

一种适用于 AVS 的快速帧间模式选择算法

薛 睿, 苏光大

(清华大学电子工程系, 北京 100084)

摘 要: 音视频编码标准和 H.264 在帧间编码时采用变长块编码, 导致在使用全部模式时计算复杂度较高。为此, 提出一种基于运动向量场同质性测量和宏块时空相关性预测的帧间模式选择算法。利用相邻块和参考帧的对应块判断当前宏块是否使用帧间 16×16 模式, 如果不使用该模式, 则利用帧间 8×8 模式产生的运动向量预测宏块的帧间模式。实验结果证明, 在峰值信噪比几乎相同的情况下, 该算法与全模式相比可平均减少 41.2% 的编码时间。

关键词: 音视频编码标准; H.264 标准; 帧间模式选择; 运动同质性

Fast Inter-frame Mode Selection Algorithm Suitable for AVS

XUE Rui, SU Guang-da

(Department of Electronic Engineering, Tsinghua University, Beijing 100084, China)

【Abstract】 Audio Video coding Standard(AVS) and H.264 achieve significant gain in coding efficiency by adopting variable sizes in inter-frame coding, which leads to high computation complexity when using the full mode decision scheme. This paper presents an inter-frame mode decision algorithm based on the evaluation of the motion homogeneity and the spatial-temporal correlation. The current macro-block 16×16 mode prediction is performed by adopting neighboring macro-blocks and the collocated macro-block in the previous frame. If the prediction fails, the inter-frame mode is predicted by the motion vector, which is generated by using the 8×8 inter-frame mode. Simulation results demonstrate that the algorithm can achieve 41.2% encoding time reduction without any noticeable loss of PSNR.

【Key words】 Audio Video coding Standard(AVS); H.264 standard; inter-frame mode selection; motion homogeneity

DOI: 10.3969/j.issn.1000-3428.2012.02.088

1 概述

音视频编码标准(Audio Video coding Standard, AVS)是我国具有自主知识产权的数字音视频编解码技术标准。近年来, 基于 H.264 的帧间模式快速选择算法已较为成熟, 主要有 2 类方法: 一类是优化模式选择中使用的率优化函数^[1], 另一类是通过模式预测, 忽略当前宏块不太可能选取的模式。AVS 和 H.264 在帧间模式的选择上有很多相似性, 下面介绍一些基于 H.264 的快速帧间模式选择算法。在文献[2]的算法中, 模式选择使用全零块检测技术作为提前终止的条件。在文献[3]中, 对于空间域同质或者时间域静止的宏块, 帧间模式选择只检测 8×8 模式划分以上的模式。空间同质通过使用 sobel 算子计算边缘信息来判断, 时间域静止由当前块和参考帧对应块的差值判定。文献[4]的算法结合了基于率失真的 SKIP 模式提前终止算法和基于运动复杂度的模式分类, 使用 2 个由量化参数决定的门限和 2 个由运动向量决定的门限将宏块帧间模式分为 5 个不同分类。文献[5]使用相位相关函数来捕捉当前帧和参考帧的运动信息, 然后推导出一个直接的模式选择算法。文献[6]的算法首先计算帧间 4×4 模式, 然后根据计算出的 4×4 块的运动向量来决定最优的帧间模式。这样由小宏块的运动向量预测宏块的同质性更加有效和鲁棒。文献[3]根据宏块纹理信息判断宏块同质性, 但并不能检测出纹理块本身有缓慢运动的情况。而根据帧间差值来判断宏块时间同质性同样只适用于具有静态背景的图片。文献[6]的算法则没有考虑时间和空间域宏块帧间模式的相关性。实际的视频序列很少使用帧间 4×4 编码, 因此, 对所有宏块都要先计算帧间 4×4 模式会浪费大量的时间。

AVS 帧间模式的宏块划分最小为 8×8, 帧间模式比 H.264 中的少, 如果在进行模式选择之前对宏块进行各种相位、边缘或者宏块同质性分析, 会带来额外的计算量, 但是可以省略的帧间模式却很有限, 因此, 用于 H.264 的快速模式算法不太适用于 AVS 标准。

本文改进了文献[6]的算法, 将其用于 AVS 标准, 使用 8×8 模式的运动向量进行帧间模式预测。在计算 8×8 模式之前, 先根据周围宏块和前一帧对应宏块的帧间模式进行预测, 如果符合一定的提前终止算法, 直接判断宏块为 16×16 模式, 减少了不必要的模式计算。

2 快速帧间模式选择算法原理

表 1 是部分 CIF 测试视频用全模式进行模式选择, 各个帧间模式所占的比重, 其中, $QP=28$ 。参考代码使用标准工作组提供的 rm52j_r1。

表 1 不同序列模式所占比例 (%)

模式	foreman	city	news	soccer
SKIP	3.60	0.37	17.03	0.93
16×16	65.00	68.16	70.18	66.05
16×8	17.50	15.33	7.24	16.43
8×16	10.40	12.76	3.97	8.53
8×8	3.00	3.36	1.42	6.88
intra	0.50	0.02	0.16	1.18

实验中, 对于更高分辨率的视频或者更大的量化参数 QP , SKIP 模式和 16×16 模式所占的比重将更大。通过观察

作者简介: 薛 睿(1986—), 男, 硕士研究生, 主研方向: 视频处理技术; 苏光大, 教授

收稿日期: 2011-05-16 **E-mail:** xuerui519@163.com

可以发现,原始的视频序列包含很多运动平滑和静态区域,这些区域比较适合大的宏块划分。如果能够准确预测出当前宏块是否适用于 SKIP 或者 16×16 模式,忽略其他不需要帧间模式的判断,将大幅度地减少编码时间。为表述方便,称当前宏块的周围宏块以及参考帧中的对应宏块为时空相关块。近年来的研究表明,在运动平滑和静态区域,当前宏块的帧间模式和时空相关块的帧间模式有很强的相关性。在运动复杂的区域,当前宏块的帧间模式和时空相关块的帧间模式之间的相关性比较弱。换言之,如果当前宏块的时空相关块都是 SKIP 模式或者 16×16 模式,当前宏块很可能也适用于比较大的宏块划分;如果当前宏块的时空相关块的帧间模式比较复杂,当前宏块的帧间模式就很难判断。

本文提出的算法只预测宏块是否适用于 16×16 模式,是基于两方面的考虑:(1)帧间模式选择的计算量主要集中在运动搜索,判断宏块是否适合 SKIP 模式并不需要运动搜索,同时可以利用很多阈值和全零块判别的方法来降低计算量。忽略 SKIP 模式节省的计算量有限,却可能带来多余的预测误差。(2)当前宏块的 16×8、8×16、8×8 模式和时空相关块的帧间模式的关系较难判断,增加这些模式的预测会增加预测函数的复杂度和预测误差。如果宏块不是 SKIP 模式,则预测宏块是否适用于 16×16 模式。宏块需要同时满足时间和空间域的条件才能判定其属于 16×16 模式。

这里提出一个宏块是否可信的问题,如果一个宏块的 16×16 模式是通过预测得到的,那么这个模式本身可能存在一定的预测误差,如果继续用这个帧间模式预测其他宏块的帧间模式,可能会让误差传递下去,造成预测的不准确,从而极大地降低编码效率。所以本文假定,如果一个宏块的 16×16 模式是预测出来的,那么这个宏块称为不可信宏块。在遇到边界时不存在的宏块也称为不可信宏块。这 2 种情况之外的其他宏块称为可信宏块。

时间域的条件:参考帧对应宏块是 SKIP 或 16×16 模式。

空间域的条件:如图 1 所示,E 为当前块,周围的宏块 A、B、C(C 不存在就用 D 代替)中任意 2 个可信宏块是 SKIP 或者 16×16 模式。A、B、C、D 中不存在的宏块设为不可信。

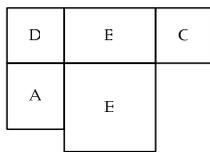


图 1 当前块 E 和相邻块的空间位置关系

如果 SKIP 模式的判断和 16×16 模式的预测都不成立,那么先对宏块进行 8×8 的模式选择,求出 4 个 8×8 宏块的运动向量。以 P 帧单参考帧为例,4 个宏块的运动向量表示为 mvx_i 、 myy_i ($i=0,1,2,3$),行、列和整体的运动同质性用 H 、 V 和 Q 表示:

$$H = 8(|mvx_0 - mvx_2| + |myy_0 - myy_2| + |mvx_1 - mvx_3| + |myy_1 - myy_3|) \quad (1)$$

$$V = 8(|mvx_0 - mvx_1| + |myy_0 - myy_1| + |mvx_2 - mvx_3| + |myy_2 - myy_3|) \quad (2)$$

$$Q = 3|mvx_0 - mvx_1 - mvx_2 - mvx_3| + |3mvx_1 - mvx_0 - mvx_2 - mvx_3| + |3mvx_2 - mvx_1 - mvx_3 - mvx_4| + |3mvx_3 - mvx_0 - mvx_1 - mvx_2| + |3myy_0 - myy_1 - myy_2 - myy_3| + |3myy_1 - myy_0 - myy_2 - myy_3| + |3myy_2 - myy_1 - myy_3 - myy_4| + |3myy_3 - myy_0 - myy_1 - myy_2| \quad (3)$$

按照式(1)~式(3)求出 H 、 V 、 Q ,帧间模式的候选模式根据下面的判决准则分为 3 种情况:

判决准则 1:当 $H < TH1$ 、 $V < TH1$ 、 $Q < TH1$ 时,运动在所有方向上都同质,开启 16×16 模式。

判决准则 2:当 $H > TH1$ 、 $V > TH1$ 、 $Q > TH1$ 时,在所有方向上运动都很复杂,没有体现明显的同质性,开启除 8×8 模式之外所有的帧间模式。

判决准则 3:求出 H 、 V 、 Q 中的最小值 M ,如果 $M=H$,开启 8×16 模式;如果 $M=V$,开启 16×8 模式;如果 $M=Q$,开启 16×16 模式。

根据实验结果, $TH1$ 设为 8,以抵抗噪声带来的影响, $TH2$ 设为 32。

3 适用于 AVS 的快速帧间模式选择算法实现

算法流程如下:

Step1 初始化,关闭所有帧间模式。

Step2 判断宏块是否为 SKIP 模式,如果是,转到 Step6。

Step3 如果参考帧对应模块的帧间模式是 SKIP 或者 16×16,并且周围有 2 个可信的宏块帧间模式是 SKIP 或者 16×16,开启帧间模式 16×16,转到 Step5;如果条件不成立,开启帧间模式 8×8。

Step4 如果开启模式 8×8,则计算 8×8 模式的帧间预测,求出 8×8 宏块的运动向量 mvx_i 、 myy_i ($i=0,1,2,3$),利用式(1)~式(3)计算水平、垂直、整体 3 个方向的运动同质性。根据判决准则 1~准则 3 决定将要开启的候选帧间模式的子集。

Step5 仅在开启的模式上进行率失真函数的计算,选出率失真代价最小的帧间模式作为最优的帧间模式。

Step6 保存最优帧间模式和运动向量,计算结束。

4 实验结果与分析

虽然基于 H.264 的快速帧间模式选择算法很多,但由于 H.264 的帧间模式比 AVS 多,标准和参考代码的选用并不相同,因此不能直接比较。而基于 AVS 的快速帧间模式选择算法较少,且没有被广泛使用和认同的优秀算法,所以,将本文算法直接和全模式进行比较。实验的测试代码是标准工作组提供的 rm52j_r1 的改写代码,称为 rm52j_new。rm52j_new 和 rm52j_r1 编码的码流相同,只是去除了 rm52j_r1 中的一些冗余代码。将 rm52j_new 帧间模式选择部分改写为本文提出的算法,与使用全模式的 rm52j_new 进行对比实验。实验采用 IPPP... 的编码方式,不使用 B 帧,硬件平台和编码器的配置如表 2 所示。

表 2 仿真环境

配置参数	参数值
操作系统	Microsoft Windows XP
CPU	Intel Pentium D 双核, 3.40 GHz, 3.39 GHz
内存	2.00 GB
帧内预测的周期	15
编码方式	场编码
参考帧数目	1
编码帧数	300
运动搜索	全搜索
搜索范围	16
RDO 优化	关闭
其他配置	默认配置

实验结果如表 3 所示,其中, $\Delta PSNR$ 表示亮度信号平均的峰值信噪比的改变量; ΔB 表示全部码流的改变量; ΔT 表示时间的改变量; $\Delta T = (T_{ref} - T_{proposed}) \div T_{ref}$, T_{ref} 是使用全模式

时帧间编码的时间, T_{proposed} 是使用本文算法时帧间编码的时间; “+”表示增加; “-”表示减少。

表3 本文算法和全模式的性能对比

CIF Sequences	QP	$\Delta\text{PSNR}/\text{dB}$	$\Delta\text{B}/(\%)$	$\Delta T/(\%)$
foreman	28	-0.02	+1.02	-35.52
	32	-0.03	+1.32	-40.05
	36	-0.04	+1.87	-42.59
city	28	-0.01	+0.80	-41.59
	32	-0.02	+1.06	-45.81
	36	-0.03	+1.16	-49.17
crew	28	-0.03	+0.97	-32.77
	32	-0.04	+1.35	-36.17
	36	-0.04	+1.62	-40.02
football	28	-0.01	+0.52	-31.31
	32	-0.02	+0.80	-35.71
	36	-0.02	+1.20	-38.78
harbour	28	-0.01	+0.52	-46.72
	32	-0.02	+0.81	-48.92
	36	-0.02	+1.04	-50.65
mother-daughter	28	-0.05	+2.06	-44.38
	32	-0.06	+1.80	-46.20
	36	-0.04	+1.14	-47.53
news	28	-0.03	+0.74	-37.18
	32	-0.04	+0.77	-40.90
	36	-0.05	+1.54	-43.30
soccer	28	-0.02	+0.73	-33.37
	32	-0.03	+1.15	-37.55
	36	-0.03	+1.61	-42.03

实验结果表明, 本文算法对于运动缓慢和运动剧烈的视频都起到了很好的减少编码时间的作用。随着量化参数的增大, 宏块被预测为 16×16 模式的概率也会增加, 使用本文算法节省的时间比率会增大; 除了 mother-daughter 视频, 其他视频增加的码率也有所提高。码率的增加是由于预测或者运动向量场同质性测量不准确造成的, 与 QP 没有直接关联。

5 结束语

本文利用宏块帧间模式在时间和空间的相关性, 以及运

动向量场的同质性判别方法, 提出一种基于 AVS 的快速帧间模式选择算法。对于运动平滑和背景区域, 通过预测可以直接判定 16×16 的编码模式; 对于运动比较复杂的区域, 通过运动同质性测量算法也可以快速确定编码模式的子集, 忽略其他帧间模式的检测。仿真结果表明, 本文算法和全模式相比, 在几乎不损失 PSNR 的情况下, 平均减少了 41.2% 的编码时间, 而平均码流仅增加了 1.15%。

参考文献

- [1] Paul M, Frater M R, Arnold J F. An Efficient Mode Selection Prior to the Actual Encoding for H.264/AVC Encoder[J]. IEEE Transactions on Multimedia, 2009, 11(4): 581-588.
- [2] 周 韬, 张茂军, 刘少华, 等. H.264/AVC 中基于全零块的预测模式选择[J]. 计算机工程, 2009, 35(24): 232-235.
- [3] Wu D, Pan F, Lim K P, et al. Fast Inter-mode Decision in H.264/AVC Video Coding[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2005, 15(7): 953-958.
- [4] Zeng Huanqiang, Cai Canhui, Ma Kai-Kuang. Fast Mode Decision for H.264/AVC Based on Macroblock Motion Activity[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2009, 19(4): 1-11.
- [5] Paul M, Lin Weisi, Lau Chiew-Tong, et al. Direct Inter-mode Selection for H.264 Video Coding Using Phase Correlation[J]. IEEE Transactions on Image Processing, 2011, 20(2): 461-473.
- [6] Liu Zhi, Shen Liquan, Zhang Zhaoyang. An Efficient Inter-mode Decision Algorithm Based on Motion Homogeneity for H.264/AVC[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2009, 19(1): 128-132.

编辑 张帆

(上接第 260 页)

表2 不同 n 时改进型算法运动矢量相比全搜索算法的比例 (%)

序列	$n=1$	$n=4$	$n=8$
Table tennis	76	89	93
Garden	71	87	92
Football	61	80	85
Susie	42	70	80
Celtrain	46	71	81

表3是全搜索算法、传统 BPM 算法(参数 $T_a=8$)和本文改进型 BPM 算法的性能对比。

表3 搜索算法性能比较

序列	测试指标	全搜索算法	传统 BPM 算法	改进型 BPM 算法	变化/(%)
Table tennis	PSNR	28.556 4	25.555 2	26.737 1	4.62
	N_{SAD}	256.00	9.12	4.00	-56.14
Garden	PSNR	25.424 5	24.958 4	25.060 7	0.41
	N_{SAD}	256.00	12.30	4.00	-67.51
Football	PSNR	25.552 3	22.960 6	23.385 4	1.85
	N_{SAD}	256.00	12.31	4.00	-67.51
Susie	PSNR	36.224 6	30.752 5	32.944 7	7.13
	N_{SAD}	256.00	19.81	4.00	-79.81
Caltrain	PSNR	32.464 2	29.626 0	30.448 0	2.77
	N_{SAD}	256.00	22.83	4.00	-82.48

与全搜索算法相比, 改进型 BPM 算法计算 SAD 的块数降低至 4 个; 与传统 BPM 算法相比, 改进型算法计算 SAD 的块数降低 56.14%~82.48%, 运算量明显降低。同时 PSNR 提高 0.41%~7.13%, 运动估计的效果有所提升。

实验结果表明, 改进型 BPM 算法在通过改进阈值的选取方式提高了位平面的区分能力, 与传统 BPM 算法相比, 降低了运算复杂度, 运算量降低 56.14%~82.48%; 同时改进位平面的匹配标准, 即运动向量选择方法, PSNR 提高 0.41%~

7.13%, 增强了运动估计的效果, 改善图像质量。

5 结束语

本文提出的改进型 BPM 运动估计算法简单, 易于控制, 适合于硬件实现。实验结果显示, 与传统 BPM 算法相比, 该算法调整了阈值选取方式, 改进了运动向量计算方法, 明显减少了运算量, 提高了图像质量。

参考文献

- [1] 魏建云, 彭玉华, 刘 微. 一种 AVS 亚像素运动估计快速算法[J]. 计算机工程, 2010, 36(3): 229-231.
- [2] Feng J, Kwok-Tung L, Mehrpour H, et al. Adaptive Block Matching Motion Estimation Algorithm Using Bit Plane Matching[C]//Proc. of IEEE International Conference on Image Processing. Washington D. C., USA: IEEE Press, 1995: 496-499.
- [3] Natarajan B, Bhaskaran V, Konstantinides K. Low Complexity Block-based Motion Estimation via One-bit Transforms[J]. IEEE Transactions on Circuits System and Video Technology, 1997, 7(4): 702-706.
- [4] Wong P. Modified One-bit Transform for Motion Estimation[J]. IEEE Transactions on Circuits System and Video Technology, 1999, 9(7): 1020-1024.
- [5] Erturk A, Erturk S. Two-bit Transform for Binary Block Motion Estimation[J]. IEEE Transactions on Circuits System and Video Technology, 2005, 15(7): 938-946.

编辑 张正兴

