

基于子块残差的快速帧间模式选择算法

李 娜¹, 王维哲²

(1. 解放军信息工程大学电子技术学院, 郑州 450004; 2. 河南职业技术学院信息工程系, 郑州 450046)

摘 要: 针对 H.264/AVC 在提高编码效率和图像质量的同时增加计算复杂度的问题, 通过分析 H.264 的编码区块样式及相邻模式间的相关性, 提出一种帧间模式选择算法。其中包括对编码区块样式中的信息进行分析, 结合相邻模式的相关性, 确定最优模式所处的模式类别, 减少需要计算的模式个数。实验证明, 该算法不需要额外的计算就可以获得 CBP, 在图像性能和码率基本保持不变的情况下, 可以降低约 58% 的编码时间。

关键词: H.264/AVC 技术; 率失真优化; 模式选择; 编码区块样式; 模式分类

Fast Mode Selection Algorithm for Inter Frames Based on Block Residual

LI Na¹, WANG Wei-zhe²

(1. Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004;

2. School of Information Engineering, Henan Polytechnic, Zhengzhou 450046)

【Abstract】 H.264/AVC enhances the coding efficiency and image quality at the cost of a high computational complexity. By analyzing the Coded Block Pattern(CBP) and the correlation of neighboring modes, a fast inter mode selection algorithm is proposed. The information contained in CBP and the correlations of neighboring modes are analyzed to gain the mode classification that the optimization mode belongs to, thus decreases the number of mode need to be calculated. Experimental results show that the coding time reduces about 58% without obvious PSNR and bitrate change.

【Key words】 H.264/AVC technology; Rate Distortion Optimization(RDO); mode selection; Coded Block Pattern(CBP); mode classification

1 概述

H.264 是由 ITU 和 ISO 两大国际标准化组织联合制定的最新的视频编码标准, 与以往标准相比, 在相同编码质量下 H.264 可以节省大约 50% 的码率^[1]。H.264 采用可变块的运动估计和补偿技术, 它引入 7 种用于帧间预测的块模式以及 13 种帧内预测模式和 SKIP/DIRECT 模式来进行帧间预测。

尽管可变块模式选择方式极大地提高了编码效率, 但是它是牺牲编码时间为代价的。对每一个宏块, H.264 需要遍历所有的帧间和帧内编码模式, 使用率失真优化方法(Rate Distortion Optimization, RDO)分别计算该模式下的编码代价 $RDCost$, 选择具有最小的编码代价的模式作为最优编码模式。但是对所有的模式进行判断和选择的方式极大地提高了计算复杂度和编码时间, 实验表明, 在本文实验的硬件配置下使用 Intel VTune 性能分析软件, 对 foreman 的 QCIF 测试序列进行分析, 发现 RD 性能的计算时间占编码总时间的 61.3%。因此, 降低计算复杂度的快速模式选择算法成为时下研究的热点, 为了降低模式选择的计算复杂度, 近年来提出了大量快速帧间模式选择算法^[2-3], 但是这些算法有些增加了额外的计算量, 有些使用自适应阈值不能满足所有的场合。

本文利用子块的编码信息和编码模式的相关性, 提出了一种新的快速模式选择算法, 通过筛选候选模式, 在保证图像质量和码率性能的前提下, 能大幅度地提高模式选择的速度。

2 可变块模式运动补偿

MPEG-4 和 H.263 标准中只采用 16×16, 8×8 这 2 种分块

方式进行预测, H.264 标准中则提供 4 种大宏块分割方式 16×16, 16×8, 8×16, 8×8, 其中, 8×8 又称亚宏块模式, 它可以继续划分为 8×4, 4×8, 4×4 模式。这些分割和子宏块可以使运动物体的划分更加精确, 减少预测误差, 提高编码效率。宏块及亚宏块分割模式如图 1 所示。

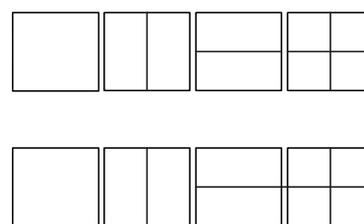


图 1 宏块及亚宏块分割

选择大的分割尺寸, MV 选择和分割类型只需很少的比特, 但是运动补偿残差在多细节区域中的能量很高。选择小的分割尺寸, 运动残差能量低, 但是需要较多的比特来表征 MV 和分割选择。H.264 采用率失真优化(RDO)技术, 选择具有最小编码代价 $RDCost$ 的模式作为宏块的最优预测模式, 如下式所示:

基金项目: 国家“863”计划基金资助项目(2008AA01A315)

作者简介: 李 娜(1982—), 女, 助教、硕士, 主研方向: 网络安全, 多媒体处理; 王维哲, 助教、硕士

收稿日期: 2009-12-18 E-mail: zzulina@163.com

$$J(s, c, MODE | QP, \lambda_{MODE}) = SSD(s, c, MODE | QP) + \lambda_{MODE} \times R(s, c, MODE | QP) \quad (1)$$

其中, s 表示原始块; c 表示重建块; SSD (Sum of Squared Difference)为误差平方和, 表示原始块与重建块差值的平方和; $MODE$ 和 QP 分别代表当前所选的模式和量化参数; R 为熵编码后的码率。 λ 在计算 P 帧时的取值如下所示:

$$\lambda_{MODE} = 0.85 \times 2^{(QP-12)/3} \quad (2)$$

H.264 通过寻找编码后失真和码率总和最小的 $RDCost$ 来确定最优的编码模式, 其步骤如下: 对每一种宏块模式都进行运动搜索, 并采用 RDO 算法分别计算编码代价, 选择编码代价最小的模式进行后续的编码。

3 编码区块样式

在 H.264 宏块层有一个语法元素: 编码区块样式(Coded Block Pattern, CBP), 即由 6 个比特长的二进制串 $b_5b_4b_3b_2b_1b_0$ 中的高 2 位 b_5b_4 表示 2 个色度分量对应的残差编码, 低 4 位 $b_3b_2b_1b_0$ 表示对应的 4 个 8×8 亮度块的残差编码, 变量的最低位比特从最低位开始, 每一位对应于 1 个子宏块, 该位等于 1 时表明对应块残差系数被传送, 该位等于 0 时表明对应块残差全部不被传送, 解码器把这些残差系数赋值为 0。 $b_3b_2b_1b_0$ 全为 1 时表示残差全部编码, 全为 0 时表示全部不编码。 CBP 中各个比特的对应关系如图 2 所示。

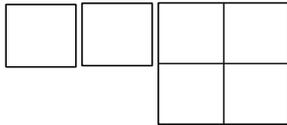


图 2 CBP 中各个比特的对应关系

CBP 的取值可以帮助判定宏块中 4 个 8×8 亮度块及 2 个色度块的残差是否进行编码。如果 CBP 的值为 0 时, 表示 4 个 8×8 亮度块和 2 个色度块都不进行残差编码, 它隐含的信息说明宏块区域中的残差能量很小, 图像的运动很缓慢, 纹理比较简单。此时 SKIP 模式、大块模式为最优模式的比例很大, 对其他模式的分析和计算就可以避免。如果 CBP 中 4 个亮度块的值为全 1 时(此时色度块取值只可能是 00, 01, 10), 则表示宏块区域中纹理比较复杂, 运动信息比较丰富, 此时小块模式为最优模式的比例很大。如果 2 个色度块取值为 00, 4 个亮度块中不全为 0 时, 则表示色度块不进行编码, 只有部分亮度块需要编码, 则所有模式都有成为最优模式的可能。 CBP 处于模式选择之前, 因此可以在不增加额外计算的前提下, 利用 CBP 中包含的信息对模式选择进行筛选, 从而减少模式选择的计算量^[4]。

4 编码区块样式及新算法的提出

4.1 编码区块样式分析

本文算法根据 CBP 取值的不同, 提前筛选掉成为最优模式几率很小的模式, 从而加快模式选择的速度。首先将所有的帧间预测模式划分为 4 种类型, 如表 1 所示。

模式类	包含模式
SKIP 类	模式 SKIP
宏块类	模式 $16 \times 16, 16 \times 8, 8 \times 16$
P8×8 类	模式 $8 \times 8, 8 \times 4, 4 \times 8, 4 \times 4$
所有模式类	包括宏块类、P8×8 类、帧内模式

根据 CBP 的值, 判断最优模式所属的类别:

(1) $CBP=000000,010000,100000(0,16,32)$

此种情况下, 4 个亮度块都不进行编码, 说明运动估计非常准确。因此, 最优模式极大可能为 SKIP 模式和宏块类。

(2) $CBP=000001,000010,000100,001000(1,2,4,8)$

此种情况下, 4 个亮度块中只有 1 个编码, 最优模式极大可能为宏块类。

(3) $CBP=001111,011111,101111(15,31,47)$

此种情况下, 所有的 4 个亮度块都需要编码, 最优模式极大可能为 P8×8 类。

(4) CBP 取除上述 3 种情况外的其他值

此种情况下, 很难从 CBP 的取值信息中判断最优模式, 全部模式都需要计算来获取最优类。

4.2 编码模式相关性分析

这里还使用率失真优化技术对宏块的编码模式类进行判断。在计算所有的帧间预测模式之前, 首先计算 SKIP 模式和 16×16 模式的编码代价, 如果 $J_{SKIP} < J_{16 \times 16}$, 就认为 SKIP 模式为最优模式。然后比较 $16 \times 16, 8 \times 8, 4 \times 4$ 这三种模式的编码代价 $J_{16 \times 16}, J_{8 \times 8}, J_{4 \times 4}$, 如果 $J_{4 \times 4}$ 最大, 就认为宏块类为最优模式; 如果 $J_{16 \times 16}$ 最大, 就认为 P8×8 类为最优模式^[5]。

4.3 基于块残差的快速模式选择算法

本文算法首先将所有帧间预测模式分为 4 种类型, 即 SKIP 模式、宏块类、P8×8 类及所有模式类。先使用 CBP 中包含的信息及对 $16 \times 16, 8 \times 8, 4 \times 4$ 这三种模式的编码代价进行比较来确定该宏块的预测模式所属的预测类型, 然后进行精确的模式选择, 对预测类型中的模式采用率失真优化技术来确定最优的帧间预测模式。具体步骤如下:

(1)取 CBP 的值, 判断是否为 0,1,2,4,8,16,32。由 4.1 节所分析的结果, $CBP=0,16,32$ 时, 最优模式为 SKIP 模式和宏块类的可能性很大。 $CBP=1,2,4,8$ 时, 最优模式为宏块类的可能性很大。然后比较 J_{SKIP} 和 $J_{16 \times 16}$, 若 $J_{SKIP} < J_{16 \times 16}$, 则判定 SKIP 模式为最优模式; 反之, 则判定宏块类为最优模式。

(2)若 CBP 的值为 15,31,47, 此时若 $J_{4 \times 4}$ 为 $J_{16 \times 16}, J_{8 \times 8}, J_{4 \times 4}$ 三者中最小的, 则判定 P8×8 类为最优模式; 反之, 则判定所有模式类为最优模式类。

(3)若 CBP 取其他值, 则说明很难从 CBP 信息中判定最有模式。此时比较 $J_{16 \times 16}, J_{8 \times 8}, J_{4 \times 4}$, 如果 $J_{16 \times 16}$ 最大, 则说明最优模式为 P8×8 类, 若 $J_{4 \times 4}$ 最大, 则说明最优模式为宏块类。

(4)使用率失真优化从预测模式类中选择具有最小编码代价的模式作最优模式。

本文算法的流程如图 3 所示。

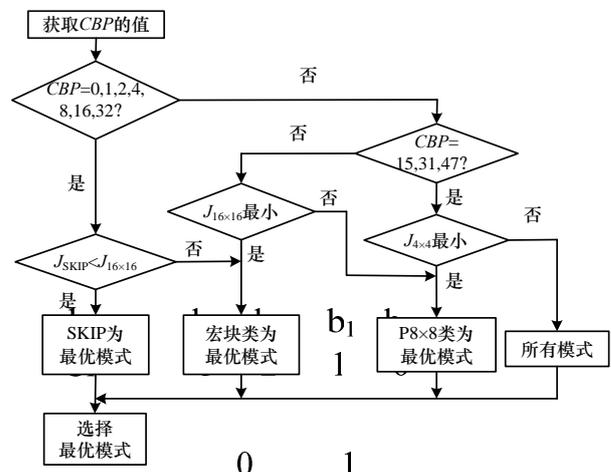


图 3 本文算法流程

5 实验结果

本文在 H.264/AVC 的参考软件 JM10.2 上对以上算法进行了仿真实验, 针对不同运动程度和图像纹理特征的视频序列分别进行测试, 选择的测试序列为 QCIF 格式的 Foreman, News 和 Football。主要的实验参数如下: 使用 Baseline 档次, 参考帧个数为 5, 编码形式为 IPPPP, 搜索范围为 ± 16 , 打开率失真优化选项, 帧率为 30 f/s, 量化参数 QP : 28, 32, 36。

本文的实验结果如表 2 所示, 表中“-”表示与 JM10.2 相比有所降低, 否则表示与 JM10.2 相比有所增高。

表 2 本文算法与 JM10.2 相比的实验结果

测试序列	QP	PSNR/(%)	码率/(%)	编码时间/(%)
Foreman	28	0.024	2.89	-67.21
	32	0.014	2.99	-66.89
	36	-0.004	3.24	-67.05
News	28	0.029	4.26	-68.26
	32	-0.003	3.48	-68.34
	36	0.041	5.49	-67.54
Football	28	0.086	3.12	-32.48
	32	0.064	3.40	-38.51
	36	-0.012	4.80	-47.19

从表 2 中可以看出, 本文算法在计算编码时间上平均降低了 58.2%, 在 PSNR 平均上升 0.026 5% 的同时, 码率平均只增加了 3.74%。实验证明, 本文算法在增加很小计算量的基础上, 提前预测了最优模式的模式类, 在良好的码率控制基础上, 算法极大地提高了编码的速度。

6 结束语

本文针对 H.264 在率失真优化中需要搜索全部的模式, 计算复杂度高, 不利于实时应用等缺点, 提出一种新的快速

帧间模式选择算法。该算法利用 H.264 在计算编码代价 $RDCost$ 之前, 为每个宏块产生的编码区块样式 CBP , 结合相邻模式的相关性, 最终确定最优模式存在的类别, 然后进行精确的编码代价计算, 选择出最优模式。使用 CBP 信息进行预测不增加额外的计算量, 也不使用只适应特定场合的自适应阈值, 可以大大减少参与计算的模式个数。实验结果证明, 本文算法在在码率和视频主观质量损失不大的情况下, 能降低编码时间, 并且适合于不同视频序列。

参考文献

- [1] Wiegand T. Joint Video Team(JVT) of ISO/IEC MPEG and ITU-T VCEG[EB/OL]. (2002-05-10). http://wftp3.itu.int/av-arch/jvt-site/2002_05_Fairfax/JVT-C167.doc.
- [2] Pan Feng, Lin Xiao. Fast Mode Decision for Intra Prediction[EB/OL]. (2003-03-24). ftp://stand-ards.polycom.com/2003_03_Pattaya/JVT-G013.doc.
- [3] 黄晓平, 沈未名. H.264 中与码率控制相结合的快速运动估计[J]. 计算机工程, 2008, 34(12): 192-193.
- [4] Lee J Y, Jeon B W. Fast Mode Decision for H.264[C]//Proc. of 2004 IEEE International Conference on Multimedia and Expo. Taipei, China: [s. n.], 2004: 1131-1134.
- [5] Yin Peng, Tourapis H C, Tourapis A M, et al. Fast Mode Decision and Motion Estimation for JVT/H.264[C]//Proc. of International Conference on Image Processing. Barcelona, Spain: IEEE Press, 2003: 853-856.

编辑 任吉慧

(上接第 223 页)

式(7)为子带合成滤波器中边带合成滤波表达式:

$$N_{ik} = \cos[(16+i)(2k+1)\frac{\pi}{64}] \quad 0 \leq i \leq 63, 0 \leq k \leq 31 \quad (7)$$

对于保存定点数据, 遍历 k 值及 i 值, 需要 $64 \times 32 = 2048$ 个地址空间, 首先根据余弦函数的对称性可以将存储空间缩减到 $2048/4 = 512$ 个内存地址, 再根据实际运算中当 $i=48$ 时, 所有值保持常数-1, 不需要保存这 16 个数据, 所以, 仅需 $512-16=496$ 个地址空间。

5 动态存储空间优化

在 MP3 解码过程中, 需要将每帧的数据解码成 PCM 格式并输出, 同时, 需要保存当前帧的数据留到下一帧使用, 这里采用 DSP 的直接内存访问(DMA)机制^[3]进行数据搬运。DMA 由一种独立于处理器的、能在内存和外存之间直接传输数据的专用控制器实现, 处理器通过提供写入数据地址、读出数据地址、读写数据深度、使能标志等参数设置 DMA, DMA 可以自动读写并发出中断信号。对于 PCM 格式的 1 152 Byte 帧的解码数据进行搬运并存储到外存, 可以节省 1 KB 的内存空间。

对于动态存储空间还可以缩减音频编码数据流缓存, 这一过程对解码结果没有影响, 只需要在解码速度和缓存块大小之间进行折中, 这里开辟 256 个动态内存空间地址存放原始数据流。

根据上述方法对存储空间进行优化, 与之前的占用资源情况进行对比, 表 2 为一些主要数据的优化比较以及最终整体占用资源状况比较, 而文献[4]的存储空间都在 20 KB RAM

的存储空间以上。

表 2 最终占用空间资源比较

查找表存储空间	优化前	优化后
Databuf/Byte	4 096.0	1 024.0
htval/Byte	5 608.0	2 804.0
filter/Byte	8 192.0	992.0
cos/Byte	576.0	32.0
sin/Byte	576.0	120.0
window/Byte	1 024.0	512.0
Total/KB	42.4	15.9

6 结束语

根据 MP3 解码算法的优化实现, 针对不同的 DSP 芯片可以进行不同的优化方案。本文针对定点 DSP 芯片, 在保证解码速率能够满足实时要求的前提下, 对解码过程中占用的内存空间进行统一调配和按模块分配, 最大限度地提高内存占用率, 降低音频解码的资源损耗成本。

参考文献

- [1] ISO/IEC. ISO/IEC 11172-3-94 Information Technology-coding of Moving Pictures and Associated Audio for Digital Storage Media at to About 1.5Mb/s-part3 Audio[S]. 1994.
- [2] Britanak V, Rao K R. An Efficient Implementation of the Forward and Inverse MDCT in MPEG Audio Coding[J]. IEEE Signal Processing Letters, 2001: 48-51.
- [3] Patterson D A, Hennessy J L. 计算机组成与设计: 硬件/软件接口[M]. 郑纬民, 译. 北京: 机械工业出版社, 2007.
- [4] 周锦锋, 高 军. 实现流水方式的并行处理的 MP3 译码器: 中国, 03808024.9[P]. 2007.

编辑 索书志