

文章编号: 1005-0523(2006)02-0086-03

H. 264/AVC 快速帧间编码算法

张晓飞¹, 彭强¹, 桂曼²

(1. 西南交通大学 信息科学与技术学院, 四川 成都 610031; 2. 南京铁路车辆段计算机中心, 江苏 南京)

摘要: 基于率失真优化的 H. 264/AVC 帧间宏块编码模式, 需要对 10 种模式进行率失真优化计算才能得到一个宏块的最优编码模式, 这种宏块的最优编码模式是以巨大计算复杂度为代价的. 为了降低计算复杂度, 在实验统计了各种宏块编码模式在不同 QCIF 视频序列中所占的比例及分析各种编码模式的适用性情况下, 提出了一种快速帧间编码算法, 它采用优先判断 SKIP 模式和图像纹理相结合的方法, 在基于 H. 264/AVC 测试模型 JM7.6 上的仿真试验表明, 所提出的快速算法, 在基本保持图像质量和码率基本不变的情况下, 可以有效提高编码速度.

关键词: H. 264/AVC; 编码模式; 帧间编码; 快速算法

中图分类号: TP391.41

文献标识码: A

1 前言

H. 264/AVC 是 ITU-T VCEG 和 ISO/IEC MPEG 联合发布的最新的视频编码标准. 实验表明, 在同等级图像质量的情况下, H. 264/AVC 比 H. 263+ 或 MPEG-4 要节省 50% 的码率^[2]. 然而, H. 264/AVC 编码效率的提高是以增加巨大计算复杂度为代价的.

2 H. 264/AVC 帧间编码

帧间编码模式是 H. 264/AVC 视频编码标准的重要内容, 与以往的视频编码标准类似, H. 264/AVC 的帧间预测也是基于块的运动补偿预测^[3]. 过去的标准中只定义了两种块的大小, 分别为 16×16 和 8×8 的正方形块 (8×8 的块是在 H. 263 和 MPEG-4 中定义的). 但是由于视频图像的复杂性, 在较大的块中可能包含多个具有不同运动状态和不同形状的对象. 特别是在运动剧烈的局部区域中, 用一个宏块或 4 个 (8×8) 运动矢量并不能准确地描述一个宏块全部的运动细节^[4]. 在 H. 264/AVC 视频编码标准中, 帧间宏块的划分有七种模式:

$$\text{Mode} = \{16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8, 8 \times 4, 4 \times 8, 4 \times 4\}$$

(1)

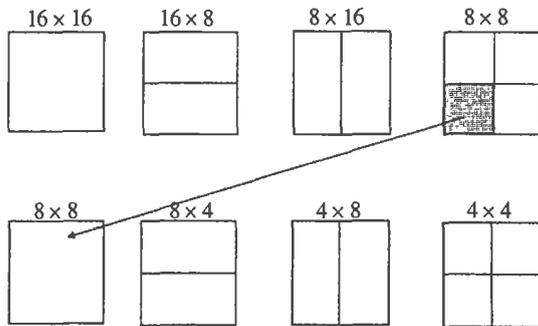


图 1 H. 264/AVC 中帧间宏块划分的 7 种模式

为了得到一个宏块的最佳编码模式, 要通过 R-D (率失真优化) 公式计算:

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

计算拉格朗日函数值 J 即率失真开销, 经过比较, 选取最小值下的编码模式 MODE 作为宏块的最终编码模式. 其中 QP 为宏块的量化参数, s, c 分别表示原始图像和重建图像的像素值, $R(s, c, \text{MODE} | QP)$ 表示在特定 QP 和 MODE 下当前宏块的编码输出比特数, $\text{SSD}(s, c, \text{MODE} | QP)$ 表示图像失真度, 计算公式为:

收稿日期: 2005-09-10

作者简介: 张晓飞 (1978-), 男, 安徽芜湖人, 硕士研究生, 主要研究方向: 视频处理与通信.

$$SSD(s, c, MODE | QP) = \sum_{x=1, y=1}^{16, 16} (s_V[x, y] - c_V[x, y, MODE | QP])^2 + \sum_{x=1, y=1}^{8, 8} (s_U[x, y] - c_U[x, y, MODE | QP])^2 + \sum_{x=1, y=1}^{8, 8} (s_V[x, y] - c_V[x, y, MODE | QP])^2, \lambda_{MODE} \text{ 为拉格朗日系数, } MODE \text{ 为当前宏块可选的一种编码模式.}$$

由上述可知,如果减少做率失真计算的编码模式数量,同时能够保持编码器的性能基本不变,那么必能节省编码时间.

在 H.264/AVC 中,为了提高编码效率,对于不同纹理特征的宏块采用不同的编码模式.纹理特征可以用宏块之间的残差表示.通常对运动补偿后残差较小的宏块采用大的块尺寸(16×16, 18×8, 8×16)编码,对于残差较大的宏块采用小的块尺寸(8×8, 8×4, 4×8, 4×4)编码,而对于残差为 0 的宏块则采用 SKIP/DIRECT 模式(P 帧为 SKIP 模式, B 帧为 DIRECT 模式)进行编码^[5].

下表是基于 H.264/AVC 测试模型 JM7.6^[6]对 QCIF 序列的 P 帧宏块编码模式进行的统计.序列编码结构为 IPPP, 每个序列分别测试了 51 帧,其中第一帧为 I 帧,然后对 50 个 P 帧进行统计.

表 1 P 帧宏块编码模式百分比

Sequence	Mode ⁰	Mode ¹	Mode ²	Mode ³	Mode ⁸	Mode ⁹	Mode ¹⁰
Foreman	24.7	3.04	10.2	18.5	16.1	0	0
Container	79.0	9.3	3.4	3.2	5.1	0	0
Mobile	2.2	20.6	12.9	12.7	51.5	0	0
News	78.4	5.5	1.9	4.6	9.5	0	0
Carphone	33.9	30.7	9.6	11.0	13.8	0.2	0.7
Mother_daughter	69.8	14.3	5.2	6.1	4.6	0	0
Glasgow	22.9	15.5	8.4	5.8	43.9	3.4	0.2

Mode⁰ 为 SKIP 模式, Mode⁸ 包括 8×8, 8×4, 4×8, 4×4 模式, Mode⁹ 为 16×16 模式, Mode¹⁰ 为 14×4 模式.

由表 1 可知,运动平缓的视频序列宏块运动补偿后残差较小,采用大的块尺寸编码较多,如 Container 序列 Mode⁰ 所占比例高达 79%, Mode¹ 所占比例为 9.3%, 两者合计占 87.3%, News 序列 Mode⁰ 所占比例为 78.4%, Mode¹ 所占比例为 14.3%, 两者合计占 84.1%. 运动剧烈的视频序列宏块运动补偿后残差较大,因而采用小的块尺寸的较多,如 Mobile 序列 Mode⁰ 所占比例仅有 2.2%, 而 Mode⁸ 所占比例有 51.1%.

3 快速帧间编码算法

SKIP 模式是直接从前一参考宏块复制,代替当前帧的宏块,它的特点是运动补偿残差为 0.一般在宏块无运动向量或宏块做整体运动即只有一个运动向量时采用 SKIP 模式编码,因而计算简单,速度快.因此首先应判断当前宏块是否用 SKIP 模式进行编码.

由第二节分析可知,序列的纹理特征直接影响帧间宏

块编码模式的划分.纹理特征可以从帧间宏块之间的 SAD 获得.

$$SAD = \frac{1}{256} \sum_{i=1}^{16} \sum_{j=1}^{16} |x_{i,j} - y_{i,j}| \quad (3)$$

其中 $x_{i,j}$ 和 $y_{i,j}$ 分别当前帧和前一帧(i, j)处像素的灰度值.另外还要设一阈值 T_1 ,用于和 SAD 进行比较.由于每一帧的运动剧烈程度是不一样的,因此阈值 T_1 应该是动态的.

$$T_1 = \frac{1}{w} \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N |x_{i,j} - y_{i,j}| \quad (4)$$

其中 M 和 N 分别为一帧的宽和高, $x_{i,j}$ 为(i, j)处像素的灰度值, w 为系数;当 QP 等于 28、32、36、40 时, w 分别为 1.2、1.0、0.8、0.6.

提出的快速算法首先计算 SKIP 模式即 Mode⁰ 的率失真代价 $RDCost^0$ 作为判断当前宏块是否采用 SKIP 模式编码的依据,如果 $RDCost^0$ 小于 T_0 ,则当前宏块采用 SKIP 模式编码,否则计算当前宏块纹理特征 SAD,其中 T_0 为通过试验和统计得到的经验值为 4500.

如果 SAD 小于 T_1 ,则采用 Mode¹ 中的块划分模式进行 R-D 计算,从中确定一最佳模式,否则在 Mode² 进行 R-D 计算,确定一最佳模式.

其中:

$$MODE1 \in \{16 \times 16, 16 \times 8, 8 \times 16\}$$

$$MODE2 \in \{8 \times 8, 8 \times 4, 4 \times 8, 4 \times 4\}$$

以上算法用于编码一帧的具体步骤如下:

1) 对当前宏块计算其 $RDCost^0$, 如果 $RDCost^0$ 小于阈值 T_0 , 则 SKIP 模式即作为当前宏块的编码模式. 否则转到第二步;

2) 计算当前宏块的 SAD, 如果 SAD 小于 T_1 , 则当前宏块的划分模式 \in Mode¹, 分别计算 16×16 、 16×8 、 8×16 的 RD-Cost, 确定该宏块的划分模式. 否则转到第三步;

3) 当前宏块的划分模式 \in Mode², 在 Mode² 中计算 $RDCost$, 确定该宏块的划分模式;

4) 重复以上步骤,直至完成当前帧所有宏块的编码.

4 实验结果

本算法在 H.264/AVC 测试模型 JM7.6 上实现, 计算机配置为: Pentium (R) 4 2.93GHz CPU, 224MB 内存, Window XP SP2. 为了客观全面地验证该算法的效果, 实验对具有代表性的多个不同运动程度的 QCIF 视频序列分别进行了测试, 测试视频序列如下:

Foreman, container, mobile, news, suize, carphone, mother_daughter.

测试模型的配置参数为: 编码 100 帧, 帧率为 10fps, 量化参数为 28, 序列编码结构为 IPPP, 采用 Hadamard 变换, 搜索范围为 ± 16 像素, 5 个参考帧, 不使用流控, 采用 CABAC.

实验结果如下:

表2 快速算法与 H.264/AVC 算法性能比较

Sequence	PSNR(dB)	Bitrate(%)	Time(%)
Foreman	-0.01	0.18	25.11
Container	-0.03	0.94	28.71
Mobile	0	0.25	22.42
News	0	-0.60	27.31
Carphone	-0.03	0.09	28.40
Mother_daughter	-0.04	1.51	29.41
Glasgow	-0.01	-0.17	23.94

表2的实验结果验证了提出的快速算法的性能.在峰值信噪比(PSNR)方面,快速算法略有下降,平均降幅为0.02dB,其中 Mobile 和 News 序列 PSNR 下降为0;比特率平均增加0.31%,其中 News 和 Glasgow 序列比特率有所降低;时间平均节省26.47%.

H.264/AVC 编码时的率失真优化算法是为了达到码率和图像失真之间的最优组合,提出的快速算法是在降低计算复杂度的情况下寻找一种码率和失真的组合,因而得到的码率和失真不一定是最佳组合,所以会导致图像的 PSNR 降低.

5 结论

支持多种宏块划分模式是 H.264/AVC 提高编码效率的重要途径.通过率失真优化计算能够达到图像质量和码率之间的最优组合,然而这是以增加计算复杂度为代价的.通过分析 H.264/AVC 帧间编码算法,提出的快速帧间编码算

法能够在图像质量和码率变化基本可以忽略的情况下,有效提高编码速度.该算法是在编码结构为 IPPP 的序列上实现的,同样可以扩展到 B 帧中.

参考文献:

- [1] "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC)," [S] in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVTG050, 2003. 8, 123-144.
- [2] Thomas Wiegand, Gary J. Sullivan, et al. "Overview of the H.264/AVC Video Coding Standard" [J]. IEEE Trans on Circuits and Systems for Video Technology, 2003, 7(13), pp: 560-572.
- [3] T. Wiegand and B. Girod. "Multi-Frame Motion-Compensated Prediction for Video Transmission" [J]. Norwell, MA: Kluwer, 2001.
- [4] Lim, K.P., Wu, S., Wu, D.J., Rahardja, S., Lin, X., Pan, F., and Li, Z.G.: "Fast inter mode selection" [J]. 9th Joint Video Team (JVT) Mtg, San Diego, CA, USA, September, 2003.
- [5] M. Flierl and B. Girod, "Generalized B pictures and the draft JVT/H.264 video compression standard" [J]. IEEE Trans. Circuits Syst. Video Technol., vol. 13, pp. 587-597, July 2003. [13] T. Wiegand, H. Schwarz, A. Joch, .
- [6] JVT Reference Software Version 7.6 [CP] <http://bs.hhi.de/~surhning/tml/download>.

Fast Intermode Coding Algorithm for H.264/AVC

ZHANG Xiao-fei¹, PENG Qiang¹, GUI Man²

(1. School of Information Science & Technology, Southwest Jiaotong University, Chengdu 610031; 2. Nanjing Railway Vehicle Department, Nanjing, China)

Abstract: In H.264/AVC, when using rate-distortion optimization to get a best coding type for a macroblock, the encoder has to compute rate-distortion cost of 10 coding types. This costs a extremely computational complexity. To reduce computation complexity, a fast intermode coding algorithm was proposed by analyzing the statistics of inter-macroblock coding types ratio and their applicability in different QCIF video sequence. It considers both SKIP mode and the texture of picture. The fast algorithm has been realized based on the testing mode JM7.6 for H.264/AVC. It can efficiently save coding time while maintaining similar picture quality and compression ratio.

Key words: H.264/AVC; coding mode; intermode coding; fast algorithm