比

在MPEG2-AVS转码器中,AVS的对每一宏块编码模式的选择是直接来自MPEG-2 中映射过来的,因此AVS编码器中的用于宏块模式选择的率失真优化在转码器中可以被屏蔽掉。这样在码率控制算法中就不会出现蛋鸡相悖理论,即此时的码率控制算法不会改变宏块编码模式,它改变的只是宏块的量化参数,因此图像的平均PSNR值不会有太大的变化。从上面的数据可以看出,测试序列的平均PSNR值在算法改变前后

(上接第 245 页)

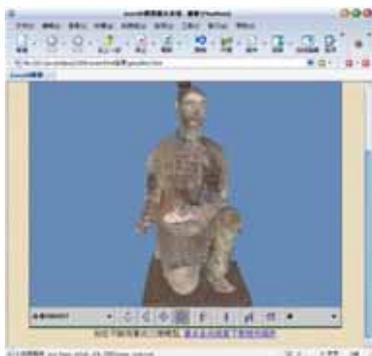


图8 三维图形浏览器展示 3

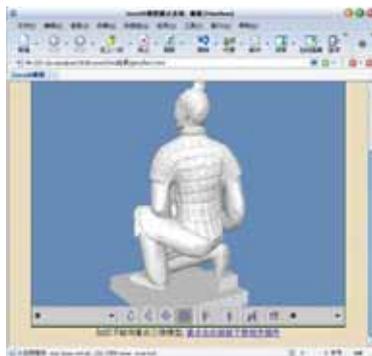


图9 三维图形浏览器展示 4

这里采用嵌入 IE 的 Java-applet 发布方式,使用 VRML 模型数据并采用 URL 导入资源文件。这样将类文件、数据文件、纹理图片等资源全部打入一个 jar 包后(1 353KB),分普通传输和递进传输两种方式进行了测试,结果如表 2 所示。

表2 测试环境和结果

测试环境		测试结果	
CPU	P4-3.0GHz	FPS	70
内存	512MB	普通传输响应时间	5s
显卡	ATI Radeon 9550	递进传输响应时间	2s
带宽	百兆局域网	递进传输总时间	6s

只有  $10^{-2}$ 数量级db的变化。在码率控制算法改进前后每帧分配的比特数相差较大时,会出现PSNR值变化较大,但总体的PSNR值趋于不变,这从上面的平均PSNR值可以看出。

### 3 结束语

实施改进的码率控制算法后,目标比特率的准确度,同原有的码率控制算法相比有所提高,特别是码流变化得比较缓和,这有利于编码后的视频码流对网络带宽的有限利用。另外,使用改进后的算法,直接使用当前宏块的MAD值,不使用前一帧图像当前宏块的MAD值,能避免场景切换带来的影响,因此不必考虑场景切换的检测以及处理,可以减少计算量。

### 参考文献

- 1 Nguyen V A, Tan V P. Efficient Transcoding Between H.263 and H.264/AVC Standards[C]//Proc. of IEEE International Symposium on Circuits and Systems. 2005-05.
- 2 Xiao Youneng, Lu Hong, Xue Xiangyang. Efficient Rate Control for MPEG-2 to H.264/AVC Transcoding[C]//Proc. of IEEE International Symposium on Circuits and Systems. 2005-05.
- 3 Li C X, Pan F, Lim K P, et al. Adaptive Frame Layer Rate Control for H.264[C]//Proc. of International Conference on Multimedia and Expo., Maryland, US. 2003, 1: 581-584.
- 4 Vetro A, Hata T, Kuwahara N, Kalva H. Complexity-quality Analysis of MPEG-2 to MPEG-4 Transcoding Architectures[C]//Proc. of International Conference on Consumer Electronics. 2002: 130-131.
- 5 Feamster N, Wee S J. An MPEG-2 to H.263 Transcoder[C]//Proc. of SPIE Voice, Video, and Data Communications Conference, Boston. 1999-09: 352-360.
- 6 范晓东, 孙 军. 基于场景切换检测的码率控制策略[J]. 上海交通大学学报, 2000, 34(6).
- 7 余 越, 周 键. 一种场景切换检测及自适应码率控制方法[J]. 通信学报, 1999, 20(5).

### 5 结束语

基于 Web 的交互式三维发布技术通过 Internet 和三维图形技术大大拓展了虚拟漫游的受众群,同时可以通过鼠标、键盘甚至操纵杆等输入设备对场景进行相当丰富的操作,大大增强了用户对虚拟场景的“参观体验”。

本文通过这些技术设计并实现了一个具有较好性能和交互能力的三维场景浏览器,并成功应用于相关项目中。然而,不同领域会对三维浏览提出不同的功能要求;对于场景漫游中一些诸如碰撞检测、场景渲染优化等问题,也需要进一步研究。因此,针对不同的需求,实现更为实用、完善的三维发布系统,是将来的工作方向。

### 参考文献

- 1 Sun Microsystems. The Java 3D API[R]. Technical White Paper, 1997-07.
- 2 Project Looking Glass[Z]. [2006-09]. [http://www.sun.com/software/looking\\_glass/](http://www.sun.com/software/looking_glass/).
- 3 杨 建, 黄正军, 唐 山. X3D 浏览器的一种实现方法[J]. 微机发展, 2004, 10(14): 91-93.
- 4 刘修国, 尚建嘎. 三维场景浏览器的设计与实现[J]. 计算机工程与应用, 2002, 38(10): 117-120.