

文章编号:1001-9081(2006)10-2383-03

## 一种改进的快速 H. 264/AVC 帧内预测 Pan 算法

田晓冬, 田裕鹏

(南京航空航天大学自动化学院, 江苏南京 210016)

(superpumer@126.com)

**摘要:**对基于边缘方向直方图的快速帧内预测 Pan 算法进行了改进,应用了帧内 Intra\_4×4 和 Intra\_16×16 块类型选择以及 4×4 亮度块模式选择提前中止技术。实验结果表明,该方法在保证图像失真度和码率性能的前提下,单帧编码时间较 Pan 算法减少了 29.093%,较大幅度地加快了帧内模式决策速度。

**关键词:**H. 264/AVC; 帧内预测; Pan 算法; MPEG; 视频编码

**中图分类号:** TP37; TP391.41 **文献标识码:** A

### Improved Pan algorithm for fast intraprediction in H. 264/AVC video coding

TIAN Xiao-dong, TIAN Yu-peng

(College of Automation, Nanjing University of Aeronautics and Astronautics, Nanjing Jiangsu 210016, China)

**Abstract:** Fast Intraprediction Pan algorithm based on edge directional histogram was improved in this paper. The block type selection method (Intra\_4×4 and Intra\_16×16) and early termination technology for 4×4 luma block mode decision were employed. Experimental results show that the one-frame scramble time of fast intraprediction algorithm, compared with Pan algorithm, is reduced by 29.093% with negligible loss of peak signal-to-noise ratio.

**Key words:** H. 264/AVC; intraprediction; Pan algorithm; MPEG; video coding

## 0 引言

H. 264/AVC<sup>[1]</sup> 是由 ISO/IEC MPEG (Moving Picture Experts Group) 和 ITU-T VCEG (Video Coding Experts Group) 联合制定的最新的视频编码标准,具有高压缩比、高图像质量和友好的网络特性,适用于交互和非交互应用环境。在相同重构图像质量下,与 H. 263+ 和 MPEG-4 ASP 标准相比,H. 264/AVC 能节约 50% 的码流。为了提高编码性能,H. 264/AVC 采用了许多先进的编码技术<sup>[2]</sup>,其中帧内预测编码是消除空间冗余的关键技术。

帧内预测编码是指利用相邻宏块的空间相关性进行预测编码,以提高帧内编码的效率。对于亮度分量,H. 264/AVC 对含有较多空域细节信息的宏块采用 4×4 预测(9 种模式),而对于较平坦的区域采用 16×16 的预测模式(4 种模式);对于色度分量有 4 种 8×8 的预测模式,其预测方向和 16×16 亮度块预测模式一致(仅模式号不同)<sup>[3]</sup>。为了获得最优的编码性能,H. 264/AVC 采用了率失真优化(Rate Distortion Optimization, RDO)技术,比较各种模式组合的性能。为了确定一个宏块 MB (Macroblock) 的帧内预测模式,需要计算 592 种组合模式的 RD 代价(RD\_Cost)<sup>[4]</sup>,从而使编码器的复杂度大幅度增加,很难应用于实时性要求较强的场合,因此研究 RDO 模式下的快速帧内预测模式选择算法具有重要应用价值。

本文对快速帧内预测 Pan 算法进行了改进,实验结果表

明,该方法在保证图像失真度和码率性能的前提下,较大幅度地加快了帧内模式决策速度。

## 1 基于边缘方向信息模式选择的 Pan 算法

Pan 算法<sup>[5]</sup>的基本思想:由于物体边界通常是连续的,沿边缘方向的像素往往具有相似的值。因此可以事先获得 N×N 块的边缘方向信息,并使预测方向和这个边缘方向相同时,可获得最优或次优的帧内预测。其过程如下:

1) 计算图像的边缘方向信息

获得局部边缘方向信息的方法很多,其中 Sobel 算子是常用的边缘检测算子。它有两个卷积核,一个反映垂直方向的变化程度,另一个反映水平方向的变化程度。当原始的视频图像分别与 Sobel 算子的两个卷积核卷积计算后,对像素  $P_{i,j}$  ( $P_{i,j}$  不在图像边界上) 产生相应的边缘矢量  $\vec{D}_{i,j} = \{dx_{i,j}, dy_{i,j}\}$ 。  $dx_{i,j}$  和  $dy_{i,j}$  分别表示垂直和水平的变化强度,定义如下:

$$\begin{aligned} dx_{i,j} &= P_{i-1,j+1} + 2 \times P_{i,j+1} + P_{i+1,j+1} - P_{i-1,j-1} - 2 \times P_{i,j-1} - P_{i+1,j-1} \\ dy_{i,j} &= P_{i+1,j-1} + 2 \times P_{i+1,j} + P_{i+1,j+1} - P_{i-1,j-1} - 2 \times P_{i-1,j} - P_{i-1,j+1} \end{aligned} \quad (1)$$

因此边缘矢量的强度近似为:

$$Amp(\vec{D}_{i,j}) = |dx_{i,j}| + |dy_{i,j}| \quad (2)$$

方向为:

收稿日期:2006-04-03; 修订日期:2006-06-06

作者简介:田晓冬(1978-),男,河南漯河人,硕士研究生,主要研究方向:视频编解码; 田裕鹏(1964-),男,山东潍坊人,副教授,硕士,主要研究方向:视频编解码、图像处理。

$$\text{Ang}(\vec{D}_{i,j}) = \frac{180^\circ}{\pi} \times \arctan\left(\frac{dy_{i,j}}{dx_{i,j}}\right), |\text{Ang}(\vec{D}_{i,j})| < 90^\circ \quad (3)$$

这样,就建立起了原始图像各像素点  $P_{i,j}$  的边缘矢量强度和方向。

#### 2) 基于边缘方向信息的 $4 \times 4$ 模式选择快速算法

对于  $4 \times 4$  亮度块的帧内预测,共有 8 个基于方向的预测模式。因此可以将整个预测方向图用二分法划分成若干个预测区间。例如,模式 1 与模式 8 的夹角分别为  $\pm 26.6^\circ$ <sup>[3]</sup>,用二分法将模式 1,8 之间和模式 1,6 之间的夹角等分,这样就可以得到包含预测模式 1 的预测区间: $(-13.3^\circ, 13.3^\circ)$ 。这

样每一个像素点可以根据其边缘矢量方向  $\text{Ang}(\vec{D}_{i,j})$  所处的预测区间确定各像素点的预测模式。在  $4 \times 4$  块中将相同预测模式像素点的边缘矢量强度  $\text{Ang}(\vec{D}_{i,j})$  求和,即可确定边缘方向直方图。

显然,具有最大边缘方向直方图的编码模式代表了这一个  $4 \times 4$  块的边缘方向信息。而这个方向以及它左右相邻的两个方向与最优和次优的预测方向是密切相关的,因此都作为候选模式被挑选出来。由于 DC 预测并非是基于方向的预测,并且考虑到图像的边界宏块或块,以及各区间出现边缘矢量强度  $\text{Ang}(\vec{D}_{i,j})$  相等的特殊情况,DC 预测也应是候选模式之一。

#### 3) 基于边缘方向信息的 $16 \times 16$ 模式选择快速算法

对于  $16 \times 16$  亮度宏块帧内预测,可以采用同样的方法,利用 Sobel 算子获得这个  $16 \times 16$  亮度块中 256 个像素的边缘矢量强度和方向,并构建  $16 \times 16$  边缘方向直方图。

由于此时只有 3 种预测方向,换句话说,各个预测方向之间的相关性很小,所以只需要考虑最大边缘方向直方图所对应的预测模式和 DC 预测作为  $16 \times 16$  帧内预测的候选模式。这样就可以将 4 种候选模式减少为 2 种。

#### 4) 基于边缘方向信息的 $8 \times 8$ 模式选择快速算法

$8 \times 8$  色度块帧内预测与  $16 \times 16$  亮度块帧内预测基本类似,只是将 DC 预测和垂直预测的模式编号进行了调换。但色度分量包含  $U$  和  $V$  两个分量,因此需要分别计算两个色度块的最大边缘方向直方图,所对应的两个预测模式相同就可以将 4 种候选模式减少为 2 种( $U$  和  $V$  两个预测模式不同时为 3 种)。

## 2 Pan 算法的改进

### 2.1 $4 \times 4$ 亮度块模式选择提前中止

经 Pan 算法估计  $4 \times 4$  亮度块边缘方向后,对剩下的 4 种帧内预测候选模式进行率失真优化 RDO,率失真代价最小的预测模式作为  $4 \times 4$  亮度块的帧内预测模式。试验发现在最优的帧内预测模式下, $4 \times 4$  残差经变换、量化后有很多成为全零块。根据这种现象可以定义一个阈值  $T$ ,如果在某一模式下其率失真代价小于  $T$ ,可认为达到最佳模式,提前中止率失真优化。其阈值  $T$  确定如下:

H. 264/AVC 采用的是类 DCT 变换<sup>[6]</sup>。(4) 式为  $4 \times 4$

DCT 整数变换,  $X$  为  $4 \times 4$  亮度块经预测后残差矩阵,  $C_f$  为类 DCT 整数变换矩阵,  $Y$  为整数变换后量化前的系数矩阵。

$$Y = C_f X C_f^T \quad (4)$$

其中:

$$C_f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

变换后的直流系数  $DC$  如式(5):

$$DC = \sum_{x=0}^3 \sum_{y=0}^3 E(x,y) = \sum_{x=0}^3 \sum_{y=0}^3 (s(x,y) - s'(x,y)) \quad (5)$$

其中,  $s(x,y)$  是当前  $4 \times 4$  块的原始像素值,  $s'(x,y)$  是预测后重建块的像素值。如果  $DC$  趋近于 0,那么其他交流系数  $AC$  也趋于 0,整个变换后的矩阵趋向于 0 矩阵,因此可省略后面的处理过程。

在 H. 264/AVC 中如果变换系数经过量化后,其量化值绝对值小于等于 1 时,则残差矩阵可视为零矩阵。即满足下面的不等式:

$$|DC| < (2^{q\_bits} - qp\_const) / \text{quant\_coef}[qp\_rem][0][0]$$

其中:  $qp\_per = QP/6$ ,  $qp\_rem = QP\%6$ ,  $qp\_bits = qp\_per + 15$ ,  $qp\_const = (1 \ll q\_bits) / 6$ ,  $QP$  为量化参数,  $\text{quant\_coef}[qp\_rem][0][0]$  为乘法因子  $MF$ <sup>[3]</sup>。对于每个  $4 \times 4$  块,  $SAD$  的计算如下式所示:

$$SAD = \sum_{x=0}^3 \sum_{y=0}^3 |s(x,y) - s'(x,y)|, \text{ 即 } |DC| \leq SAD$$

则如果下式成立,则满足零矩阵的判断。

$$SAD < (2^{q\_bits} - qp\_const) / \text{quant\_coef}[qp\_rem][0][0]$$

阈值  $T$  由  $4 \times 4$  块的 RDO 确定:

$$J_{mode}(mode) = SSD + \lambda_{mode} R(mode, X)$$

其中  $mode$  表示所选的预测模式,  $R$  包括模式类型和残差的熵编码。而  $SSD = \sum_{x=0}^3 \sum_{y=0}^3 |s(x,y) - p(x,y)|^2$ ,  $p(x,y)$  是预测后残差经过变换、量化、反量化、反变换后与参考块的值相加得到,因而  $p(x,y)$  与  $s'(x,y)$  相匹配(大致相等),故下面不等式成立:

$$SAD \leq SAD^2 < ((2^{q\_bits} - qp\_const) / \text{quant\_coef}[qp\_rem][0][0])^2$$

这时,该块残差量化系数将趋于零,因此得到  $T$ 。

$$T = ((2^{q\_bits} - qp\_const) / \text{quant\_coef}[qp\_rem][0][0])^2$$

实际中可能存在偏差,因此设置一个校正因子  $\alpha$ 。

$$T = \alpha \times ((2^{q\_bits} - qp\_const) / \text{quant\_coef}[qp\_rem][0][0])^2$$

试验确定  $\alpha$  等于 0.5。

### 2.2 基于空间相关性的块类型选择

在 Pan 算法中,对  $4 \times 4$  亮度块候选预测模式与  $16 \times 16$  亮度块候选预测模式分别进行率失真优化 RDO,率失真代价小的作为宏块的编码模式。试验表明编码宏块亮度成分块类型选择是帧内  $4 \times 4$  还是帧内  $16 \times 16$ ,在很大程度上与左边宏块、上边宏块的块类型相关,并且这种相关性随着量化参数  $QP$  的增大而减小<sup>[7]</sup>。利用这种相关性,可确定当前宏块亮度成分的最大可能块类型(Most Probable Block Type, MPBT)。

同时,研究表明帧内  $16 \times 16$  和帧内  $4 \times 4$  随宏块纹理复杂的 RDO 曲线是不同的。

从图 1 可知,在纹理平坦区域,帧内  $16 \times 16$  生成的重建

图像能更好地与原始图像相匹配,故率失真较小,选择帧内  $16 \times 16$  概率大于帧内  $4 \times 4$ ;但随着纹理复杂度的逐渐增加,帧内  $16 \times 16$  在细节内容上率失真逐渐增加,到某个阈值  $T_0$  时大于帧内  $4 \times 4$  的率失真,这时帧内  $4 \times 4$  更适合作为宏块编码的选择模式。

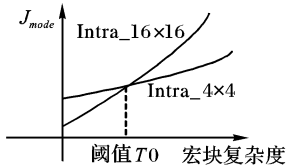


图 1 Intra\_16×16 与 Intra\_4×4 率失真变化

由以上分析可知:选择帧内  $4 \times 4$  还是帧内  $16 \times 16$  编码模式,关键是确定阈值  $T_0$ 。我们可以利用空间相关性以及最大可能块类型 MPBT,来确定当前宏块的纹理复杂程度。如果当前编码宏块亮度成分的最大可能块类型 MPBT 是帧内  $4 \times 4$  且其阈值为  $T_1$ ,当前宏块亮度成分按 MPBT 进行编码,若率失真代价  $RDCost_{currMB}$  大于  $T_1$ ,说明当前宏块的纹理更复杂、细节更丰富,可认为帧内  $4 \times 4$  更适合作为宏块编码的选择模式。同理,如果当前编码宏块亮度成分的最大可能块类型 MPBT 是帧内  $16 \times 16$  且其阈值为  $T_2$ ,当前宏块亮度成分按 MPBT 进行编码,若率失真代价  $RDCost_{currMB}$  小于  $T_2$ ,说明当前宏块的纹理更平坦,可认为帧内  $16 \times 16$  更适合作为宏块编码的选择模式。其算法流程以及阈值  $T_1$ 、 $T_2$  更新如下:

1) 若当前宏块相邻左边、上边宏块不全都能获得时,帧内  $4 \times 4$  与帧内  $16 \times 16$  分别进行率失真优化 RDO。率失真代价小的作为宏块的编码模式,跳转到 4); 否则跳转到 2)。

2) 在相邻宏块(左、上)中,具有最小率失真代价的宏块编码类型决定当前宏块的最大可能块类型(MPBT)。若  $MPBT = Intra\_4 \times 4$ ,按 MPBT 进行编码且率失真代价  $RDCost_{currMB} > T_1$ ,  $Intra\_4 \times 4$  作为宏块的编码模式,跳转到 4)。同理,若  $MPBT = Intra\_16 \times 16$ ,按 MPBT 进行编码且率失真代价  $RDCost_{currMB} < T_2$ ,  $Intra\_16 \times 16$  作为宏块的编码模式,跳转到 4); 否则,跳转到 3)。

3) 对除 MPBT 以外的另一种块类型,进行率失真优化 RDO,选择率失真代价最小的作为宏块的编码模式。

4) 若还有其他宏块,跳转到 1)。

在 2) 中,阈值  $T_1$ 、 $T_2$  的确定如下:

若左、上边宏块的块类型都是  $Intra\_4 \times 4$ ,则  $Intra\_4 \times 4$  就是 MPBT 且  $T_1 = \min\{RDCost_{above}RDCost_{left}\} + 15 \times \lambda(Q_p)$ ;

若左、上边宏块的块类型都是  $Intra\_16 \times 16$ ,则  $Intra\_16 \times 16$  就是 MPBT 且  $T_2 = \max\{RDCost_{above}RDCost_{left}\}$ ;

若左、上边宏块的块类型不相同:

当  $MPBT = Intra\_4 \times 4$  时,  $T_1 = RDCost_{Intra\_4 \times 4} + 15 \times \lambda(Q_p)$

当  $MPBT = Intra\_16 \times 16$  时,  $T_2 = RDCost_{Intra\_16 \times 16}$

### 3 试验结果

该算法应用于 H.264/AVC JM8.6 参考软件中<sup>[8]</sup>。在全 I 帧、baseline 档次下,按照文献[9]的建议分别在量化参数 28、32、36、40 条件下,分别对 5 个 CIF、5 个 QCIF 150 帧测试序列进行试验,就编码时间、比特率和 PNSR 等性能指标,比较改进的 Pan 算法(IPan)、Pan 算法、JM86 率失真全搜索(JM86)三者帧内预测的性能。其中表 1 比较三者的编码时间: $\Delta t\_P\_JM$  为 Pan 算法较 JM86 平均节省编码时间百分比; $\Delta t\_IP\_JM$  为改进 Pan 算法较 JM86 平均节省编码时间百分比; $\Delta t\_IP\_P$  为改进 Pan 算法较 Pan 算法平均节省编码时间百

分比。试验结果显示 Pan 算法的平均编码时间仅需 JM86 编码时间的 41.17%,改进 Pan 算法的平均编码时间仅需 JM86 编码时间的 29.118%,改进 Pan 算法的平均编码时间较 Pan 算法编码时间减少 29.093%。图 2 表示 Mobile 视频序列三种算法的 RD 曲线,从中看出本算法编码时间的减少是以峰值信噪比(PSNR)的减小、单帧比特数的增加为代价的,但基本保证图像率失真和码率性能。

表 1 改进 Pan 算法、Pan 算法、JM 编码时间比较

格式	序列	$\Delta t\_P\_JM$	$\Delta t\_IP\_JM$	$\Delta t\_IP\_P$
QCIF	Foreman	-57.981	-70.492	-29.665
	News	-57.633	-70.156	-29.483
	Container	-58.658	-71.012	-29.799
	Silent	-58.549	-72.695	-33.948
	Coastguard	-55.953	-70.224	-32.208
CIF	Paris	-60.107	-71.603	-28.753
	Mobile	-59.921	-68.412	-21.073
	Tempete	-60.555	-71.728	-28.145
	Bus	-59.298	-71.975	-31.019
	Stefan	-59.643	-70.522	-26.841

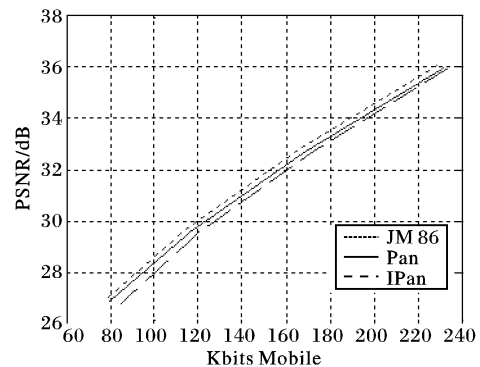


图 2 Mobile 曲线

### 参考文献:

- [1] ITU-T Rec. H. 264/ ISO/ IEC14496-10 AVC( Doc JVT - G050), Draft ITU-T recommendation and final draft international standard of joint video specification [ S ].
- [2] WIEGAND T, SULLIVAN T, BJNTEGAARD GJ, et al. Overview of the H. 264/AVC Video Coding Standard[ J ]. IEEE Trans. Circuits Syst. Video technol, 2003, 13: 560 - 576.
- [3] RICHARDSON I. H. 264 and MPEG - 4 Video compression [ M ]. London: John Wiley & Sons, 2003.
- [4] PAN F, LIN X. Fast Mode Decision for Intraprediction ( Doc JVT-G013)[ R ]. Pattaya, Thailand: JVT of ISO / IECMPEG & ITU-T VCEG, 2003.
- [5] PAN F, LIN X, RAHARDJA S. Fast Mode Decision Algorithm for Intraprediction in H. 264/AVC Video Coding[ J ]. IEEE Trans. Circuits Syst Video technol, 2003, 15: 813 - 822.
- [6] HALLAPURO A, KARCZEWICZ M. Low complexity transform and quantization—Part 1: Basic Implementation ( Doc JVT-B038) [ R ]. Geneva: JVT of ISO / IEC MPEG & ITU-TVCEG, 2001.
- [7] FU FW, LIN XG, XU LD. Fast Intra Prediction Algorithm in H. 264/ave[ A ]. International Conference Signal Processing[ C ]. 2004. 1191 - 1194.
- [8] SCHÄFFERR. H. 264 JVT Reference Software Version8.6[ CP/OL ]. [http://iphome.hhi.de/suehring/tml/download/old\\_jm/](http://iphome.hhi.de/suehring/tml/download/old_jm/).
- [9] JVT Test Model Ad Hoc Group. Evaluation Sheet for Motion Estimation[ R ]. Draft version 4, 2003.