

# AVS 编码器 Slice 并行处理算法研究与实现

许昌满 李国平 王国中

(上海广电(集团)有限公司中央研究院,上海 200233)

**摘要** 介绍了一种基于多 slice 的并行 AVS 实时编码器的算法研究与实现,基于 slice 的并行编码的优点是处理速度快、延迟小和数据处理简单,缺点是条纹效应。介绍了一种对基于 slice 的并行编码带来的条纹效应缺点进行改进的方法。实验结果表明,实现标清 AVS 实时编码器是可行的。

**关键词** AVS 实时编码器 条纹效应 并行处理 条带

中图法分类号: 文献标识码: A 文章编号: 1006-8961(2009)06-1108-06

## Study and Implementation of AVS Encoding Algorithm Based on Slice Parallel

XU Chang-man, LI Guo-ping, WANG Guo-zhong

(Central Research Academy, SVA Group, Shanghai 200233)

**Abstract** This paper introduces a kind of AVS real encoding algorithm based on slice Parallel. The advantages of AVS real encoder based on slice parallel is high encoding speed, less delay time and low process complexity. This paper also puts forward a method of improving the slice effect. The experimental result demonstrates that it is absolutely feasible to implement a real-time AVS SD encoder.

**Keywords** AVS realtime encoder, slice effect, parallel process, slice

## 1 引言

AVS(audio video coding standard)<sup>[1]</sup>是基于中国创新技术和部分公开技术的自主知识产权的标准,其性能与最新的国际标准 H. 264 相当,编码效率比 MPEG-2 高 2~3 倍,算法复杂度比 H. 264 低,并且其专利授权模式简单,专利方面的费用明显低于同类压缩编码标准。AVS 视频标准采用了一系列技术来达到高效率的视频编码,包括帧内预测、帧间预测、变换和量化、熵编码等。

## 2 编码器并行处理

由于 AVS 视频编码过程是一个非常复杂的计算过程,它对计算机的处理能力要求非常高。传统上,处理速度的提高往往是依赖于处理器速度的提高,但串行计算机的性能总是有限的。因此,在分布式系统上采取并行方式编码是必然的<sup>[2]</sup>。现在多处理器计算机已经非常普及,各大计算机公司都推出各种各样的多处理机。同时一些专用的信号处理硬件被设计出来用于加速视频编码的处理。然而,用软件实现视频流编码有很多好处:对采用新算法改进编码器具有很大的灵活性,可以不使用昂贵的

基金项目:上海市科学技术委员会基金项目(06QB14050)

收稿日期:2007-08-10;改回日期:2007-11-13

第一作者简介:许昌满(1964~)男。高级工程师,2002年于上海大学获通信与信息系统专业博士学位。研究方向为视频解码与嵌入式系统。E-mail:xu\_cm@sva.com.cn

硬件设备,可以运行在通用的系统,把它移植或应用于其他领域也很方便。所以采用并行程序设计方法来提高 AVS 视频比特流的编码效率。采用多处理器并行处理的方式<sup>[3]</sup>,分担计算任务,从而使得数据得以及时处理、达到图像实时编码的要求。

在图像编码时,图像序列(sequence)是指一个被处理的连续图像,图像序列可以是隔行扫描的也可以是逐行扫描的。图像组(GOP)是将一个图像序列中连续的几幅图像组成一个小组,这样方便随机存取编辑。图像以亮度数据阵列为基准被分为若干个 $16 \times 16$ 像素的单位,称做宏块(macroblock)。在图像中,数据被分成包括若干个连续宏块的宏块条(slice)。

在 AVS 中可能被选择作为一个任务的有:sequence, GOP, picture, slice, macroblock。但一般不采用 picture 作为任务并行编码运行,由于 picture 中含有 P 帧和 B 帧,这两种帧的编码要用相邻的其他帧的信息,这样就会造成在任务之间的相互等待,增加处理器之间的通讯量。而 GOP 和 sequence 是一个可以独立编码的帧序列。GOP 之间、sequence 之

间可以独立并行编码。但采用 sequence 作为任务就会使任务过于庞大,同时也不利于平衡处理器之间的负载,因此 GOP 作为任务是一个比较好的选择。

采用 slice 作为任务原因是在 AVS 具有 slice 的独立语法结构,可以通过选择 slice 的类型和 slice 所包含的宏块数目来定义处理器处理 slice 的大小,由各个隶从机对各个 slice 分别编码,再由管理机负责各个数据流重新组合成正确的视频数据流。

综上所述,在实际应用中一般选择 GOP 或 slice 作为任务而被并行编码运行。

GOP 级并行编码软件的结构如图 1 所示。

一般的 GOP 序列只有第一帧为 I 帧,假设有这样一个 GOP 序列:IBBPBBPBBPBB。由于各个 GOP 在不同的隶从机上被压缩,因而 GOP 必须被独立编码,即不能以其他组内的图像作为 GOP 序列图像进行估计或运动补偿,这是将最后一幅图像为 I 图像的原因。每台隶从机一次把从采集卡获得的 13 幅图像读入内存,分别进行处理,编码后的数据流送到缓存区依次输出。

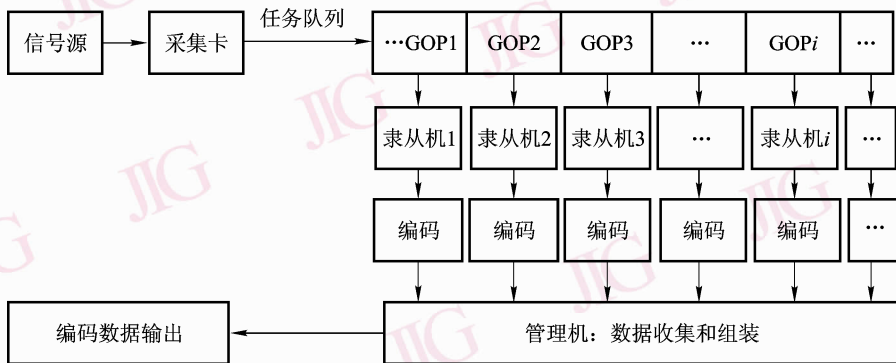


图 1 GOP 级并行处理

Fig. 1 GOP level parallel process

基于 GOP 级的并行的优点是 GOP 的并行易于平衡负载,任务比较大、独立,因此访问任务队列的频率较少,编程实现较容易,GOP 的独立性使得编码隶从机之间相互等待时间非常小,等待的时间主要是管理机等待接收隶从机发送过来的数据上。基于 GOP 的并行编码可以减少处理器间参考数据的交换。因为每个 GOP 的头后面紧跟的是 I 帧,而 I 帧是内部编码的,不需要参考先前的数据,而在文中 GOP 的最后一幅图像也为 I 帧,这使得前一个 GOP 中的帧数据不需要参考后一个 GOP 中的帧数据,这样不会造成处理机之间的相互等待以及引入多余的

通信费用。但是基于 GOP 的并行带来的编码延迟比较大,各个隶从机需要把一个完整的 GOP 数据都读入缓存才能开始编码,这在 GOP 的数量较大时,编码的延迟更加明显。基于 GOP 的并行运算的另外一个缺点是系统对内存的需求比较大,对内存需求的大小取决于隶从机编码器的数目和图像的尺寸和 GOP 的大小。因为在图像编码中 P 帧、B 帧的处理在每个隶从机上需要保存一些参考帧,隶从机数目越多,帧缓存数目越多,图像尺寸越大,帧缓存所需要的内存空间越大,GOP 的尺寸越大,任务队列所需要的空间越大。

基于 slice 级的并行编码,各个隶从机按照一定的顺序读取由视频采集卡得到的原始视频信号数据,再对每一帧 picture 进行 slice 划分,slice 划分后对一帧 picture 的 slice 进行并行编码,编码数据发送到管理机,由管理机对收到的视频比特流进行数据组装,当一幅图像 picture 中的 slice 处理完毕之后,再处理下一个 picture 中的 slice,同样管理机负责收集和组装数据。slice 级并行编码软件的结构如图 2 所示。

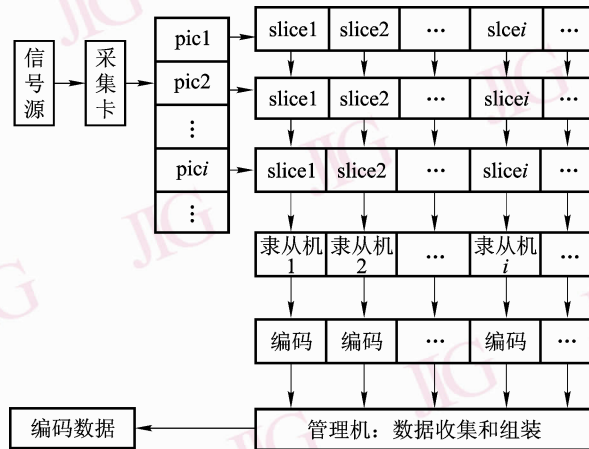


图 2 slice 级并行处理

Fig. 2 Slice level parallel process

slice 级并行编码的优点是编码的延迟小,延迟来自编码器把待编码的图像和 2 个参考图像读入内存的时间。

slice 级并行编码的另外一个优点是软件对内存需求相对基于 GOP 级的并行编码要少得多,因为系统中所有的隶从机只对一帧图像进行编码,在系统中只需保存与该帧图像编码有关的一些参考帧图像数据。内存的需求只与图像的尺寸有关。

在 slice 级并行编码器中,首先把图像序列进行排序,I 帧图像只做帧内编码,P 帧是使用前向帧作为参考帧形成的编码帧。既可以用前向帧作为参考帧,同时也可以以后向帧作为参考帧形成的编码帧叫 B 帧,B 帧既可以以 I 帧,也可以以 P 帧作为参考帧。在 I 帧的编码过程中编码器将整幅图像编码成一个条带,不采用多个条带,这样不仅在本 I 帧没有条纹,而且有效地降低了在以后需要参考 I 进行编码的 P 帧和 B 帧接下来的帧中的条带间条纹,在 P 帧和 B 帧中进行编码时为了提高编码速度,需要把一帧图像分割成多个 slice 进行编码,每个 slice 独立编码,各个 slice 之间不互相参考,但每个 slice 都可

以参考已经编码完的图像。由于一帧图像的每个 slice 同时并行进行编码,因此能够大大提高编码速度,从而对标清图像进行编码时,编码速度能够达到实时要求。实验表明,对于小图像,一般情况下 slice 的数量分为 2 个。如果 slice 分多了,因为各个 slice 在编码时,是独立进行的,slice 之间不能互相参考,这样在帧间编码时运动矢量和帧内编码模式的参考像素不能相互参考,导致编码效率下降。

由 AVS 的编码数据流的语法可知,对于管理机来说,对 slice 的并行编码比对 GOP 的并行编码在码流的封装上要简单些,因为在编码数据流中所有的视频序列头信息、图像头信息和 slice 头信息都放置在输出码流的前部,所以管理机将各个隶从机的压缩比特流信息收集后,进行数据封装时要简单一些。

但是基于 slice 的并行编码会带来一个突出的缺点是条纹效应。

### 3 条纹效应消除

视频传输面临的问题就是如何在给定带宽条件下实时传输编码的数据,它对编码器的要求是编码器的输出必须码率恒定。这就需要在编码端采用合适的码率控制策略,对编码参数进行调整,一般是通过调节量化参数(QP),从而有效地分配码率,使实际使用的码率尽可能和目标码率一致,同时获得高的视频质量<sup>[4]</sup>。一般的码率控制策略是:根据给定的目标码率,计算 GOP 的编码比特数,再根据 frame 的不同类型分配给每帧不同的比特数,然后根据每个 frame 的比特数和 frame 的 QP 确定的 MB 的初始 QP,再根据已编码的 MB 的目标比特与统计已用的比特进行比较,如果已用的比特比目标比特多,则增加 QP,如果已用的比特比目标比特少,则减少 QP。

在实际应用中,采用 slice 级并行处理对图像进行实时编码时,由于采用了多个 slices,在相邻的 slices 之间,由于相互间编码的独立性,编码后的图像纹理特征不可避免地产生了一致性,这种不一致性使得编码后相邻 slices 的图像看起来好像有条线,这种现象称为条纹效应(slice Effect)。这种条纹严重影响了解码图像的主观质量,特别是当一幅图像中采用多个条带时,这种条纹就多起来,这样更加损害了图像的主观质量。

为了消除条纹效应,提出了一种消除条带间条

纹的方法,步骤如下:

(1)在I帧中只采用一个条带,消除I帧的条带间条纹。

I帧图像采用帧内编码方式,所以在I帧中,每个宏块都采用帧内编码,而宏块的集合就是条带,如果存在多个条带,则相邻条带间的宏块不能相互参考,就使得各个条带间的图像纹理特征很难相同。而如果I帧存在条带间条纹,在接下来的各个帧中都将存在条带间条纹,而且很难消除;

在I帧的编码过程中编码器将整幅图像编码成一个条带,不采用多个条带,而是只采用一个条带,这样不仅在本I帧没有条纹,而且有效地降低了在以后接下来的帧中的条带间条纹。

(2)消除B帧和P帧的条带间条纹。

P帧和B帧图像采用帧间编码方式,即同时利用了空间和时间上的相关性;P帧图像只采用前向时间预测,可以提高压缩效率和图像质量;B帧图像采用双向时间预测,可以大大提高压缩倍数;

① 计算一帧图像内相邻两个条带交界处的量化参数QP值;

在编码过程中,编码器先对图像块进行残差变换,然后根据量化参数值对变换系数进行量化,如果相邻图像块的QP相同,则这些相邻块的边界块效应就很小,如果在条带边界处图像块的QP相同,条带边界处的条纹就被消除了;

假设一帧图像由 $N$ 个条带组成,则每个条带的平均QP分别为 $QP_1, QP_2, \dots, QP_N$ ,则第 $i$ 和 $i+1$ 条带交界处的QP为

$$QP_{i,i+1\text{th-slice}} = (QP_i + QP_{i+1})/2 \quad (1)$$

② 限制条带内宏块的QP值;

在第 $i$ 和 $i+1$ 条带交界处,距离这个交界处的第 $n$ 行宏块位置的QP值由式(2)限制。

$$\begin{aligned} QP_{i,i+1\text{th-slice}-n} &\leq QP_{n,i,i+1\text{th-slice}} \\ &\leq QP_{i,i+1\text{th-slice}+n} \end{aligned} \quad (2)$$

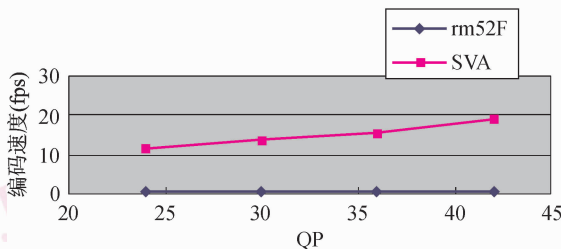
式中, $n \leq NE$ ;

$NE$ 为边界处理区域, $NE$ 取值由图像和 $N$ 的大小决定,一般 $NE \geq 2$ ;如果图像的像素小于288时, $N \leq 2, NE = 2$ ;如果图像的像素大于288而且小于480时, $N \leq 4, 2 \leq NE \leq 3$ ;如果图像的像素大于480时, $N \leq 16, 2 \leq NE \leq 8$ ;

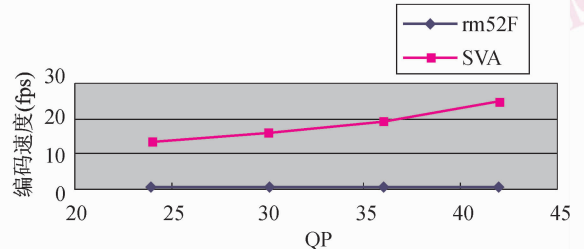
通过步骤①和步骤②的处理,使得条带交界处宏块的QP值基本相同,这样的控制使得交界处区域图像融合成一体,从而消除了条带间条纹。

## 4 AVS多slice并行编码器实验结果

为了验证提出的多slice算法对编码器性能的影响,在同一台双核、两个超线程服务器上,编码器编码时采用4个slice,对AVS标准参考编码器rm52f和基于多slice的SVA编码器进行测试,分别把QP设置为:24,30,36和42,在测试的序列为:Crew\_720×480 30 fps和flower garden: 720×576 25 fps,运动搜索的范围均为:48,测试序列的总帧数为300,分别测试不同量化步长时在这两个编码器下的PSNR值、编码比特率和编码速率,根据测试数据的结果绘出的图见图3和图4。图4是率失真曲线。



(a) Flower garden序列



(b) Crew序列

图3 相同的QP编码不同图像序列的encoding speed比较

Fig. 3 The comparison of encoding speed with same QP in different sequence

对测试序列flower garden\_720×576和crew\_720×480,从图3可以看出采用多slice并行处理的SVA encoder比rm52F编码器在QP分别为:24,30,36和42时,编码速度分别提高18.06,20.67,

23.22,28.41倍和22.03,24.89,28.47,33.93倍。把一幅图像分为4个slice进行编码,理论上编码速度可以提高4倍,但实际上的编码速度的提高大约为2~3倍,因为把一幅图像分为4个slice进行编

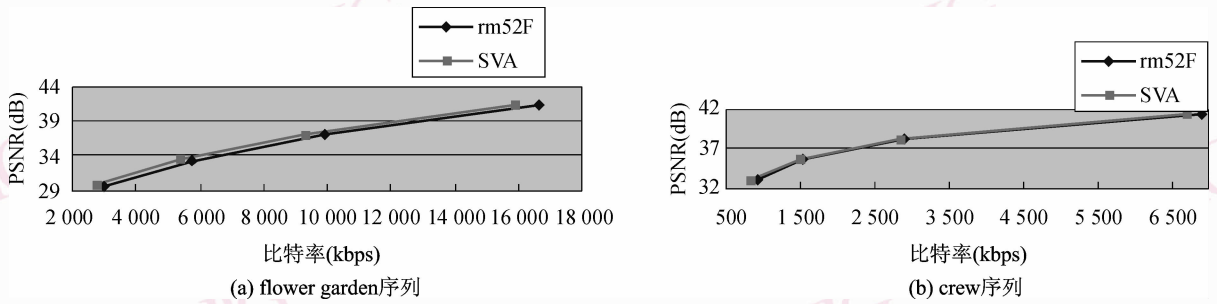


图 4 相同的 QP 编码图像不同序列的编码重建帧 PSNR 比较

Fig. 4 The comparison of reconstructed image with same QP in different image

码时,4 个 slice 的编码速度快慢不一样,必须等编码最慢的那个 slice 编码结束后,才能把一帧图像的编码数据输出,码流的输入和输出是串行的,文件的读写也是串行的,把一幅图像分为 4 个 slice 进行编码时,增加了多线程之间的通信开销,同时实验机器是双 CPU,双核,不是完全意义上的四 CPU,也就是说速度达不到四 CPU 的速度。整个编码程序的速度提高的原因还在于运动搜索的快速算法、模式判别的快速算法和最后的汇编优化等,所有的优化综合的结果,SVA 编码器的编码速度比 rm52F 的编码速度提高了 18~22 倍。从数据处理量上看,以  $720 \times 576$  为例,一个宏块(MB)大小为  $16 \times 16$ ,把一幅图像当作 1 个 slice 进行编码,需要编码 1 620 个宏块,而把一幅图像当作 4 个 slice 进行编码,一个 slice 仅包含 405 个宏块,也就是说 4 个 slice 编码时需要处理的数据大约为一个 slice 编码需要处理的数据的 1/4,仅此一项,编码的数据减少使编码速度提高大约 4 倍。运动搜索的快速算法、模式判别的快速算法和最后的汇编优化等模块的应用,使得需要处理的数据大大减少,从而最终使编码速度提高达 18~22 倍。从图 4 可以看出 SVA 编码器在保持图像质量和编码比特率几乎不变的情况下,编码速度有较大提高。同时,在图 4 中,bitrate 的数据应该大约相等,由于 SVA 编码器中的运动搜索和模式选择做了大量的优化,因此,和 rm52F 相比,SVA 编码器的 bitrate 的数据有所减少。

消除条纹效应的基本思路是在 I 帧的编码过程中,编码器将整幅图像编码成一个条带,不采用多个条带。而是只采用一个条带。这样不仅在本 I 帧没有条纹,而且有效地降低了在以后需要参考 I 进行编码的 P 帧和 B 帧接下来的帧中的条带间条纹。在 P 帧和 B 帧中进行编码时,为了提高编码速度,

需要把一帧图像分割成多个 slice 进行独立编码。由于每个 slice 编码时是按照各自的 QP 值进行比特分配,这样在 slice 的边界处会出现一条条纹,影响了图像的质量。为了消除条纹效应,必须使条纹的连接处的 QP 控制相同,使得 QP 在 slice 的边缘的变化是连续的,这样使得条带交界处宏块的 QP 值基本相同,这样的控制使得交界处区域图像融合成一体,从而消除了条带间条纹,具体的算法参考前文叙述。条纹效应如图 5 所示,图 5 为有条纹效应的多 slice 编码后重建图像,图 6 是采用消除条带间条纹的方法编码后的重建的同一帧图像。和图 6 比较可以看出图 5 的中部有 1 条(把图像分为 2 个 slice)明显的条纹,图 7 是 cctv1 序列(共 300 帧)的有条纹效应的编码重建图像和采用消除条带间条纹的方法后的重建图像 PSNR 比较,由图 5、图 6 和图 7 可以看出采用消除条带间条纹的方法后的编码重建图像不仅主观质量得到改善而且客观 PSNR 值也有所提高,图 7 中的——线是没有采用消除条带间条纹的方法的重建图像的率失真曲线,另一条线是采用消除条带间条纹的方法后的重建图像的率失真曲线。

图 5 多 slice 编码后重建图像的条纹效应  
Fig. 5 The slice effect of reconstructed image



图6 采用消除条带间条纹的方法后的重建图像

Fig. 6 The reconstructed image with the method of slice effect eliminated

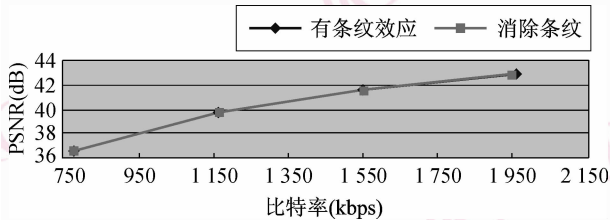


图7 有条纹效应的编码重建图像和采用消除条带间条纹的方法后的重建图像 PSNR 比较

Fig. 7 The PSNR comparison of reconstructed image

## 5 结论

针对 AVS 编码器的特点,提出了基于条带的 AVS 编码器的并行处理设计方案。该方案能大大

加快编码器的编码速度,而且编码的延迟小,数据处理简单。同时提出一种消除条带间条纹的方法,该方法完全消除了条带间条纹,而且极大地提高了图像的主观质量,提出的消除条带间条纹的方法计算复杂度很低,不仅适合各种软件应用,而且也适合硬件的实时工作。基于条带的并行处理设计方案,在视频编码器优化中,采用程序优化、编码算法优化和针对特定机器的汇编优化相结合,在保证编码性能的基础上,大幅度提高了编码速度。从实际效果看,用此设计方案实现 AVS 标清编码器已经达到编码器实时性的要求。该编码器系统由上海广电(集团)有限公司中央研究院研发成功,已经进入产业化应用。

## 参考文献 (References)

- 1 GB/T 20090.2 - 2006, Information Technology-Advanced Coding of Audio and Video-Part 2:Video[S]. [GB/T 20090.2-2006. 信息技术. 先进音视频编码. 第二部分:视频[S].]
- 2 Ning Hua, Li Jin-tao. A study of parallelism in MPEG-4 video encoder [J]. Computer Engineering and Applications, 2002; **38**(7):9-12. [宁华,李锦涛. MPEG-4 视频编码的并行实现[J]. 计算机工程与应用, 2002; **38**(7):9-12.]
- 3 Shen K, Lawrence A R, Edward J D. A parallel implementation of an MPEG1 encoder [A]. In: Proceedings of the SPIE Conference on Digital Video Compression: Algorithms and Technologies [C], Washington: SPIE Press, 1995: 407-418.
- 4 Lin L J, Ortega A. Bit-rate control using piecewise approximated rate-distortion characteristics[J]. IEEE Transactions on Circuits and Systems for Video Technology, 1998, **8**(4):446-459.