

一种基于残差分析的 H. 264/AVC 帧间块尺寸选择快速预测算法

王正宁 彭强 诸昌铃

(西南交通大学计算机与通信工程学院, 成都 610031)

摘要 为了进一步减小宏块的帧间预测误差,新的视频编码标准 H. 264/AVC 采用了灵活多样的块划分类型,块划分尺寸包含 16×16 到 4×4 共 7 种情况。更准确的块匹配带来了编码效率的提高,同时也使计算复杂度急剧增加,当采用率失真全搜索算法遍历所有可能的块划分尺寸组合时更是如此。这一问题也成了影响 H. 264/AVC 实际应用的因素之一,针对这一问题提出了一种基于残差纹理分析的快速块尺寸划分算法。首先通过一次宏块级的运动搜索预处理得到各个 4×4 块运动补偿后的残差;然后根据得到的残差纹理大小及分布情况对帧间块的划分尺寸做出快速选择。大量的实验验证了该算法在几乎不牺牲图像质量和压缩效率的基础上将这一过程的计算复杂度平均减少了 50%。

关键词 视频编码 H. 264/AVC 块划分 预测残差

中图分类号: TN941.1 **文献标识码**: A **文章编号**: 1006-8961(2005)12-1584-07

Residual Texture Based Fast Block-Size Selection for Inter-frame Coding in H. 264/AVC

WANG Zheng-ning, PENG Qiang, ZHU Chang-qian

(School of Computer & Communication Engineering, Southwest Jiaotong University, Chengdu 610031)

Abstract One of the new features adopted in video coding standard H. 264/AVC is the utilization of flexible block sizes ranging from 16×16 to 4×4 in inter-frame coding. The aim is to reduce the error due to fixed block size prediction within a macroblock. However, this feature requires the encoder to compute the rate-distortion cost for every available block size. As a result, the complexity and computation load of video coding increase drastically, which make it very difficult for practical applications. In this paper we proposed a residual texture based fast block size selection algorithm for inter-frame coding. Finally, we do a motion estimation (ME) for a macroblock and get the residual, then we predict the block size by analyzing the residual texture. Extensive simulations verify that the proposed method speeds up the block-size selection procedure by 50% without sacrificing picture quality and compression ratio. A comparison between the proposed algorithm and other algorithms is also provided.

Keywords video coding, H. 264/AVC, block-size selection, residual

1 引言

H. 264/AVC 是 ITU-T 和 MPEG 联合推出的最新视频编码标准。该标准中使用了许多的编码工具来提高编码器和解码器的性能,其中之一就是进

行帧间预测时采用的多种块划分尺寸^[1],如图 1 所示。由于块划分尺寸的增多,每个块的预测误差将会减小,但带来的却是计算复杂度的急剧增加,这严重地影响了对视频的实时编码。

目前已提出了一些算法对块划分尺寸进行快速选择,但这些算法都是基于被编码宏块 (Macroblock,

基金项目:铁道部科技研究开发项目(2003X040-A)

收稿日期:2004-12-06;改回日期:2005-03-10

第一作者简介:王正宁(1979 ~),男。现为西南交通大学计算机应用专业博士研究生。主要研究方向为视频压缩与传输控制技术。

E-mail: znwang@home.swjtu.edu.cn

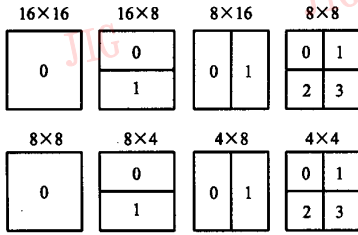


图 1 H. 264/AVC 中宏块的 7 种帧间预测块尺寸

Fig. 1 Seven block size for inter-frame coding in H. 264/AVC

MB)本身的纹理信息进行的,忽略了帧间预测自身的特点^[2]。为此,针对这一不足,提出了结合运动搜索预处理,得到块的预测残差,然后对预测残差的纹理进行分析并根据分析结果对宏块划分尺寸做出快速选择的算法。

2 H. 264/AVC 中块划分算法模型

在 H. 264/AVC 中进行帧间预测时为了得到最小的预测误差,标准不再采用固定块尺寸而是可变速块尺寸(16×16 到 4×4)对每个划分的块进行运动搜索和运动补偿以得到最佳的匹配块,从而提高压缩效率。为了得到一个宏块的最佳划分尺寸,H. 264/AVC 采用率失真优化模型^[3-5]计算宏块在每种划分尺寸下的率失真开销:

$$J_{mode} = D + \lambda_{mode} R \quad (1)$$

其中, J_{mode} 代表率失真开销(rate distortion cost); λ_{mode} 为拉格朗日乘数; D 表示原始宏块与预测宏块之间的失真度; R 为当前划分尺寸下编码该宏块所需的实际比特数。在帧间预测时,宏块最终选取的是划分尺寸为所有的块尺寸组合中能使 J_{mode} 最小的那一个。整个率失真开销的计算如图 2 所示。

从图 2 中可以看出:为了计算每种划分尺寸下的率失真开销,编码器要重复执行相同的计算:整数 DCT 变换/量化、熵编码、反量化/整数 DCT 反变换。

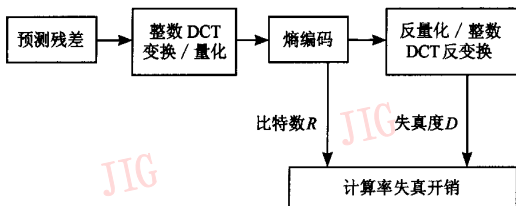


图 2 H. 264/AVC 中率失真开销的计算

Fig. 2 Computation of R-D cost in H. 264/AVC

而且,对于不同特点的视频序列,有些划分尺寸出现的概率很小,但是由于率失真算法本身的特点,为了确定最佳的块划分尺寸,编码器不得不计算出所有块划分尺寸组合下的率失真开销并选择出开销最小的那种块尺寸组合。这些都说明了率失真优化算法的计算复杂度相当高。

3 现有快速算法的特点及不足

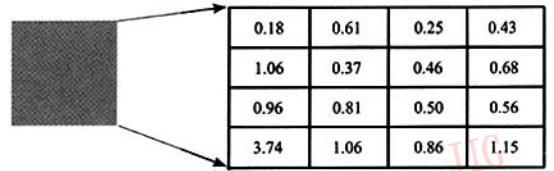
针对率失真优化算法的不足,文献[6]、[7]提出了在帧间预测时宏块划分尺寸的快速选择算法。文献[6]提出了基于物体运动精度计算相应纹理信息,并根据纹理的复杂程度计算可能采用的宏块划分尺寸;文献[7]提出了利用当前块的 DCT 系数计算该块的能量(需要单独进行 DCT 变换),从而得出块的纹理复杂度,并进行快速的划分尺寸选择。

上述两种快速算法都是利用图像的纹理复杂度预测当前宏块的划分尺寸。但是通过大量的实验发现,在进行帧间预测时,原始宏块的纹理复杂度与该宏块的最终划分尺寸并没有必然联系。也就是说,纹理复杂的宏块并非按 8×8 甚至是 4×4 的块尺寸进行划分;纹理简单的宏块也并不完全按 16×16 的尺寸进行划分。例如,当纹理复杂度非常高的图像连续重复出现时,每一帧中的所有宏块都可以按照 16×16 的尺寸进行划分。所以说,块的划分尺寸与块的预测准确程度有关,而现有的快速算法却未考虑到这一点。

图 3 和图 4 分别是 Foreman 序列中抽取出的 2 个宏块的原始图像和残差图像及各自的纹理分布。可以看出,图 3(b)中残差图像的纹理复杂度普遍较小且分布比较均匀,而图 4(b)中残差图像的纹理复杂度值较大且分布不均匀。在测试模型中通过率失真优化算法得到的图 3 中的宏块最终划分尺寸为 16×16;图 4 中的宏块最终划分尺寸为 8×8。在各个 8×8 块的子划分中:第 0 个 8×8 块的划分尺寸为 4×4,第 1 个 8×8 块的划分尺寸为 4×8,第 2 个 8×8 块的划分尺寸为 8×8,第 3 个 8×8 块的划分尺寸为 4×4。通过观察和分析图 3 和图 4 中纹理的大小及分布并比较率失真算法得到的块尺寸划分结果,可以发现通过原始宏块的纹理信息很难判断出宏块的划分尺寸及其子块的划分尺寸;相反可以看出运动补偿残差纹理的大小及分布与划分尺寸有着紧密的联系。



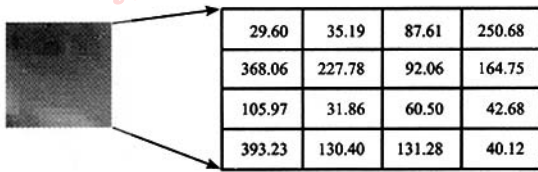
(a) 原始宏块及纹理分析



(b) 运动补偿后的残差宏块及纹理分析

图 3 foreman(QCIF)第 2 帧第 8 个宏块的图像及纹理分布

Fig. 3 Images and textures of the 8th macroblock in frame 2 of foreman QCIF sequence



(a) 原始宏块及纹理分析



(b) 运动补偿后的残差宏块纹理分析

图 4 foreman(QCIF)第 66 帧第 39 个宏块的图像及纹理分布

Fig. 4 Images and textures of the 39th macroblock in fram 66 of foreman QCIF sequence

4 本文算法

针对现有快速算法及率失真算法的不足,提出了结合运动搜索对宏块的预测残差进行纹理分析,并根据残差的纹理大小及分布对块划分尺寸进行快速选择的算法,取得了很好的效果。

该算法的主要思想是:假设当前要进行帧间预测的宏块及其各种划分尺寸的块在参考帧中都存在可补偿的块。当运动补偿之后残差的纹理大于该块划分尺寸下的纹理阈值并且当前块的划分尺寸不是 4×4 这种最小尺寸时,则认为当前划分尺寸下通过运动搜索得到的运动向量不能够很好地满足各个子块的运动补偿,因此该块需要进一步划分成更小的块。当运动补偿之后残差的纹理小于该块划分尺寸下的纹理阈值时,说明这种划分尺寸下该块在参考帧中已经存在较匹配的运动补偿块,因此不需要进一步划分该块。整个算法中运动搜索分为两级: 16×16 块和 8×8 块。 16×16 块运动搜索及残差纹理分析主要用来决定 $16 \times 16, 16 \times 8, 8 \times 16$ 和 8×8 这 4 种划分尺寸; 8×8 块运动搜索及残差纹理分析主要用来决定 $8 \times 4, 4 \times 8, 4 \times 4$ 这 3 种划分尺寸,这是因为相对于 16×16 块,整个 8×8 块的运动补偿

匹配程度对它下一级子块划分尺寸的影响更大,因此为了提高 $8 \times 4, 4 \times 8, 4 \times 4$ 这 3 种划分尺寸的预测准确性需要对 8×8 块进行一次运动搜索。算法具体步骤如下:

(1) 对当前宏块进行一次 $1/4$ 像素精度的运动搜索,得到最佳运动向量处宏块的预测残差。

(2) 将残差宏块分为 16 个 4×4 大小的子块。

(3) 按下式计算并记录第 $m(m = 0, 1, 2, 3)$ 个 8×8 块中的第 $n(n = 0, 1, 2, 3)$ 个 4×4 残差子块的纹理复杂度。

$$C_{m,n} = \frac{1}{16} \sum_{i=0}^3 \sum_{j=0}^3 (x_{i,j} - \bar{x})^2 \quad (2)$$

其中, $x_{i,j}$ 代表 4×4 残差块中的每个残差值, \bar{x} 代表该块残差值的均值。

(4) 把第 m 个 8×8 块的 4 个 4×4 块的纹理复杂度进行相加,结果作为该 8×8 块的纹理复杂度。整个宏块的纹理复杂度用所有 8×8 块的复杂度之和表示。

$$C_m = \sum_{n=0}^3 C_{m,n} \quad (3)$$

$$C_{MB} = \sum_{m=0}^3 C_m \quad (4)$$

(5) 统计 8×8 块中纹理复杂度大于 8×8 块纹理复杂度阈值 T_8 的块的个数 $count8$ 及位置 $Track_b8$ 。

(6)判断当前宏块的划分尺寸。

如果 $count8 > 1$ 且 $C_{MB} > 4T_8$, 则宏块采用 8×8 划分尺寸, 并执行第 8 步;

如果 $count8 = 0$, 则宏块采用 16×16 的划分尺寸;

否则, 执行第 7 步;

(7)如果 $Track_b8 = 0$ or $Track_b8 = 3$, 且 $C_1 \leq C_2$, 则宏块采用 8×16 划分尺寸; 否则, 采用 16×8 划分尺寸。

如果 $Track_b8 = 1$ or $Tranck_b8 = 2$, 且 $C_3 \leq C_0$, 则宏块采用 8×16 划分尺寸; 否则, 采用 16×8 划分尺寸。

(8)计算单个 8×8 块的划分尺寸。如果 $count4 >$

1 且 $C_m > T_8$, 则对该 8×8 块单独进行一次运动搜索, 重新统计运动搜索后该 8×8 块中残差纹理复杂度大于 T_4 的 4×4 块的个数 $count4$ 及所在位置 $Track_b4$ 。否则检查 $count4$ 是否等于零, 如果成立则采用 8×8 的划分尺寸; 否则采用与第 7 步相似的做法, 根据 $Track_b4$ 的位置及 $C_{m,1}$ 和 $C_{m,2}$, $C_{m,0}$ 和 $C_{m,3}$ 的关系选择 8×4 或 4×8 块划分尺寸。如果运动搜索后 $count4$ 仍然大于 1 且 $C_m > T_8$ 则该 8×8 块采用 4×4 划分尺寸; 重复步骤 8 直至完成宏块中所有 8×8 块的划分尺寸预测。

通过以上的算法得到最终的划分尺寸并结束对当前宏块的划分尺寸的预测。整个算法的流程图如图 5 所示。

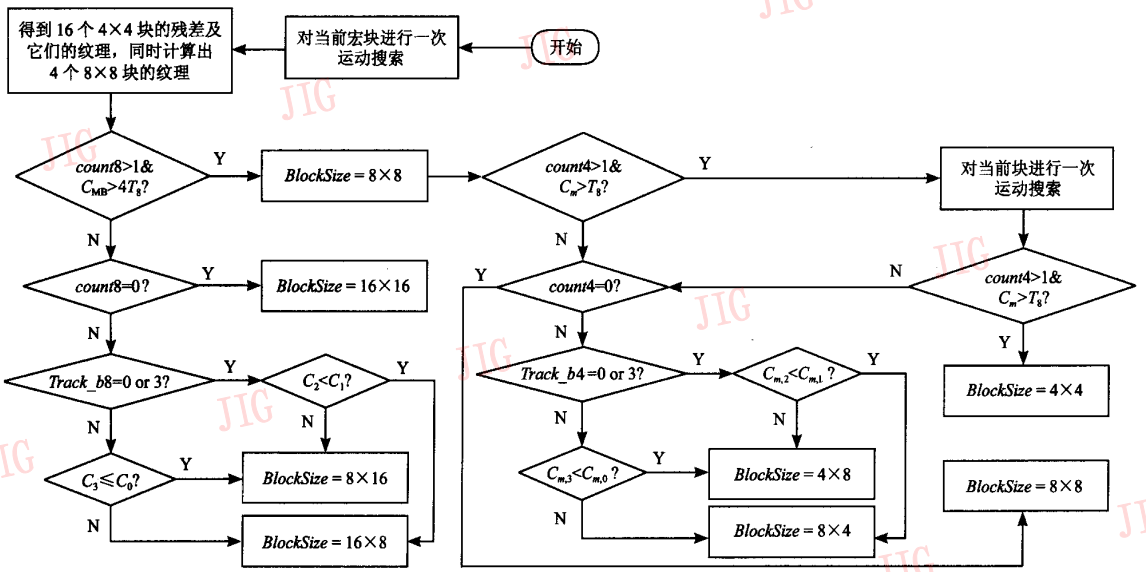


图 5 快速块划分尺寸预测算法流程图

Fig. 5 The flowchar of the proposed algorithm

5 实验结果

本文算法是基于 H. 264/AVC 的测试模型 JM7. 5c^[8]实现的, 运动搜索算法采用的是全搜索算法。在对图 3 和图 4 中的宏块进行处理时, 图 3 通过本文算法得到的划分尺寸与率失真优化算法得到的结果是一致的; 图 4 中的宏块通过本文算法得到的划分尺寸除了第 3 个 8×8 块的尺寸与率失真优化算法得到的结果不同外, 其余各 8×8 块的划分尺寸均与率失真优化算法得到的结果完全相同。为了

客观全面地验证算法的有效性, 对不同大小、不同运动程度和纹理特点的视频序列分别做了测试, 选择的测试序列具体为:

QCIF: foreman, Suzie, news, container

CIF: Akiyo, coastguard, mobile

实验中配置文件的主要参数设置如表 1 所示。表 2 统计比较了各个测试序列在本文算法、文献[7]算法下相对于 H. 264/AVC 中率失真优化算法的性能差异。PCHG、BINC 和 TSAV 分别代表不同算法下整个序列的平均 PSNR(峰值信噪比)变化值、码率增加值和节省的计算时间比例。在 PSNR 值方面

表 1 编码器主要实验参数设置

Tab.1 The main configure parameters for the encoder

参数	取值
编码帧数	整个序列
GOP 结构	IPPP
是否使用 Hadamard 变换	是
输入序列帧频 (Hz)	15
熵编码算法	采用 CABAC 编码
量化参数 (QP)	28
运动向量精度	1/4-像素精度
运动搜索范围	±16 像素
参考帧数目	1
块划分尺寸	全部采用

表 2 本文算法、文献[7]算法相对于 H.264/AVC 算法的性能比较

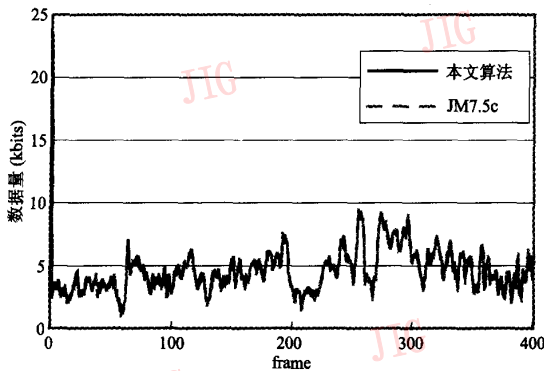
Tab.2 The experimental results of the proposed algorithm and the algorithm in [7] VS. H.264/AVC

视频序列	PCHG (dB)		BINC (%)		TSAV (%)	
	文献[7]	本文	文献[7]	本文	文献[7]	本文
foreman	-0.09	-0.01	4.98	1.97	25.14	48.98
Suzie	-0.08	0.00	5.35	2.67	31.47	49.59
news	-0.08	-0.03	2.86	1.29	31.32	47.89
container	-0.02	-0.02	0.93	1.24	29.81	48.15
Akiyo	-0.03	-0.01	1.56	2.02	29.48	49.00
coastguard	-0.11	-0.01	4.07	0.88	28.21	50.34
mobile	-0.10	0.00	2.09	0.54	17.05	49.53

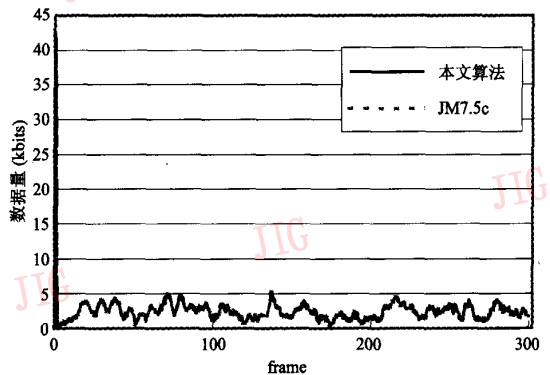
本文算法略有下降,平均为 0.01dB,而对 Suzie 和 mobile 序列做到了平均 PSNR 值保持不变。在最后产生的码率方面,本文算法平均高出 1.52%。从表 2 中可以看出本文算法不仅相对于率失真算法取得了很好的效果,而且与文献[7]算法相比较,在表 2 中的几项指标下本文算法均优于文献[7]的快速算法。图像质量的下降和码率的增加主要是因为:率失真优化算法本身就是为了追求码率和图像质量最优组合而提出的,它通过遍历所有可能的块划分尺寸组合来得到最佳块划分尺寸;而本文算法是为了通过最小的计算复杂度来尽可能准确地预测宏块的划分尺寸,而这种预测又很难通过较低的计算量做到与率失真算法的结果完全一致。所以这些差异也是可以理解的。

然而本文算法却以图像质量和码率方面的微弱牺牲换来了计算复杂度方面的极大降低。在算法速度上,本文算法与 JM7.5c 的率失真优化算法之间的比较结果为对 coastguard 序列,速度平均提高 50.34%,所有序列算法速度平均提高 49.07%。图 6~图 9 分别是 foreman 和 Akiyo 序列在两种算法下各帧数据量,PSNR 值以及计算时间方面的比较。图 9 为 foreman 和 Akiyo 序列在本文提出的算法下相对于率失真算法节省的计算时间的比例。

从图 6、图 7 中可以看出在数据量和 PSNR 值方面,本文算法与 H.264/AVC 中的率失真算法得到的结果基本上是吻合的。从图 8、图 9 中可以看出在算法的计算时间方面,本文算法较率失真算法耗时要小很多。



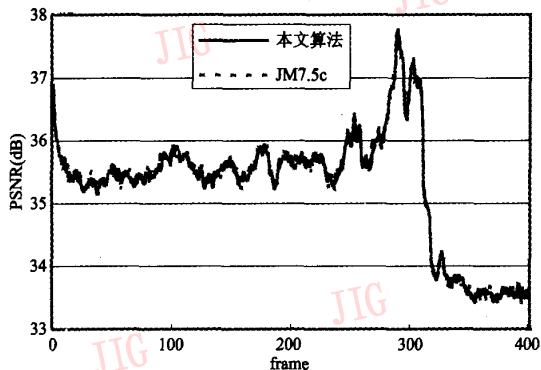
(a) foreman



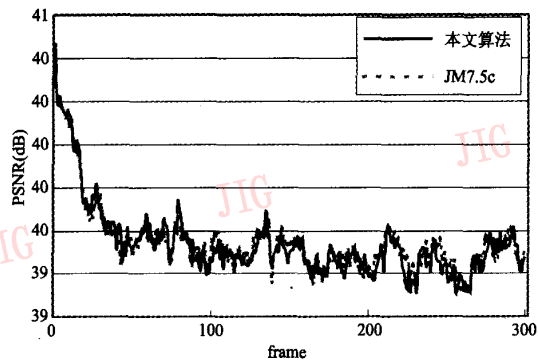
(b) Akiyo

图 6 不同视频序列在两种算法下产生的数据量比较

Fig.6 Data bits comparison for foreman and Akiyo sequence



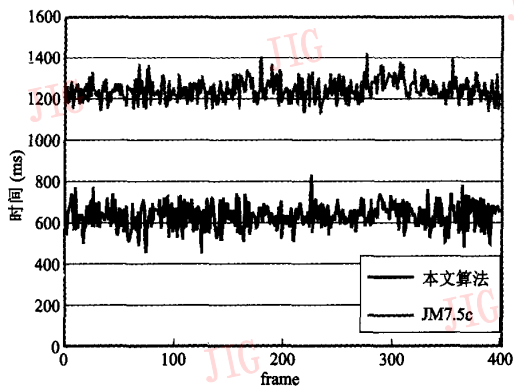
(a) foreman



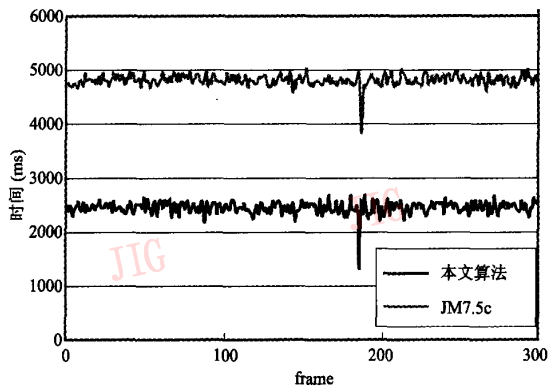
(b) Akiyo

图 7 不同视频序列在两种算法下图像的 PSNR 值比较

Fig.7 PSNR comparison for foreman and Akiyo sequence



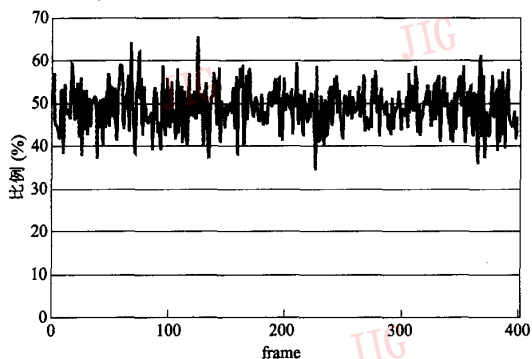
(a) foreman



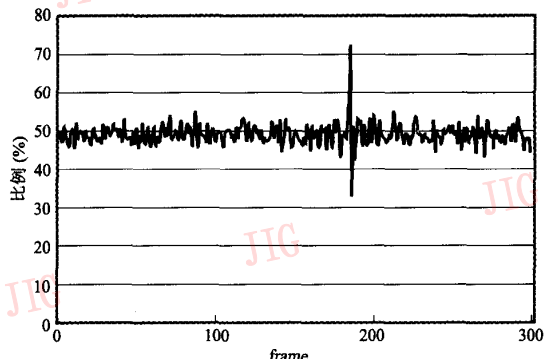
(b) Akiyo

图 8 不同视频序列在两种算法的计算时间比较

Fig.8 Time comparison for foreman and Akiyo sequence



(a) foreman



(b) Akiyo

图 9 不同视频序列在本文算法下节省的时间比例

Fig.9 Time saving of the proposed algorithm for foreman and Akiyo sequence

6 结 论

块划分尺寸选择是 H.264/AVC 提高压缩效率和保证图像质量的重要工具之一。采用率失真优化虽然能得到最佳的块划分尺寸,却需要非常大的计算量。本文针对现有快速块划分尺寸选择算法的不足,提出通过运动搜索预处理,基于运动补偿残差的纹理大小及分布来确定块划分尺寸的快速选择算法。本文算法在数据量、PSNR 值方面与率失真模型的结果基本吻合,而在计算时间上开销却不大,而且算法简单易于实现。从实验结果可以看出本文算法在平均计算速度方面最大提高达到 50.34%,而且计算速度方面的大幅提高只以 PSNR 和码率方面的微小牺牲作为代价。如果本文算法再结合优秀快速运动搜索算法的话,块划分尺寸选择的计算速度将会进一步加快。

参考文献 (References)

1 Ostermann Jorn, Bormans Jan, List Peter, *et al.* Video coding with

H.264/AVC: Tools, performance, and complexity [J]. IEEE Circuits and Systems Magazine, 2004, 4(1): 7~28.

2 Wiegand Thomas, Zhang Xiaozheng, Girod Bernd. Long-term memory motion-compensated prediction for video coding [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1999, 9(2): 70~84.

3 ITU-T JVT-A003, Joint Model Number-1(JM-1)[S].

4 ITU-T JVT-C084, Lagrange Multiplier and RD-characteristics[S].

5 Sullivan Gary J, Wiegand Thomas. Rate-distortion optimization for video compression [J]. IEEE Signal Processing Magazine, 1998, 15(11): 74~90.

6 Chang Andy, Au Oscar C, Yeung Y M. A novel approach to fast multi-block motion estimation for H.264 video coding [A]. In: Proceedings of the IEEE International Conference on Multimedia & Expo. [C]. Baltimore, Maryland, USA, 2003, I: 105~108.

7 Yu Andy C. Efficient block-size selection algorithm for interframe coding in H.264/MPEG-4 AVC [A]. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing [C], Montreal, Quebec, Canada, 2004, III: 169~172.

8 ITU-T JVT-G050, Draft ITU-T recommendation and final draft of Joint Video Specification (ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC) [S].