

一种基于 H.264/AVC 的高性能快速运动估计算法

毛小明, 鲍可进

(江苏大学 计算机科学与通信工程学院, 江苏 镇江 212013)

摘 要: 为了降低运动估计的计算量, 提出一种基于 H.264/AVC 的快速运动估计算法。该算法使用了提前终止策略和自适应的搜索范围, 结合运动矢量预测以及多模板搜索。实验结果表明, 在编码性能接近全搜索 (full search, FS) 算法的同时, 本算法比 FS 和 UMHExagonS 算法平均节省了 65.42% 和 32.76% 的运动估计时间, 大幅度提高了编码速度。

关键词: H.264/AVC; 运动估计; 提前终止; 搜索范围; 运动矢量; 多模板搜索

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1001-3695(2012)04-1598-03

doi:10.3969/j.issn.1001-3695.2012.04.112

High-performance fast motion estimation algorithm for H.264/AVC

MAO Xiao-ming, BAO Ke-jin

(College of Computer Science & Telecommunication Engineering, Jiangsu University, Zhenjiang Jiangsu 212013, China)

Abstract: In order to reduce the computation of motion estimation (ME) module, this paper presented a novel fast ME algorithm for H.264/AVC with an early termination strategy and an adaptive search range method. The proposed algorithm adopted generalized motion vector predictors and multiple search patterns to ensure good quality. Experimental results show that the proposed algorithm can highly speed up video coding. It have saved 65.42% and 32.76% on average of motion estimation time compared to FS and UMHExagonS respectively, and have been negligible objective quality degradation compared to FS.

Key words: H.264/AVC; motion estimation; early termination; search range; motion vector; multiple search patterns

H.264/AVC 是由 ISO/IEC MPEG 和 ITU-T 共同制定的最新视频编码标准。与以往的标准相比, 在编码质量相同时, H.264/AVC 可以节约大约 50% 的码率。然而其编码效率的提高是以增加算法复杂度为代价的, 其复杂度大约是 H.263 的 4、5 倍^[1]。运动估计是视频压缩编码的关键部分, 其所花费的时间占整个 H.264/AVC 编码器编码时间的 60% (一个参考帧) ~ 80% (五个参考帧)^[2]。因此在牺牲较少质量的条件下, 使用快速运动估计算法就显得很有必要了。

为此, 研究者们提出了许多新的算法, 比较典型的有三步搜索 (three-step search) 法、钻石搜索 (diamond search) 法、二维对数搜索 (2D logarithm search) 法等。这些算法在处理小运动视频序列时效果比较好, 而对大运动视频序列效果不理想, 容易陷入局部最优陷阱。UMHexagonS 算法在一定程度上避免了这一问题, 但其算法的复杂度仍然较高。

近些年, 许多研究 (如文献 [3~6]) 致力于如何在运动估计过程中利用提前终止策略加速 H.264/AVC 的视频编码。在块匹配过程中, 当匹配代价小于事先定义好的提前终止门限值 (early termination threshold, ETT) 时, 可认为当前搜索的块为全局最佳并终止搜索, 节省了搜索点数, 从而加速运动估计的进程。可见, 为了保证编码性能的损失尽可能小, ETT 的设置就显得至关重要。

在 H.264/AVC 的运动估计过程中搜索范围一般都是固定的, 如果能够根据视频序列的特性动态调整搜索范围, 就可以更快地找到最佳运动矢量, 节省搜索点数, 从而降低编码的计算量。文献 [7~10] 提出的搜索范围的调整策略都是基于

同一帧内已编码相邻块的运动矢量 (MV) 的大小来改变搜索范围的。然而较大的 MV 并不意味着需要一个大的搜索范围, 这是因为搜索的起点不是原点, 而是相对原点位移等于预测矢量的点。

1 本文算法

本文针对以上讨论的问题提出了一个算法: 使用提前终止策略, 结合运动矢量预测以及不同的搜索模板在自适应的搜索范围内进行块匹配, 可以有效地降低运动估计的计算量。

在 H.264/AVC 中, 匹配代价函数定义为

$$J(mv, \lambda) = SAD(s, c(mv)) + \lambda R(mv - pmv) \quad (1)$$

其中: mv 表示与当前编码块进行匹配的块的 MV; pmv 是预测矢量; $SAD(s, c(mv))$ 表示两块匹配得到的绝对误差和; λ 为拉格朗日常数。本文算法中的匹配代价均由式 (1) 计算得到, 最终搜索得到的最小匹配代价记为 J_{min} , 对应的矢量为最佳 MV。

1.1 预测矢量的选择

当前编码块与位置相邻的块属于同一个物体或两者运动相同时, 它们之间存在很高的运动相关性。此时, 同一帧内的已编码相邻块的 MV 以及它们的中值预测矢量 MV_{median} 的预测准确性比较高。在本文中, 同一帧内已编码的相邻块为当前编码块的左、左上、上和右上的四个相邻块, 分别记为 a 、 b 、 c 、 d , 其运动矢量分别为 MV_a 、 MV_b 、 MV_c 和 MV_d 。通过式 (1) 分别计算 MV_a 、 MV_b 、 MV_c 、 MV_d 和 MV_{median} 处的匹配代价, 代价最小

收稿日期: 2011-07-12; 修回日期: 2011-09-09

作者简介: 毛小明 (1986-), 男, 江苏扬州人, 硕士研究生, 主要研究方向为信息处理与视频编码 (mxmhhh@163.com); 鲍可进 (1958-), 男, 教授, 硕士, 主要研究方向为嵌入式系统、信息处理与视频编码。

的矢量作为空域预测矢量,记为 PMV_{spa} 。

当前编码块与位置相邻的块属于不同的物体并且做着不同的运动时,它们的相关性比较低,此时 PMV_{spa} 的预测准确性不高。考虑物体运动的惯性,引入前一帧相同位置的块及其上下左右四个相邻块的 MV,并通过式(1)分别计算五个 MV 处的匹配代价,代价最小者作为时域预测矢量,记为 PMV_{tem} 。当以 PMV_{spa} 为起点进行搜索得到的最小匹配代价不小于 ETT 时,以 PMV_{tem} 为新起点进行搜索,取经过两次搜索得到的匹配代价最小的 MV 为最佳 MV。

设 N 表示 a, b, c, d 的存在数目 ($0 \leq N \leq 4$), $Is_Correlated$ 表示相邻块 MV 之间是否具有高相关性,其定义如下:

- a) 如果 $N \geq 3$, 并且对于 $\forall i, j \in \{a, b, c, d\}, i \neq j$, 都满足
- $$|MV_i - MV_j| \leq Cth$$

则相邻块的 MV 的相关性较高, $Is_Correlated = 1$ 。

b) 其他情况则相邻块的 MV 之间的相关性较低, $Is_Correlated = 0$ 。

1.2 ETT 的计算

H.264/AVC 支持七种不同尺寸和形状的宏块(尺寸为 16×16 像素)分割用于运动估计。首先可以将每个宏块分割为一个 16×16 , 或两个 16×8 , 或两个 8×16 , 或四个 8×8 的块;对每个 8×8 可以进一步划分为 8×4 , 4×8 和 4×4 这三种形状的块。

当前编码宏块的 J_{min} 与前一帧相同位置宏块的 J_{min} 相关性较高, 16×8 , 8×16 和 8×8 块的 J_{min} 与其所属宏块的 J_{min} 相关性较高, 8×4 , 4×8 和 4×4 块的 J_{min} 与其所属的 8×8 块的 J_{min} 相关性较高^[5]。因为相邻块的运动具有相关性,所以本文在计算 ETT 时也将它们的 J_{min} 考虑在内。设 $Th_1, Th_2, Th_3, Th_4, Th_5, Th_6$ 和 Th_7 分别表示 $16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8, 8 \times 4, 4 \times 8$ 和 4×4 块的 ETT, 当前帧的序号为 t , 16×16 块的 ETT 定义为

$$Th_1 = \min \{ J_{t-1}, J_{t(a)}, J_{t(b)}, J_{t(c)}, J_{t(d)} \} \quad (2)$$

其中: J_{t-1} 表示前一帧相同位置宏块的 J_{min} ; $J_{t(a)}, J_{t(b)}, J_{t(c)}, J_{t(d)}$ 分别表示相邻块 a, b, c, d 的 J_{min} 。式(2)的含义是取五个 J_{min} 中的最小值作为当前编码宏块的 ETT。

其他子块的 ETT 定义如下:

$$Th_i = C_i + \Delta_i, i \in [2, 7] \quad (3)$$

$$C_i = \begin{cases} \frac{J_{16 \times 16}}{S_i} & i = 2, 3, 4 \\ \frac{J_{8 \times 8}}{S_i} & i = 5, 6, 7 \end{cases} \quad (4)$$

$$S_i = \begin{cases} 2 & i = 2, 3, 5, 6 \\ 4 & i = 4, 7 \end{cases} \quad (5)$$

其中: $J_{16 \times 16}$ 和 $J_{8 \times 8}$ 分别为宏块 16×16 和 8×8 块的 J_{min} ; Δ_i 用来动态调整 Th_i 。如果终止某一子块搜索时得到的 J_{min} 比 Th_i 大,说明 Th_i 可能过低而没有发挥应有的作用,则需适当提高 Th_i 以避免搜索下一个同一模式子块时出现过低的情况;反之说明 Th_i 可能过高,得到的 J_{min} 可能比实际的全局最小值大许多,此时需适当降低 Th_i 。如果该子块的模式在当前宏块内首次被分析, $\Delta_i = 0$; 否则 $\Delta_i = \frac{1}{2}(J_{min} - Th_i)$ 。

1.3 搜索范围的计算

由于相邻块间的运动相关性,它们的 MV 大小接近。当位置相邻块的 MV 比较大时,当前编码块的 MV 较大的可能性高。然而一个大的 MV 并不意味着需要一个大的搜索范围,这

是因为搜索的起点不是原点,而是相对原点位移等于预测矢量的点。例如,预测矢量为 $(13, 12)$, 而当前编码块经过块匹配搜索后得到的最佳 MV 为 $(13, 12)$, 此时大小为 0 的搜索窗口足以搜索到最佳 MV。因此本文利用 MV_a, MV_b, MV_c 和 MV_d 与预测矢量的差值来计算搜索范围的大小。

当相邻块的相关程度高时,相邻块的 MV 与预测矢量的差值变化很小,反之则可能很大。为了尽可能避免因搜索范围过小而产生较大预测误差的情况发生,本文取 MV_a, MV_b, MV_c 和 MV_d 与预测矢量坐标差值的最大值作为搜索范围 $search_range$ 的大小,定义如下:

$$search_range = \max \{ |MV_{i(x)} - PMV_x|, |MV_{i(y)} - PMV_y| \} \quad (6)$$

$$i \in \{a, b, c, d\}$$

其中: $MV_{i(x)}$ 和 $MV_{i(y)}$ 表示 MV_i 的横坐标和纵坐标, PMV_x 和 PMV_y 分别为预测矢量的横坐标和纵坐标。式(6)计算得到的当前搜索窗口为边长等于 $2 \times search_range$ 的方形区域。

1.4 运动估计算法的描述

当相邻块的运动相关性较高时, PMV_{spa} 给出了比较精确的预测,此时根据 PMV_{spa} 的大小将当前编码块的运动分为中小运动 ($|PMV_{spa}| \leq 4$) 和大运动 ($|PMV_{spa}| > 4$)。根据运动的大小选择不同的搜索模板(图1~图3)。

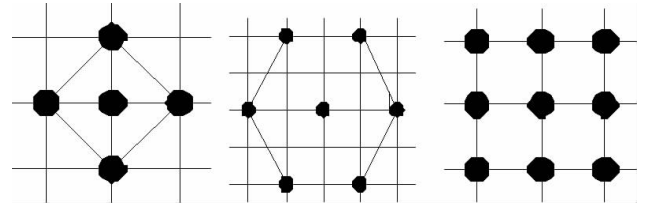


图1 小十字形模板

图2 六边形模板

图3 九点均布的矩形模板

本文算法在由式(6)计算得到的搜索范围中进行搜索,各尺寸块的 ETT 由 1.2 节中的公式计算得到。算法分下面三种情形进行描述:

- a) $Is_Correlated = 1$ 且 $|PMV_{spa}| \leq 4$ 。搜索流程如下:

(a) 以 PMV_{spa} 为起始搜索中心, 如果其匹配代价小于 ETT, 终止搜索; 否则进行步骤(b)。

(b) 进行一次步长为 1 的小十字形模板(图1)搜索。如果当前最佳(匹配代价最小)点的匹配代价小于 ETT 或为搜索中心, 终止搜索; 如果当前最佳点在搜索范围内, 以最佳点为中心重复步骤(b); 否则转至步骤(c)。

(c) 为了进一步避免因搜索范围过小而产生较大预测误差的情况发生, 搜索范围 $search_range++$, 以当前最佳点为中心进行小十字形模板搜索, 直到出现最佳点的匹配代价小于 ETT、最佳点在中心和最佳点在搜索范围外这三种情形之一为止, 终止搜索。

- b) $Is_Correlated = 1$ 且 $|PMV_{spa}| > 4$ 。搜索步骤如下:

(a) 以 PMV_{spa} 为起始搜索中心, 如果其匹配代价小于 ETT, 终止搜索; 否则进行步骤(b)。

(b) 进行一次边长为 2 的六边形模板(图2)搜索。如果当前最佳点的匹配代价小于 ETT, 终止搜索; 如果当前最佳点为搜索中心, 进行步骤(d); 如果当前最佳点在搜索范围内, 以最佳点为中心重复步骤(b); 否则跳转至步骤(c)。

(c) 基于情形 a) 中步骤(c)的相同原因, 搜索范围 $search_range++$, 以当前最佳点为中心进行六边形模板搜索, 直到出现最佳点的匹配代价小于 ETT、最佳点在中心和最佳点在搜索范围外这三种情形之一为止。如果当前最佳点的匹配代价小

于 ETT, 终止搜索; 否则进行步骤(d)。

(d) 以当前最佳点为中心进行一次九点均布的矩形模板(图3)搜索, 终止搜索。

c) $Is_Correlated = 0$ 。首先执行情形 b) 的搜索流程, 得到的最佳点的匹配代价记为 $J_{min(spa)}$ 。如果 $J_{min(spa)}$ 小于 ETT, 终止搜索; 否则令 $PMV_{spa} = PMV_{tem}$, 重复情形 b) 的搜索流程, 得到的最佳点匹配代价记为 $J_{min(tem)}$ 。如果 $J_{min(tem)}$ 小于 ETT, 则 $J_{min(tem)}$ 对应的 MV 为最佳 MV; 否则比较 $J_{min(spa)}$ 和 $J_{min(tem)}$, 取较小者对应的 MV 为最佳 MV。

2 实验及结果分析

本文算法是在 JM 9.6 代码中实现的, 通过与 UMHExagonS 以及 FS 算法进行比较来分析算法的性能。选取四个 CIF 格式的视频序列在量化系数 QP 分别为 20、24、28 和 32 的条件下编码 200 帧进行测试。选取的测试序列 (mother&daughter、foreman、bus、mobile) 具有较强的代表性。其中, mother&daughter 背景简单, 人物运动幅度不大, 属于小运动序列; foreman 是中等运动序列, 人物和镜头均运动; bus 是大运动序列; mobile 运动形式丰富, 画面有多个运动物体, 镜头也在移动, 且背景较为复杂。测试的帧频为 30 fps, 参考帧数取 5, 采用七种分块模式, 编码帧结构为 IPPPP, 搜索半径为 16 像素, 其余编码参数使用默认值。经过实验确定, 阈值 Cth 取值为 4。

实验对各算法编码后得到的 PSNR、码率和运动估计耗时进行了统计, 并计算了本文算法相对 FS 和 UMHExagonS 算法的 PSNR 增量 ($\Delta PSNR$)、码率增加百分比 ($\Delta rate$) 以及运动估计耗时增加百分比 (ΔMT), 并在表 1~4 中给出了相应的数据。表中的数据正值表示增加, 负值表示减少。

表 1 $QP = 20$ 时算法性能比较

测试 序列	$\Delta PSNR$		$\Delta rate/\%$		$\Delta MT/\%$	
	FS	UMHS	FS	UMHS	FS	UMHS
m&d	-0.007	-0.005	-0.186	-0.357	-60.19	-17.51
foreman	-0.004	0.006	0.544	0.311	-59.61	-27.72
bus	-0.003	0.002	0.473	0.376	-73.36	-35.81
mobile	-0.003	-0.001	0.065	0.081	-72.68	-38.37

表 2 $QP = 24$ 时算法性能比较

测试 序列	$\Delta PSNR$		$\Delta rate/\%$		$\Delta MT/\%$	
	FS	UMHS	FS	UMHS	FS	UMHS
m&d	-0.012	-0.01	-0.122	-0.034	-59.47	-22.92
foreman	-0.006	0.005	0.631	0.591	-57.61	-28.21
bus	-0.002	0.002	0.764	0.443	-73.76	-37.45
mobile	-0.004	-0.001	0.046	0.012	-73.33	-39.94

表 3 $QP = 28$ 时算法性能比较

测试 序列	$\Delta PSNR$		$\Delta rate/\%$		$\Delta MT/\%$	
	FS	UMHS	FS	UMHS	FS	UMHS
m&d	-0.01	-0.005	0.051	0.451	-57.99	-22.98
foreman	-0.006	0.011	1.317	1.203	-54.31	-29.65
bus	-0.003	0.003	1.211	0.812	-74.27	-39.14
mobile	-0.002	-0.001	0.119	0.076	-73.48	-41.09

表 4 $QP = 32$ 时算法性能比较

测试 序列	$\Delta PSNR$		$\Delta rate/\%$		$\Delta MT/\%$	
	FS	UMHS	FS	UMHS	FS	UMHS
m&d	-0.018	-0.011	1.433	1.141	-55.56	-23.05
foreman	-0.008	0.037	2.585	2.45	-51.74	-31.09
bus	-0.005	0.007	1.157	0.837	-75.58	-43.91
mobile	-0.004	-0.002	0.096	0.382	-73.81	-45.39

从表 1~4 可以看出, 对于不同运动大小和画面背景的视频率, 本文算法在不同量化系数下都能保持优异的压缩性能。与 FS 算法相比, 其 PSNR 最多降低 0.018, 平均降低 0.006, 对重建图像质量的影响基本可以忽略; 码率增加很小, 最大不超过 2.585%, 平均值为 0.637%, 压缩比基本不变; 运动估计耗时明显下降, 最多可节省 75.58%, 平均值为 65.42%。可见本文算法在保持与 FS 算法接近的编码性能的同时, 计算复杂度大幅降低。与 UMHExagonS 算法相比, 在重建图像质量和码率接近的情况下, 最多可减少 45.39% 的运动估计时间, 平均节省 32.76%。

3 结束语

本文针对 H.264/AVC 中运动估计的高计算复杂度问题, 提出在运动估计过程中使用自适应的搜索范围和提前终止策略以节省不必要的搜索点, 并且引入前一帧相同位置的块及其上下左右四个相邻块的 MV 以提高预测矢量的准确性, 同时基于预测矢量的大小采用不同模板进行搜索。实验结果表明, 与 FS 和 UMHExagonS 算法相比, 本文算法在保证编码质量的情况下, 有效地降低了 H.264/AVC 运动估计的复杂度。因此, 本文算法是一种适合 H.264/AVC 的高效快速运动估计算法。

参考文献:

- [1] LAPPALAINEN V, HALLAPURO A, HAMALAINEN T D, Optimization of emerging H.26L video encoder[C]//Proc of IEEE Workshop on Signal Processing Systems. 2001:406-415.
- [2] CHEN Zhi-bo, ZHOU Peng, HE Yun. JVT-F017, fast integer pel and fractional pel motion estimation for JVT[S]. 2002.
- [3] YANG Li-bo, YU K, LI Jiang, et al. An effective variable block-size early termination algorithm for H.264 video coding[J]. IEEE Trans on Circuits and Systems for Video Technology, 2005, 15(6): 784-788.
- [4] KIM S E, HAN J K. Efficient motion estimation using a modified early termination algorithm in H.264[J]. IEICE Trans on Information and Systems, 2005, E88-D(7): 1707-1715.
- [5] SARWER M G, WU Q M J. Adaptive variable block-size early motion estimation termination algorithm for H.264/AVC video coding standard[J]. IEEE Trans on Circuits and Systems for Video Technology, 2009, 19(8): 1196-1201.
- [6] ISMAIL Y, McNEELY J, SHAABAN M, et al. A generalized fast motion estimation algorithm using external and internal stop search techniques for H.264 video coding standard[C]//Proc of IEEE International Symposium on Circuits and Systems. 2008:3574-3577.
- [7] YAMADA T, IKEKAWA M, KURODA I. Fast and accurate motion estimation algorithm by adaptive search range and shape selection [C]//Proc of IEEE International Conference on Acoustics, Speech, and Signal Processing. 2005: II/897-II/900.
- [8] HONG M C, OH H H. JVT-B022, range decision for motion estimation of VCEG-N33[S]. 2002.
- [9] XU Xiao-zhong, HE Yun. JVT-Q088, modification of dynamic search range for JVT[S]. 2005.
- [10] PARK K S, KIM M J, CHO J H, et al. Search range selection method to reduce the complexity in H.264[C]//Proc of International Conference on Advanced Language Processing and Web Information Technology. Washington DC: IEEE Computer Society, 2008: 230-234.