第 34 卷 第 19 期 计 算 机 工 程 2008 年 10 月
Vol.34 No.19 Computer Engineering October 2008
·多媒体技术及应用· 文章编号：1000—3428(2008)19—0228—02 文献标识码：A 中图分类号：TP301.6

改进的 H.264 快速帧内预测模式选择算法

王 慧，常建平
(南京航空航天大学信息科学与技术学院，南京 210016)

摘 要：为了减小 H.264 编码算法的复杂度，提出一种快速的帧内模式选择算法。根据宏块内部相邻像素差的特点判定宏块预测类型，对基于边缘方向直方图的 Pan 算法进行改进，利用改进的算法对选定了预测类型的宏块进行预测，选出最佳的预测模式。实验结果表明，该算法在保证失真率和码率性能的前提下，编码时间平均减少了 71.6%，大大提高了编码效率。

关键词：H.264 标准；帧内预测；预测模式选择

Improved Fast Intra Prediction Mode Decision Algorithm for H.264

WANG Hui, CHANG Jian-ping
(College of Information Science & Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016)

【Abstract】 In order to reduce the complexity of H.264 encoding algorithm, this paper proposes a fast mode decision algorithm for intra prediction. The prediction type of a macroblock is decided according to the differences between the neighboring pixels of a macroblock. Pan's fast algorithm based on edge direction histogram is improved to predict the macroblock; therefore the best prediction mode is chosen. Experimental results show that the method can save the encoding time up to 71.6% averagely when guarantee the capability of the rate distortion and the bits rate, and it improves the encoding efficiency tremendously.

【Key words】 H.264 standard; intra prediction; prediction mode decision

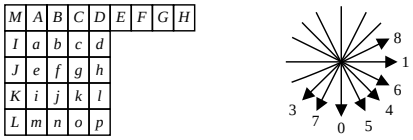
1 概述

新一代视频压缩标准 H.264(或 MPEG4\AVC)是由 ISO/IEC MPEG 和 ITU-T VCGE 共同建立的联合视频工作组(JVT)制定的。它的视频压缩效率相比目前所有视频压缩标准都有显著的提高，但压缩效率的提高是以增加计算复杂度为代价的，这使得 H.264 很难应用于实时性要求较强的场合。

H.264 采用了许多新技术[1-2]，其中，帧内预测编码是消除空间冗余的关键技术。它利用已解码的相邻块像素对当前块像素从多个方向进行多种模式帧内预测。目前，对帧内预测算法的研究和改进大多集中在减少候选预测模式个数方面[3-4]，在预测前先判断宏块预测类型的方法较少。本文首先根据宏块内部相邻像素差值的特点判断宏块采用帧内 4×4 预测还是帧内 16×16 预测，然后对文献[4]中基于边缘方向直方图的算法进行改进，提出一种高效的快速帧内预测模式选择算法。

2 H.264 帧内预测

在 H.264 帧内预测中，当前宏块先通过其左边和上边宏块的相邻像素进行预测，然后对当前宏块和预测宏块之间的差值进行变换、量化和熵编码。其中，亮度分量有 2 种预测类型：4×4 预测(9 种模式)和 16×16 预测(4 种模式)；色度分量有 4 种 8×8 预测模式。图 1 给出了 4×4 块的帧内预测示意图，包括图 1(b)所示的 0~1, 3~8 等 8 种方向预测模式以及直流模式 2。图 1(a)中 a~p 的值可用相邻块的 A~M 来预测。而亮度 16×16 块和色度 8×8 块仅支持垂直、水平、直流和平面模式。H.264 采用 RDO 判决准则，利用拉格朗日代价函数选出最优的预测模式。它以色度模式为外循环，依次扫描亮度的所有模式，每一个宏块共需要进行(9×16+4)×4=592 次 RDO 计算。



(a)4×4 图像分布 (b)9 种预测方向模式示意图

图 1 4×4 块帧内预测

3 改进的快速帧内预测模式选择算法

3.1 基于相邻像素差值的宏块类型判决

16×16 预测模式是以整个宏块为处理对象，适用于比较平坦的宏块；而 4×4 预测模式则把一个宏块分为 16 个子块，再对子块分别处理，适用于细节较多的宏块。本文使用图像相邻像素差值表征宏块细节丰富程度的特征。

一幅图像中相邻像素差值较大的数量越多，该图像的细节就越多；相邻像素差值较小的数量越多，该图像的细节就越少。本文通过统计确定表征图像细节多少的 3 个阈值 $T1$, $T2$, N 来提前决定最佳预测块的类型。理论上，一个像素和周围像素点的像素差值有 8 个，但为了简化计算，本文只考虑水平和竖直方向的差值，这对判决结果几乎无影响。如果相邻像素差值大于 $T1$ 的个数大于 N，则认为该宏块细节较多，采用 4×4 预测模式；如果相邻像素差值小于 $T2$ 的个数大于 N，则认为宏块平坦，采用 16×16 预测模式；不满足以上 2 种情况时，采用 H.264 的标准算法。这样可以为特征明显的宏块直接选择一种模式，减小运算复杂度；对特征不明显的宏块采用标准方法，严格控制码率的增加。算法流程如图 2 所示。

作者简介：王 慧(1984－)，女，硕士研究生，主研方向：图像处理，计算机视觉；常建平，副教授

收稿日期：2007-11-20 **E-mail：**wanghui823@163.com

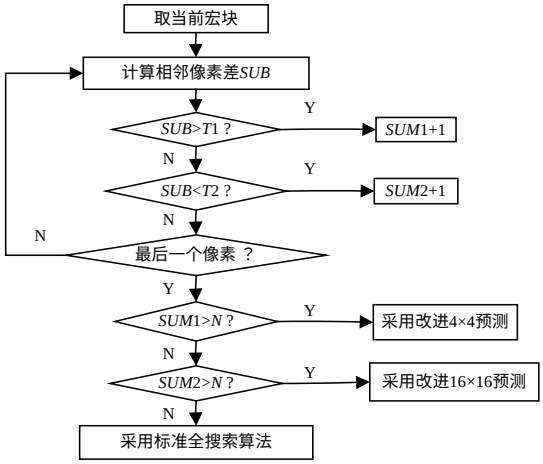


图 2　改进算法的流程

选择 4×4 预测模式还是 16×16 预测模式，其关键是阈值 *T*1, *T*2, *N* 的确定。本文通过一段嵌在 JM90 模型里的程序进行统计试验。先给定 *T*1, *T*2 的值，然后统计出采用 4×4 预测的宏块中相邻像素差值大于 *T*1 的个数 *SUM*1 和采用 16×16 预测的宏块中相邻像素差值小于 *T*2 的个数 *SUM*2。给定不同的 *T*1, *T*2 值对几个序列进行大量的实验，统计得出，当阈值 *T*1=3, *T*2=3, *N*=260 时，预测效果比较好。

3.2　改进的基于边缘方向模式选择

文献[4-5]提出了基于边缘方向直方图的快速帧内预测模式选择算法。该算法首先用 Soble 算子计算每个像素点的边缘方向矢量，然后利用边缘方向矢量求得每一个块的边缘方向直方图，最后通过该直方图的分布特点，选出可能性较大的几个模式。算法中定义边缘矢量 $\boldsymbol{D}_{i,j}=\{dx_{i,j},dy_{i,j}\}$ 为

$$dx_{i,j}=p_{i-1,j+1}+2\times p_{i,j+1}+p_{i+1,j+1}-p_{i-1,j-1}-2\times p_{i,j-1}-p_{i+1,j-1}$$

$$dy_{i,j}=p_{i+1,j-1}+2\times p_{i+1,j}+p_{i+1,j-1}-p_{i-1,j-1}-2\times p_{i-1,j}-p_{i-1,j+1}$$

其中，$dx_{i,j}$ 和 $dy_{i,j}$ 分别表示垂直和水平方向的分量。

本文算法先根据 Soble 算子求得每个像素点的边缘方向矢量，再计算每个块中所有像素点边缘方向矢量的水平方向之和 $\sum dy$ 和垂直方向之和 $\sum dx$，然后根据 $(\sum dy/\sum dx)$ 确定可能的预测模式。这样相对于 Pan 算法可以大大减少计算量。定义一个块的边缘方向角为

$$\alpha=\frac{180^\circ}{\pi}\arctan(\sum dy/\sum dx),\ |\alpha|<90^\circ$$

根据每个块的边缘方向角所在范围确定该块可能的预测模式。如果已判断出一个宏块预测类型为亮度 4×4 预测，则根据 α 选择预测模式，见表 1。对于亮度 16×16 预测和色度 8×8 预测，也根据块的边缘方向角选择候选预测模式(两者都选出与块方向角对应的预测模式和 DC 模式)。本文算法和参考模型的全搜索算法复杂度比较见表 2。

表 1　预测模式选择

α 范围	候选预测模式
(-103.3°, -76.6°]	0,2,5,7
(-13.3°, 13.3°]	1,2,6,8
(35.8°, 54.2°]	3,2,7,8
(-54.2°, -35.8°]	4,2,5,6
(-76.7°, -54.2°]	5,2,0,4
(-35.8°, -13.3°]	6,2,1,4
(54.2°, -76.7°]	7,2,0,3
(13.3°, 35.8°]	8,2,1,3
其他	2

表 2　本文算法和参考模型的全搜索算法 RDO 计算次数

算法	宏块判为 4×4 预测	宏块判为 16×16 预测	未判出时
本文	(16×4)×2=128	2×2=4	(16×4+2)×2=132
全搜索		(16×9+4) ×4=592	

4　实验结果与分析

将本文的快速算法应用到 JM90 测试模型中在 VC6.0 平台上进行实验，并和 JM90 及 Pan 算法在编码时间 *Time*、输出码率 *Bits*、峰值信噪比 *PSNR* 3 方面进行比较。*PSNR* 分为一个亮度分量 *Y*、2 个色度分量 *U* 和 *V*。实验环境是 CPU 为 1.7 GHz、内存为 512 MB 的计算机。实验参数为：允许 RDO，允许 hardarmard 变换，熵编码采用 CABAC，Transform8×8Mode 被设置为 0。实验取 3 个具有代表性的 QCIF 测试序列：低速运动的 claire，快速运动的 foreman，纹理细节较多的 mobile。每个序列取前 100 帧，采用全 I 帧编码方式。表 3、表 4 分别给出 *QP*=28, 32, 36 时本文算法相对于 JM90 和 Pan 算法的结果比较。

表 3　本文算法相对于 JM90 的结果比较

QP	比较参数	Claire	foreman	mobile
28	Δ *Time*/(%)	-72.33	-71.70	-71.98
	Δ *Y*/dB	-0.09	-0.01	-0.20
	Δ *U*/dB	+0.20	-0.05	-0.02
	Δ *V*/dB	-0.03	-0.02	-0.02
	Δ *Bits*/(%)	+14.01	+15.24	+12.60
32	Δ *Time*/(%)	-72.44	-71.77	-70.15
	Δ *Y*/dB	-0.13	-0.05	-0.17
	Δ *U*/dB	+0.01	-0.04	-2.19
	Δ *V*/dB	-0.04	-0.07	-1.70
	Δ *Bits*/(%)	+13.53	+16.11	+17.02
36	Δ *Time*/(%)	-72.32	-71.61	-71.39
	Δ *Y*/dB	-0.14	-0.17	-0.18
	Δ *U*/dB	+0.32	-0.09	-0.01
	Δ *V*/dB	-0.16	-0.12	-0.03
	Δ *Bits*/(%)	+19.10	+13.18	+15.25

表 4　本文算法相对于 Pan 算法的结果比较

QP	比较参数	Claire	foreman	mobile
28	Δ *Time*/(%)	-11.35	-10.73	-19.42
	Δ *Y*/dB	-0.12	-0.01	-0.13
	Δ *U*/dB	+0.09	0.00	0.00
	Δ *V*/dB	+0.13	-0.01	0.00
	Δ *Bits*/(%)	-0.94	-0.98	-1.85
32	Δ *Time*/(%)	-10.11	-16.52	-14.30
	Δ *Y*/dB	-0.10	-0.05	-0.13
	Δ *U*/dB	+0.03	0.00	-1.01
	Δ *V*/dB	+0.02	-0.04	-0.52
	Δ *Bits*/(%)	-0.58	-1.21	+1.30
36	Δ *Time*/(%)	-13.13	-11.91	-12.04
	Δ *Y*/dB	-0.10	-0.13	-0.11
	Δ *U*/dB	+0.16	-0.03	-0.01
	Δ *V*/dB	+0.14	-0.05	-0.01
	Δ *Bits*/(%)	+0.01	-0.56	-1.17

实验数据采用大量测试取平均值的处理方法，其中，负值数据表示减小；正值数据表示增加。可以看出，本文算法和 JM90 相比，编码时间减少了 71%~72%，*PSNR* 变化很小，对人的视觉来说几乎不影响图像质量；本文算法与 Pan 算法相比，编码时间减少达 10%~20%，*PSNR* 变化很小。另外，本文算法产生的码率相对于 JM90 有些增加，但和 Pan 算法相比，码率有所减少。

（下转第 241 页）