

自适应阈值的宏块 MAD 快速帧内算法

杨 军, 龚声蓉, 刘纯平

YANG Jun, GONG Sheng-rong, LIU Chun-ping

苏州大学 计算机科学与技术学院, 江苏 苏州 215006

School of Computer Science & Technology, Soochow University, Suzhou, Jiangsu 215006, China

E-mail: shrgong@suda.edu.cn

YANG Jun, GONG Sheng-rong, LIU Chun-ping. Fast adaptive thresholds intra-frame prediction algorithm based on MAD. Computer Engineering and Applications, 2009, 45(34): 189-191.

Abstract: H.264/AVC is the newest video coding standard with high compression efficiency. But it will take a long time to code with extremely complex algorithm, it can not achieve real-time application. In this paper, according to the characteristic of intra-frame prediction algorithm, a fast adaptive thresholds algorithm based on MAD is proposed. This algorithm will predict every macro block before coding, then select between intra 4×4 (I4) and intra 16×16 (I16) prediction modes according to the adaptive thresholds. Then, in I4 prediction modes, select the best mode from its nine prediction modes according to the threshold. Therefore, the complexity of algorithm and coding time are reduced dramatically. Experimental results show that the coding time of proposed algorithm is reduced average by 61.3%, with a little bit increase, and the PSNR is invariable.

Key words: H.264/AVC standard; video coding; intra prediction; Mean Absolute Differences (MAD)

摘 要: 最新的视频压缩标准 H.264/AVC 具有极高的压缩率, 但其算法极其复杂, 编码时间较长, 无法达到实时应用的要求。针对其帧内预测算法的特点, 提出了一种基于 MAD 的自适应阈值快速帧内预测算法。算法充分利用宏块的 MAD (平均绝对误差, Mean Absolute Differences) 信息及时空相关性, 在进行帧内预测之前先对宏块预判, 同时采用自适应阈值的方法在帧内 4×4 (I4) 和帧内 16×16 (I16) 预测模式之间快速进行选择; 然后针对 I4 预测, 采用阈值法在 9 种预测模式间快速选择, 从而减少了算法的复杂度, 提高了压缩速度。实验结果表明, 所提的算法在码率只有少许增加的情况下, 编码时间平均减少 61.3%, PSNR 值基本不变。

关键词: H.264/AVC 标准; 视频编码; 帧内预测; 平均绝对误差

DOI: 10.3778/j.issn.1002-8331.2009.34.059 **文章编号:** 1002-8331(2009)34-0189-03 **文献标识码:** A **中图分类号:** TN919.81

1 引言

H.264/AVC 是由 ISO/IEC MPEG 和 ITU-T VCEG 共同建立的联合视频工作组 JVT (Joint Video Team) 制定的, 于 2003 年 5 月完成。其在视频压缩效率方面比目前所有的视频压缩标准都有显著地提高。H.264/AVC 采用了传统的基于宏块和运动补偿的方案, 同时引入了多模式帧内预测、灵活的宏块分割、更精细像素的运动估计与补偿、率失真优化 (RDO)、环路滤波器新技术。与 MPEG-4 相比, H.264 提高了约 50% 的压缩效率, 但同时增加了几倍甚至几十倍的运算复杂度。其中, 帧内预测和帧间预测是主要的耗时模块^[1-2]。

H.264/AVC 编码标准中, 为了提高帧内编码的效率而引入了帧内预测算法, 只用很少的比特就可以表示该宏块的信息。但是, 帧内预测算法具有极高的算法复杂度, 为了确定一个宏块的帧内预测模式, 需要计算多达 592 种组合模式的代价函

数。正是由于这种计算的高复杂度使它的编码速度变得较慢, 在通常情况下还不能满足实际应用的要求, 所以在不改变 H.264/AVC 标准码流结构和维持原有码率的情况下, 找到简单可行的算法替代原有的复杂算法, 提高其编码速度就成为当前的研究热点。

将利用宏块的 MAD 信息及时空相关性, 结合自适应多阈值的方法, 取代 H.264/AVC 原有的判断宏块编码模式的方法, 对于特征明显的宏块, 直接从两种预测模式中选取一种, 从而避免对每一个宏块都做两种模式的预测, 提高了编码速度。对于那些特征不是很明显的宏块, 采用标准算法的方法进行判断, 以保证用最佳的模式来编码。这样既能提高编码速度, 又能将码率的增加控制在极小的范围内。同时在 Intra 4×4 的 9 种预测模式中利用阈值判断的方法进行快速选择, 进一步提高了编码速度。

基金项目: 国家自然科学基金 (the National Natural Science Foundation of China under Grant No.60673092); 教育部科研重点项目 (The Ministry of Education Key Science Research Project No.205059); 江苏省高校自然科学研究计划项目 (the Natural Science Research Project of Higher Education of Jiangsu Province, China under Grant No.07KJD520186)。

作者简介: 杨军 (1983-), 男, 硕士研究生, 主要研究方向为图像与视频处理、视频编码与传输等; 龚声蓉, 男, 教授, 主要研究方向: 数字媒体与数字技术, 包括图像与视频处理、数字媒体中的信息隐藏技术、模式识别等。

收稿日期: 2008-07-21 **修回日期:** 2008-10-23

2 H.264/AVC 帧内预测分析

2.1 预测模式

帧内预测是 H.264/AVC 编码标准中 I 帧所采用的预测方法。H.264/AVC 利用了相邻像素的相关性,采用了新的帧内预测模式。通过当前像素块的左边和上边的像素(如图 1)进行预测,只对实际值和预测值的差值进行编码,这样就能用较少的比特数来表达帧内编码的像素块信息。在当前的标准中,亮度块可以有 9 种 4x4 块的 Intra 预测模式,通常称为 Intra 4x4 模式,适用于带有大量细节的图像编码;另有 4 种 16x16 块的 Intra 预测模式,通常称为 Intra 16x16 模式,适用于平坦区域图像编码。图 2 说明了 Intra 4x4 的 9 种预测方向,表 1 是其相关说明。图 3 是 Intra 16x16 预测的 4 种预测模式,表 2 是其相关说明^[3]。

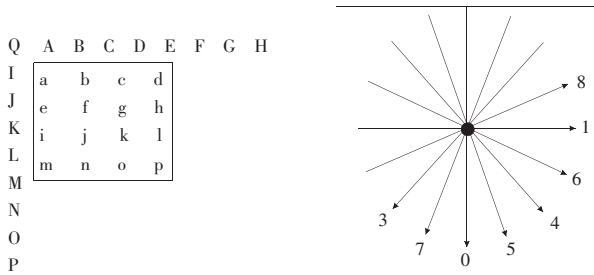


图 1 预测像素

图 2 Intra 4x4 的预测方向

表 1 Intra 4x4 预测模式说明

模式	描述
模式 0(垂直)	由 A、B、C、D 垂直推出相应像素值
模式 1(水平)	由 I、J、K、L 水平推出相应像素值
模式 2(DC)	由 A~D 及 I~L 平均值推出所有像素值
模式 3(下左对角线)	由 45°方向像素内插得出相应像素值
模式 4(下右对角线)	由 45°方向像素内插得出相应像素值
模式 5(右垂直)	由 26.6°方向像素值内插得出相应像素值
模式 6(下水平)	由 26.6°方向像素值内插得出相应像素值
模式 7(左垂直)	由 26.6°方向像素值内插得出相应像素值
模式 8(上水平)	由 26.6°方向像素值内插得出相应像素值

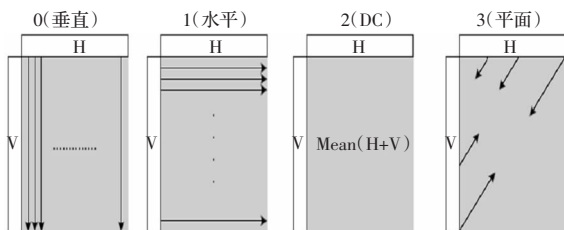


图 3 Intra 16x16 预测模式

表 2 Intra 16x16 预测模式说明

模式	描述
模式 0(垂直)	由上边像素推出相应像素值
模式 1(水平)	由左边像素推出相应像素值
模式 2(DC)	由上边和左边像素平均值推出相应像素值
模式 3(平面)	利用线形“plane”函数及左、上像素推出相应像素值,适用于亮度变化平缓区域

2.2 帧内预测分析

H.264/AVC 编码标准使用率失真优化 (Rate Distortion Optimization, RDO)方法对每个宏块遍历各种预测模式,并由式(1)确定每种模式的率失真代价,选择最小的代价对应的帧内预测模式作为最佳预测模式。

$$J(s, c, m|QP, \lambda_m) = SSD(s, c, m|QP) + \lambda_m * R(s, c, m|QP) \quad (1)$$

式(1)中, QP 是宏块的量化参数, λ_m 是拉格朗日乘子, $SSD(\cdot)$ 是原始的块 s 和重建块 c 之间的误差平方和, 计算如公式(2)。 $R(\cdot)$ 是利用所选择的模式 m 进行编码的比特数。

$$SSD = \sum_{i,j} [s(i,j) - c(i,j)]^2 \quad (2)$$

H.264/AVC 为了获得最好的预测结果,对每个宏块同时进行 Intra 16x16 预测和 Intra 4x4 预测, 比较最后编码后的结果,再决定使用哪种预测方式。然而帧内预测中,亮度分量有 4x4 和 16x16 两种分块方式, 它们分别有 9 和 4 种预测模式; 8x8 色度块有 4 种预测模式。因此, 对于一个宏块组合共有 $4 \times (9 \times 16 + 4) = 592$ 种不同的模式组合, 需要进行 592 次 RDCost 计算, 才能得出最佳的帧内预测模式。这样, 算法复杂度大大提高, 压缩速度也就相应变的很慢。因此如何减小帧内预测算法的复杂度, 提高编码速度成为迫切需要解决的问题。

目前, 减小帧内预测复杂度的方法主要有两类: 一类是缩小预测模式的选择范围, 另一类是简化代价函数。第一类方法, 可以利用当前块及其周围像素的某些特征, 预先排除某些可能性很小的预测模式, 或提前终止某些可能性小的模式的代价计算, 从而降低帧内预测的复杂度。为了获取图像的某些特征, 可以直接在空间域上进行分析, 也可以在变换域上进行分析。文献[4]提出了利用前帧对应位置宏块和当前宏块的相关性, 跳过一些宏块的模式选择过程, 减少 33% 的计算复杂度, 但是 PSNR 下降较多, 约为 0.2 dB。该算法只考虑了宏块级的时间相关性, 忽略了空间相关性及 4x4 子块级。文献[5]提出了利用相邻方向的编码模式率失真性能接近的特点, 减少了约 30% 的编码时间。文献[6]提出了采用二维直方图方法判断宏块预测模式, 编码速度提高 16.26%, 但是算法复杂度较高, 不利于实现。文献[7]提出了利用边缘方向柱状图以及 sobel 算子, 排除一些可能小的模式, 大大减少了编码时间, 但是码率增加 3% 以上, PSNR 下降 0.2 dB 以上。文献[8]提出了利用相邻帧对应位置宏块之间的关系及基于量化因子的修正帧, 编码速度提高了 22%~35%, 但是码率增加大约 3%。文献[9]属于第二类, 提出了新的简化的模型来提高编码速度。

采用基于 MAD 的自适应阈值方法, 提前判断宏块采用的预测模式, 并在 14 的 9 种预测模式中进行快速选择, 算法复杂度明显降低, 很大的提高了编码速度。

3 基于 MAD 的自适应阈值快速帧内预测算法

3.1 宏块 MAD 计算

对于图像平滑区域的宏块, 适合采用 Intra 16x16 预测模式, 而对于图像细节运动比较丰富区域的宏块, 适合采用 Intra 4x4 预测模式。目前有三种方法描述图像平滑程度的信息: MAD 方法、直方图法和 MSE (Mean Square Errors, 均方误差) 方法。直方图法统计一幅图像中各个灰度级出现的次数和频率, 以此反映图像细节的丰富程度。该方法计算比较复杂, 不利于硬件实现。MSE 方法的计算方法如公式(3):

$$MSE = \frac{1}{16 \times 16} \sum_{y=0}^{15} \sum_{x=0}^{15} |p(x,y) - m|^2 \quad (3)$$

式中: $p(x,y)$ 是坐标为 (x,y) 处的像素值; m 是这个宏块所有像

素的平均值。由于采用了平方运算,对浮点运算和设备要求比较高,因此也不利于实现。MAD 法利用图像平滑区域的 MAD 值要远小于图像细节变化较快的区域,采用阈值法预先判断采用 I4 模式还是 I16 模式,编码速度大大提高,且 MAD 方法只涉及到加法和移位操作,计算简单,利于实现。

宏块的 MAD 定义如公式(4):

$$MAD = \frac{1}{16 \times 16} \sum_{y=0}^{15} \sum_{x=0}^{15} |p(x,y) - m| \quad (4)$$

式中: $p(x,y)$ 是坐标为 (x,y) 处的像素值; m 是这个宏块所有像素的平均值。

3.2 Intra4x4 快速预测模式算法 FTIntra4

Intra4x4 模式共有 9 种预测方向,H.264/AVC 采用穷举的方法遍历每一种预测方向,经过对实际视频序列进行测试发现:在 Intra4x4 预测的 9 种模式中,水平、垂直、DC(平均)三种预测方向占有实际预测方向中的大部分,如表 3 所示。

表 3 三种预测方向在序列中所占比例

序列名	三种方向所占比例/(%)
Foreman	48.1
Missa	62.2
Mother_And_Daughter	60.9
News	58.5
Coastguard	68.7
Salesman	58.5
Slient	75.7
Mobile	69.5
Container	66.2

因此,对于大多数的子块预测,在前面的预测方向中已经找到了最佳的预测方向,后面的预测就显得多余了。在此基础上提出了一种新的算法 FTIntra4,通过设定阈值的方法优先判断采用哪种预测模式,具体步骤:

(1)首先计算水平、垂直、DC 预测方向,用率失真公式从其中选择最佳的方向,将结果 RDcost 与事先设定的经验阈值 Tcost 比较。

(2)If $RDcost \leq Tcost$,则将此方向作为最后的预测方向; If $RDcost > Tcost$,跳到第(3)步。

(3)计算其他的预测方向,每计算一个方向,就与刚才选择的最佳值比较,同时与阈值 Tcost 比较,如果同时小于两个值,则将其作为最后的预测值,如果不满足条件则继续,直到计算完所有预测方向。

(4)计算完毕,得到最佳预测方向。

3.3 基于 MAD 的自适应阈值快速帧内预测算法

基于 MAD 的自适应阈值快速帧内预测算法(FATMAD)实现步骤如下:

(1)计算第一帧内各个宏块的 MAD,并采用率失真优化技术判断各个宏块的帧内预测方式。找出这样的宏块 MAD 值 MAD_i, MAD_j :

对该帧内的任何某宏块 x ,如果 $MAD_x < MAD_i$,则其帧内预测方式为 I16 模式。如果 $MAD_x > MAD_j$,则其帧内预测方式为 I4 模式。

令 $T_{min} = MAD_i, T_{max} = MAD_j$ 。

(2)对后面的每个 I 帧,计算所有宏块的 MAD 值,用其 MAD 和之前设定的两个阈值 T_{min} 和 T_{max} 比较:

If $MAD \leq T_{min}$,则该宏块采用 I16 帧内预测;

If $MAD \geq T_{max}$,则该宏块采用 I4 帧内预测,预测方法采用 FTIntra4 算法。

否则,同时进行 Intra 16x16 预测和 Intra 4x4 预测,Intra 4x4 预测采用 FTIntra4 算法,比较最后编码后的结果,再决定使用哪种预测方式。

一帧处理完后,依照第一步的方法更新 T_{min}, T_{max} 。

(3)所有帧处理完毕,结束。

4 实验结果与分析

针对提出的快速帧内预测算法 FATMAD,在 H.264/AVC 标准测试模型 JM13.2 上进行了测试,一共选取了 4 个运动程度不同的 QCIF 格式的测试序列 Foreman, Silent, Claire 和 News,每个序列取前 100 帧进行测试,编码方式全部采用 I 帧。实验结果以表格形式给出,其中 O 代表原 H.264/AVC 标准算法,N 代表 FATMAD 算法, $\Delta SNRY$ 代表 SNRY 的变化, ΔBR 代表比特率的变化, ΔT 表示编码时间的变化,+表示增加,-表示减少。

实验硬件环境: Intel 奔腾处理器 2.8 GHz, 512 M 内存。软件环境: VC++ 6.0。

表 4 实验结果

序列	算法	SNRY	$\Delta SNRY$	BR	ΔBR	T/s	ΔT
		/dB	/dB	/(kbit·s ⁻¹)	/(%)		/(%)
Foreman	O	37.10	-0.01	696.09	+0.13	63.722	-59.10
	N	37.09		696.99		26.062	
Silent	O	34.47	-0.02	863.58	+0.64	65.806	-68.56
	N	34.45		808.72		20.689	
Claire	O	40.43	-0.01	387.39	+0.35	55.742	-50.04
	N	40.42		388.75		27.849	
News	O	37.75	-0.03	813.56	+0.30	66.304	-67.50
	N	37.72		816.00		21.549	

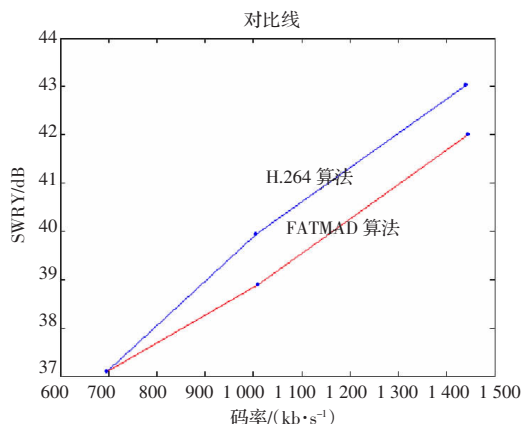


图 4 Foreman 序列码率-SNRY 对比图

从实验结果看出,该文算法相对 H.264/AVC 标准算法在 PSNR 方面基本不变,编码时间平均减少 61.3%,码率平均增大 0.355%,从不同序列所对应的数据中可以看出,无论是对于大量采用 Intra 16x16 模式的序列如 Claire,还是较多采用 Intra 4x4 模式的序列如 Foreman,该文的算法对帧内编码速度都能有较大的提高,同时码率增加的范围都很小。因此,所提算法具有理论意义和实际应用价值。