

一种基于 H.264 的可变块快速运动估计算法

陈 纯 杨 智 卜佳俊 莫林剑

(浙江大学计算机科学与技术学院, 杭州 310027)

摘 要 H.264 是最新的视频编码标准,它相对于以前的视频编码标准在编码效率上有了巨大的提高。高编码效率得益于采用了一系列新的编码技术,这也导致了较高的计算复杂度。H.264 运用了可变块运动估计方法,这也是 H.264 编码过程中最耗时的模块。提出了一种基于 H.264 的可变块快速运动估计算法。该算法基于以下 3 种策略:静止块的预测、非静止块的预测搜索和合并过程中的自适应精细搜索。试验结果证明,该算法能够将计算复杂度降低到快速全搜索运动估计算法的 3%,而 PSNR 和码率几乎与快速全搜索算法得到的结果相当。

关键词 H.264 视频编码 运动估计

中图分类号: TN919.81 **文献标识码:** A **文章编号:** 1006-8961(2007)02-0272-05

A Fast Variable-block-size Motion Estimation Algorithm for H.264

CHEN Chun, YANG Zhi, BU Jia-jun, MO Lin-jian

(College of Computer Science, Zhejiang University, Hangzhou 310027)

Abstract H.264, the latest video coding standard, achieves significant improvement in coding efficiency compared with previous standards. The high coding efficiency is based on some new techniques which also lead to high computational-complexity. H.264 uses variable-block-size motion estimation, which is the most time-consuming module. In this paper, we propose a fast variable-block-size motion estimation algorithm for H.264. The algorithm is based on three schemes: stationary block prediction, predictive search for non-stationary blocks and adaptive refinement search in merging process. Experimental results show that our algorithm can reduce the computational-complexity to 3% of fast full search motion estimation algorithm, with negligible PSNR loss and bit rate increase.

Keywords H.264, video coding, motion estimation

1 引 言

H.264 是目前最新的国际视频编码标准^[1]。由于 H.264 采用了一系列新的编码技术,如帧内块预测、可变块运动估计、多参考帧、环内滤波器等,使得 H.264 的编码效率相对于以前的视频编码标准有了巨大的提高^[2]。与此同时,由于许多新技术的引入,H.264 的编码复杂度也大大提升。怎样有效地降低 H.264 的编码复杂度同时保持其较高编码效率成为了当前研究的热点,特别是针对视频实时编码传输的应用领域。

与以前的视频编码标准类似,H.264 同样采用了块匹配的运动估计算法来消除帧间的冗余信息。运动估计仍然是 H.264 编码过程中最耗时的模块,全搜索运动估计将占用 80% 以上的编码时间。然而,与先前的视频编码标准不同的是 H.264 采用了可变块的块匹配运动估计技术。宏块可被划分为 7 种子块模式进行匹配,如图 1 所示。宏块中的子块被划分得越小,块匹配也就越精确,那么编码效率就越高。但是,子块的模式划分越多,运动估计中搜索过程所带来的计算复杂度就越大。

传统的快速块匹配运动估计算法基本上是基于以下几种策略。一是采用快速搜索的模型,如著名

基金项目:国家自然科学基金项目(60203013);浙江省科技计划重点项目(2005C23047)

收稿日期:2005-10-14;改回日期:2005-11-06

第一作者简介:陈纯(1955~),男,教授,博士生导师。1990 年于浙江大学计算机应用专业获博士学位。研究方向为图形图像处理、计算机视觉、嵌入式系统等。E-mail: chenc@zju.edu.cn

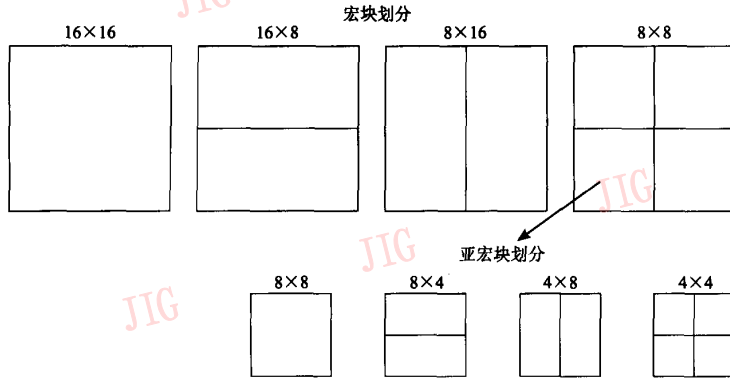


图 1 宏块和亚宏块的划分模式

Fig. 1 Partition modes of macroblock and sub-macroblock

的菱形搜索算法^[3],六边形搜索算法^[4]等;二是利用时间和空间的相关性,采用运动矢量的提前预测算法^[5];三是利用统计规律,设定阈值来提前终止搜索过程^[6];四是采用快速的匹配度计算方程^[7]。虽然以上提及的算法都能减轻运动估计的计算复杂度,但它们都没有充分地利用 H. 264 运动估计中的可变块特性。近来,一些基于 H. 264 的快速运动估计算法也被提出。在联合视频工作小组(JVT)的参考软件^[8]中,采用了快速的全搜索运动估计算法。在文献^[9]中,采用了自底向上的合并和自适应阈值来加速运动估计的搜索过程。

本文提出了一种基于 H. 264 的可变块快速运动估计算法。此算法包含 3 个阶段。首先,自上而下地对静止块进行预测;其次,采用提前预测的快速搜索算法对非静止的 4 × 4 块进行搜索;最后,在自底向上的合并搜索过程中采用自适应精细搜索方法。试验结果证明,该算法能够大大地降低可变块运动估计的计算复杂度,同时保证了与快速全搜索

算法相当的搜索精确度。

2 算法介绍

2.1 静止块的预测

在诸如视频会议、视频电话等许多实时视频编码传输的应用中,视频前后两帧之间往往只存在很少的运动,在这种情况下,很多块都可认为是静止块。而且通过全搜索的统计结果,发现在宏块划分中,16 × 16 的块模式占据了很大比例;在亚宏块的划分中,8 × 8 的块模式占据了较大比例。表 1 呈现了其试验结果。所以,如果能够在运动估计之前预测出静止块,那么计算量将大大的降低。进一步的研究分析指出了静止块的绝对误差和(SAD)值远低于非静止块在(0,0)位置的 SAD 值。因此,静止块的预测可以通过计算当前块在(0,0)位置的 SAD 值,然后与预先设定的阈值 T 进行比较。如果 SAD 值小于此阈值,当前块可判定为静止块,从而避免了其运动矢量的搜索。

表 1 16 × 16 和 8 × 8 的块模式比例

Tab. 1 Proportion of 16 × 16 and 8 × 8 modes

测试序列	16 × 16 在宏块划分中比例	静止 16 × 16 在宏块划分中比例	8 × 8 在亚宏块划分中比例	静止 8 × 8 在亚宏块划分中比例
news	91.84	87.34	58.24	25.76
mother&daughter	90.33	70.81	67.94	20.43
foreman	81.03	20.81	60.16	12.11
mobile&calendar	69.60	8.91	53.11	6.43
coastguard	68.33	3.81	55.76	6.75

单位:%

既然大多数的块模式是 16×16 和 8×8 , 设定两个阈值来预测 16×16 和 8×8 的静止块。通过对众多测试序列的试验分析, 16×16 静止块的平均 SAD 值在 450 ~ 1 000 范围内, 8×8 静止块的平均 SAD 值在 150 ~ 350 范围内。阈值越大将造成静止块预测误差的增大, 因此设定 $T_1 = 500, T_2 = 160$ 来预测 16×16 和 8×8 的静止块。这两个阈值能够保证在减少计算量的同时几乎不会影响到运动估计的精确度。

2.2 非静止块的预测搜索

从表 1 中可以看出, 针对某些视频序列, 如“foreman”、“mobile&calendar”、“coastguard”, 它们包含了大量的乱序运动和全局运动, 静止块的预测对这些视频序列产生的意义不大。

在静止块预测之后, 将对非静止宏块中的 4×4 块进行运动矢量(MV)的搜索。根据文献[5], 可以利用空间和时间的相关性来预测当前块的运动矢量。然而, 利用时间相关性必须记录前一帧的所有运动矢量; 利用空间相关性必须记录当前帧中相邻块的运动矢量。为了避免过多存储空间的占用和避

免过多条件判断的运算, 仅仅考虑当前宏块内部的空间相关性来进行运动矢量的预测, 如图 2 所示。

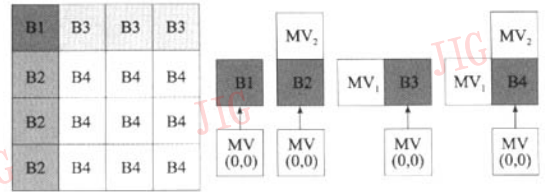


图 2 宏块内运动矢量的预测

Fig.2 MV prediction in macroblock

选择 $MV(0,0)$, 左边块 $MV(MV_1)$ 和上边块 $MV(MV_2)$ 来作为预测值。从图中可以看到, 左上角的块 B1 仅有一个预测值 $MV(0,0)$, 左边块 B2 和上边块 B3 拥有两个预测值, 其他块 B4 拥有 3 个预测值。分别算出当前块在所有预测 MV 位置的 SAD 值, 选择 SAD 最小的预测 MV 作为搜索的中心点。运动矢量预测之后, 将采用菱形搜索算法^[3]来完成 4×4 块的运动矢量搜索, 菱形搜索过程如图 3 所示。

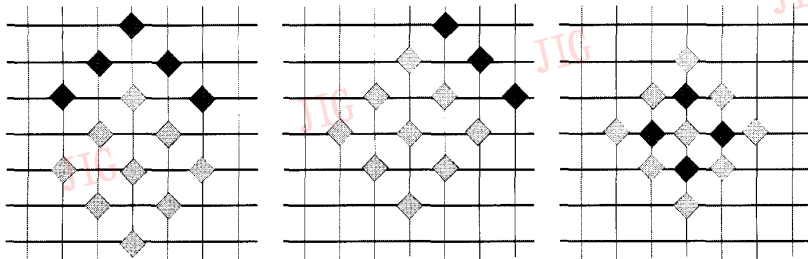


图 3 菱形搜索过程

Fig.3 Diamond search process

2.3 合并过程中的自适应精细搜索

在对非静止宏块中的所有 4×4 块进行 MV 搜索之后, 将采用自底向上的合并过程对其他模式块的 MV 进行搜索。为了降低搜索的复杂度, 将利用 4×4 块的 MV 值来预测合并块的 MV 值, 如图 4 所示。

合并块的 MV 预测值由下式确定:

$$MV_n = \begin{cases} (MV_1 + MV_3)/2 & x = 5 \\ (MV_2 + MV_4)/2 & x = 6 \\ (MV_1 + MV_2)/2 & x = 7 \\ (MV_3 + MV_4)/2 & x = 8 \\ (MV_{5-6} + MV_{7-8})/2 & x = 9 \end{cases} \quad (1)$$

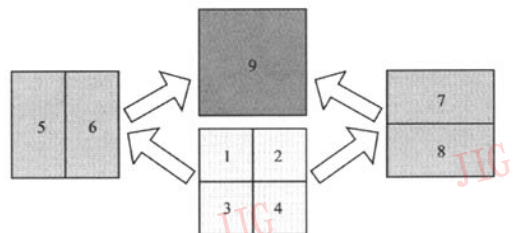


图 4 自底向上的合并预测

Fig.4 Bottom-up merging prediction

其中, $MV_{5-6} = (MV_5 + MV_6)/2, MV_{7-8} = (MV_7 + MV_8)/2, n$ 表示图 4 指定的相应块。预测的 MV 将作为下一步精细搜索的中心点。

在对合并块的 MV 进行预测之后,将自适应的采用图 3 所示的菱形搜索算法和图 5 所示的小菱形搜索算法来完成合并块 MV 的精细搜索。

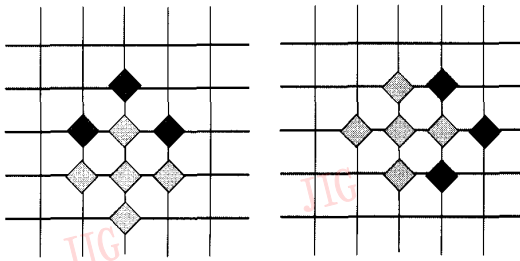


图 5 小菱形搜索过程

Fig. 5 Small diamond search process

从式(1)可以看到,合并块的预测过程都归结于使用了两个预测值。那么定义两个预测值的差分值

$$D1 = \max \{ |MV_{n1x} - MV_{n2x}|, |MV_{n1y} - MV_{n2y}| \} \quad (2)$$

其中, MV_{n1x} 和 MV_{n1y} 表示预测值 1 的 x 和 y 分量; MV_{n2x} 和 MV_{n2y} 表示预测值 2 的 x 和 y 分量。

同时定义合并块的预测 MV 和最终全搜索得到的真实 MV 之间的差分值

$$D2 = \max \{ |MV_{Px} - MV_{Tx}|, |MV_{Py} - MV_{Ty}| \} \quad (3)$$

其中, MV_{Px} 和 MV_{Py} 表示预测 MV 的 x 和 y 分量; MV_{Tx} 和 MV_{Ty} 表示真实 MV 的 x 和 y 分量。

通过统计分析,找到了 $D1$ 与 $D2$ 之间的相关性。从表 2 可以看出,如果 $D1 \leq 3$,那么 $D2$ 在范围为 2 的区域内已经占有很高的比例;如果 $D1 > 3$,那么 $D2$ 在范围为 6 的区域内才能达到较高的比例。

表 2 $D1$ 与 $D2$ 的统计关系

Tab.2 Relationship between $D1$ and $D2$

测试序列	单位: %			
	$D2 = 0$	$D2 \leq 1$	$D2 \leq 2$	$D2 \leq 3$
news	89.38	97.30	98.58	99.06
mother&daughter	69.73	89.34	93.21	95.09
$D \leq 3$ foreman	63.71	91.47	95.78	97.16
mobile&calendar	77.87	97.01	98.57	99.10
coastguard	79.18	97.01	98.70	99.14
news	1.46	36.20	79.73	95.02
mother&daughter	1.24	29.53	68.84	94.16
$D1 > 3$ foreman	1.40	32.21	73.73	95.20
mobile&calendar	1.34	38.76	87.76	97.59
coastguard	1.01	34.90	83.90	97.11

因此,自适应的精细搜索可以定义为:如果 $D1 = 0$,那么 $MV_T = MV_P$;如果 $D1 \leq 3$,那么采用小菱形搜索来完成合并块 MV 的精细搜索,并限制搜索范围在 $-2 \sim +2$ 区域内;如果 $D1 > 3$,那么采用菱形搜索来完成合并块 MV 的精细搜索,并限制搜索范围在 $-6 \sim +6$ 范围内。

2.4 算法小结

基于 H. 264 的可变块快速运动估计算法总结如下:

- (1) 计算宏块中 16×16 模式和 8×8 模式在 $(0, 0)$ 位置的 SAD 值;
- (2) 如果 $SAD_{16 \times 16} < T_1$,那么此宏块被认为是静止块,搜索结束。否则进入第 2 步;
- (3) 对于 4 个 8×8 块,如果 $SAD_{8 \times 8} < T_2$,那么此 8×8 块被标记为静止块,否则被标记为非静止块;
- (4) 如果 4 个 8×8 块都被标记为静止块,那么此宏块被认为是静止块,搜索结束,否则进入第 5 步;
- (5) 对于非静止宏块中的所有 4×4 块,采用菱形搜索算法进行 MV 的搜索;
- (6) 采用自底向上的合并和自适应精细搜索算法对其他模式划分块的 MV 进行搜索;
 - ① 从 4×4 块合并到 4×8 和 8×4 块,采用自适应精细搜索;
 - ② 从 4×8 和 8×4 块合并到 8×8 块,采用自适应精细搜索;
 - ③ 从 8×8 合并到 8×16 和 16×8 块,采用自适应精细搜索;
 - ④ 从 8×16 和 16×8 块合并到 16×16 块,采用自适应精细搜索。

3 试验结果与分析

为了验证可变块快速运动估计算法的性能,将此算法集成到 H. 264 参考软件 (JM8.5)^[8] 中进行试验测试。测试序列选择了 CIF (352 × 288) 分辨率的一系列视频,它们包含了各种各样的运动情况。编码帧率设定为每秒 30 帧;选择了 H. 264 的基本档次;量化参数的范围从 20 ~ 40;运动估计的搜索范围为 $-16 \sim +16$;采用了单参考帧。将此算法的速度,以及得到的 PSNR 和码率结果与快速全搜索算法进行比较和分析。结果如表 3 所示。

从表 3 中可看到,对于“news”和“mother&daughter”

表 3 此算法与快速全搜索算法的试验结果比较

Tab.3 Comparison of experimental results between proposed algorithm and fast full search

测试序列	量化参数	PSNR 变化	码率变化 (%)	加速比
news	20	-0.18	+3.70	40.5
	24	-0.11	+3.65	44.0
	28	-0.08	+1.88	46.7
	32	-0.03	+2.34	47.3
	36	-0.06	+5.19	48.9
	40	-0.03	+4.25	49.8
mother&daughter	20	-0.17	+0.61	38.4
	24	-0.10	+1.14	41.3
	28	-0.06	+2.11	44.1
	32	-0.04	+1.23	46.9
	36	-0.03	+2.17	48.9
	40	-0.02	+0.01	50.1
foreman	20	-0.10	+2.62	25.2
	24	-0.07	+3.27	25.4
	28	-0.06	+4.40	25.7
	32	-0.07	+6.04	26.4
	36	-0.08	+6.92	26.8
	40	-0.07	+7.71	27.1
mobile&calendar	20	-0.07	+0.18	29.9
	24	-0.05	+0.28	30.1
	28	-0.05	+0.24	30.2
	32	-0.05	+0.30	30.4
	36	-0.04	+0.47	30.7
	40	-0.05	+1.00	30.8
coastguard	20	-0.04	+0.23	31.1
	24	-0.03	+0.30	31.2
	28	-0.03	+0.72	31.3
	32	-0.05	+1.16	31.6
	36	-0.04	+2.30	31.7
	40	-0.04	+3.67	31.9
平均值		-0.065	+2.43	35.8

这样的序列,拥有静止的背景而且运动主要集中在中间区域,本文算法比快速全搜索算法快 40 倍以上,PSNR 仅下降 0.076,码率上升 2.35%;对于“foreman”这样的序列,拥有大量的乱序运动,本文算法比快速全搜索算法快 25 倍以上,PSNR 仅下降 0.075,码率上升 5.32%;对于“mobile&calendar”和“coastguard”这样拥有全局运动的序列,本文算法比快速全搜索算法快 30 倍以上,PSNR 和码率几乎相当。从试验结果可以看出,本文算法能够适应于各种各样运动情况的视频。特别需要提出的是,对于

视频电话和视频会议这样低码率视频编码的情况,静止块预测能够发挥较大的作用;对于镜头移动这样的全局运动视频,利用空间相关性的预测搜索能够体现较强的优势。

4 结 论

本文提出了一种基于 H.264 的可变块快速运动估计算法。通过统计分析,在运动估计过程中引入了 3 种预测方法。试验结果证明,该算法能够在计算复杂度和编码效率上获得一个较好的折中。计算复杂度可以降低到 H.264 参考软件中快速全搜索算法的 3%,而 PSNR 和码率几乎与快速全搜索算法相当。

该快速算法仍然拥有提升的空间,比如引入自适应的阈值来进行静止块的预测或者采用更加先进的搜索模式对 MV 进行搜索。进一步的工作是对算法进行扩展,在保持较低的计算复杂度的情况下,使其能够支持多参考帧的运动估计。

参考文献 (References)

- 1 ITU-T Rec. H.264/ISO/IEC 11496-10 Document JVT-G050 2003, Advanced Video Coding-Final Committee Draft[S].
- 2 Ostermann J, Bormans J, List P, et al. Video coding with H.264/AVC: tools, performance, and complexity[J]. IEEE Circuits and System Magazine, 2004, 4(1):7~28.
- 3 Zhu S, Ma K K. A new diamond search algorithm for fast block-matching motion estimation [J]. IEEE Transactions on Image Processing, 2000, 9(2):287~290.
- 4 Zhu C, Lin X, Chau L P. Hexagon-based search pattern for fast block motion estimation [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2002, 12(5):349~355.
- 5 Yoon H S, Lee G S. Motion estimation based on spatio-temporal correlations[A]. In: Proceedings of the International Conference on Image Processing[C], Barcelona, Spain, 2003, 2:359~362.
- 6 Yang J F, Chang S C, Chen C Y. Computation reduction for motion search in low rate video coders[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2002, 12(10):948~951.
- 7 Lengwehasarit K, Ortega A. Probabilistic partial-distance fast matching algorithms for motion estimation[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2001, 11(2):139~152.
- 8 Joint Video Team(JVT) Reference Software Version JM8.5[CP/OL]. <http://iphome.hhi.de/suehring/tml/download/>,2005-12-08.
- 9 Tu Y K, Yang J F, Shen Y N, et al. Fast variable-size block motion estimation using merging procedure with an adaptive threshold[A]. In: Proceedings of International Conference on Multimedia and Expo [C], Baltimore, MD, USA, 2003, 2:789~792.