

# H.264 帧间编码可变块阈值选择算法

夏定元<sup>1,2</sup>, 吴通<sup>1</sup>

(1. 武汉理工大学信息工程学院, 武汉 430070; 2. 桂林电子科技大学信息与通信学院, 桂林 541004)

**摘要:** 为有效预测运动矢量, 降低运动估计复杂度, 提出一种阈值选择算法。设定阈值以确定当前块是否为预测运动矢量块, 对最佳运动矢量块做出判定, 以提前终止运动估计。在 JM10.2 平台上与 UMHExagonS 算法进行对比测试, 验证该算法的性能。实验结果表明, 该算法缩短了约 84% 的编码时间, 提高了编码效率, 而视频质量无明显下降, 满足实时应用的需求。

**关键词:** 帧间编码; 运动估计; 阈值选择

## Threshold Selection Algorithm of Variable Block for H.264 Inter-coding

XIA Ding-yuan<sup>1,2</sup>, WU Tong<sup>1</sup>

(1. School of Information Engineering, Wuhan University of Technology, Wuhan 430070;

2. School of Information & Communication, Guilin University of Electronic Technology, Guilin 541004)

**【Abstract】** A threshold selection algorithm is presented to effectively predict vector and reduce the complexity of Motion Estimation(ME). Whether the current block is or not the Predicted Vector Block(PVB) is decided by setting up a threshold, and some decisions on the optimization PVB are made in order to terminate the ME process as early as possible. Compared with the UMHExagonS algorithm, the performance of the algorithm is testified on the JM10.2 platform. Experimental results show that the coding time is approximately saved by 84% and the coding efficiency is greatly raised while the degradation of video quality is negligible for real-time application demand.

**【Key words】** inter-coding; Motion Estimation(ME); threshold selection

可变大小块运动估计是 H.264 视频编码器中最为复杂和耗时的部分之一, 极大地制约了其实际应用<sup>[1]</sup>。目前主要有 2 种途径降低运动估计的运算量。一种是提升算法本身的效率。如今已有许多快速算法被提出来, 如 HBS, EPZS, UMHExagonS 等。另一种是运动估计的提前终止技术。这是因为在运动估计过程中, 许多块的 DCT 系数量化后为 0, 通过检测这些块, 可以有效地减少运算量。基于这一特点, 本文提出一种可变块阈值选择算法, 获得了良好效果。

### 1 可变块尺寸模式选择

H.264 中采用 7 种大小可变的块尺寸模式(16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4), 同时利用率失真策略对这 7 种模式进行遍历。相对于以往的编码标准, 这使得编码器可以根据图像中的运动情况灵活地选择块的大小, 从而实现高精度的运动估计<sup>[2]</sup>。在 JM 模型中, 运动估计过程的代码实现流程如下: 首先, 对当前块进行数据初始化; 然后, 根据运动矢量的空间相关性或时间相关性预测当前块的运动矢量, 以确定搜索起始点; 最后, 通过各种搜索模式搜索匹配块, 同时根据匹配准则确定最佳匹配块位置, 得到运动矢量<sup>[3]</sup>。其中, 运动矢量的预测利用了相邻块运动矢量的空间相关性和相邻帧运动矢量的时间相关性。

### 2 可变块阈值选择

#### 2.1 最佳运动矢量块预测

在 H.264 中, 当前块与相邻块位置关系如图 1 所示。通常, 当前块 E 的预测运动矢量可由当前块 E 的相邻左边块 A、上方块 B 和左上方块 C(或右上方块 D)三者运动矢量的中值

确定<sup>[4]</sup>。

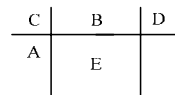


图 1 当前块与相邻块位置关系

因为相邻块的运动矢量通常都是相关的, 所以通过相关性得到的预测运动矢量很有可能就是块的最佳运动矢量。为了方便描述, 定义预测运动矢量块(Predicted Vector Block, PVB)就是最佳运动矢量块。事实上, 对于大部分序列, 在运动估计过后, 存在大量的 PVB, 当前块的预测运动矢量就是最佳运动矢量, 如表 1 所示。

表 1 各种模式中 PVB 所占比重 (%)

序列	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7
Akiyo	98.05	97.51	97.23	97.37	97.27	96.61	97.24
News	91.80	91.70	91.66	92.83	92.71	92.22	93.16
Tempete	86.23	86.17	86.01	87.34	88.72	90.54	92.27
Foreman	54.19	56.26	55.46	57.79	60.00	59.15	63.15
Mobile	50.86	50.68	50.35	51.22	51.38	52.69	54.53
Football	47.44	47.28	47.13	47.65	47.98	48.27	48.73
Bus	34.54	34.36	33.79	34.19	34.58	34.82	35.31

由表 1 可以看出, 在给定的 7 种模式中, PVB 在运动估计过程中占有相当大的比重, 特别是对于运动场景较小的

**基金项目:** 广西科学研究与技术开发计划基金资助项目(05112001-7A2)

**作者简介:** 夏定元(1963 - ), 男, 教授、博士, 主研方向: 多媒体通信, 多媒体信息处理, 图像处理与机器视觉; 吴通, 硕士研究生  
**收稿日期:** 2009-06-27 **E-mail:** labxiady@263.net

序列,如 Akiyo, News 和 Tempete 序列,比重超过 86%。因此,对 PVB 的检测可以跳过运动估计过程中的常规步骤,简化运动估计程序,降低运算复杂度,对缩短运动估计时间具有积极作用。

### 2.2 最佳阈值选择

在 H.264 中,以绝对误差和 SAD(Sum of Absolute Difference)匹配准则作为选择最佳匹配块的标准。在这里,用代价函数  $J$  来代替 SAD,这样可以在考虑预测精度的同时顾及到计算量的大小,因此更加合理<sup>[5]</sup>。代价函数  $J$  的公式如下:

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

其中,  $m=(m_x, m_y)^T$  是当前运动矢量;  $p=(p_x, p_y)^T$  是预测运动矢量;  $\lambda$  是拉格朗日系数;  $R(m-p)$  是编码运动矢量的比特数。

如果一个块是 PVB,那么一般在预测矢量位置的匹配块具有最小代价值。因此,分别为 7 种不同尺寸的块设定阈值  $THZ_i (i=1,2,\dots,7)$ 。在运动估计过程中,首先检测预测运动矢量点,如果满足下式,那么此块被认定为 PVB,其余的搜索点则可以跳过。

$$J_i < THZ_i \quad (i=1,2,\dots,7) \quad (2)$$

如何选定不同大小块的阈值是 H.264 中判定 PVB 的关键。很显然,阈值越大,PVB 越多,则可被跳过的搜索点也越多。然而,这样也会使很多块被错误地认定为 PVB 而使视频质量大大下降。因此,在算法性能和复杂度之间存在一个平衡问题。在实际应用中,防止视频质量的下降比降低运算复杂度更有意义。因此,以检测精度为向来选择阈值。在这里,精度代表当一个块的代价值小于阈值时它被判定为 PVB 的概率。本文的方法如下:首先通过大量试验得到基于不同精度的几个候选阈值,然后将视频质量和运算复杂度之间良好折中的阈值作为最佳选择。

对不同序列进行对比实验,结果表明,在阈值相同的情况下,小运动场景下的精度值(即块被判定为 PVB 的概率)要明显高于大运动场景下的值。在表 1 中,Foreman 序列代表中等变化的运动场景。因此,基于 Foreman 序列选择阈值,并将其运用到其他序列上,可以保证必要的检测精度。

表 2 给出了 Foreman 序列不同精度下各模式的候选阈值。

表 2 Foreman 序列不同精度下各模式候选阈值

精度/(%)	模式 1	模式 2	模式 3	模式 4	模式 5	模式 6	模式 7
80	600	550	570	400	315	300	250
75	960	650	660	480	360	350	270
70	1 650	920	890	600	410	400	300
65	2 270	1 200	1 160	700	470	460	350
60	2 800	1 500	1 430	880	530	530	400

在实际应用中,精度选为 66%,阈值(2 250, 1 180, 1 150, 690, 465, 460, 345)能够实现视频质量和运算量之间良好折中。另外,可将模式 1 的阈值选定为 800,以确定整个宏块搜索是否结束。

### 2.3 PVB 判定

根据选定的最佳阈值,对最佳运动矢量块做出预测,即对 PVB 进行判定,可提前终止运动估计。基本过程如下:如果块的预测运动矢量代价值小于相应的阈值,则认为预测运动矢量为最佳运动矢量,并跳至本宏块的下一块继续检测;如果当前块已经是宏块的最后一块,则跳至下一模式。当所有模式检测完毕,具有最小代价值的模式被选为最佳模式。特别地,在模式 1 中,因为宏块只有一个块,如果预测运动矢量代价值足够小,那么该预测运动矢量不仅是最佳运动矢

量,模式 1 也是本宏块的最佳模式。此时,选一个更小的阈值给模式 1,如果预测运动矢量代价值小于该阈值,则整个宏块搜索结束。

## 3 实验结果及分析

为了客观而全面地检验阈值选择算法在不同条件下的性能,根据表 1 中列出的运动状况不同的 7 种序列,在 JM10.2 平台上,与 H.264 中普遍采用的快速算法 UMHexagonS 进行对比测试。测试环境如表 3 所示。

表 3 阈值选择算法编码参数

RDO 模式	GOP 结构	搜索范围	参考帧	量化参数	熵编码
关闭	IPPP...	-16~+16	3	28, 32	CAVLC

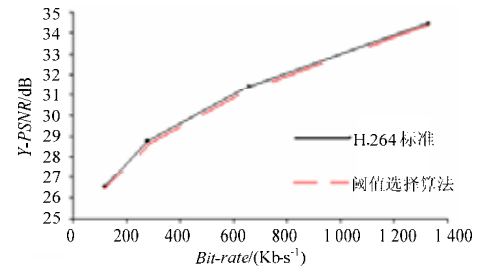
表 4 给出了不同量化参数  $QP$  下阈值选择算法与快速算法 UMHexagonS 的性能比较。其中,  $\Delta$  代表两者相比的差值;  $P$  代表峰值信噪比;  $B$  代表比特率;  $T$  代表编码时间。

表 4 阈值选择算法的性能

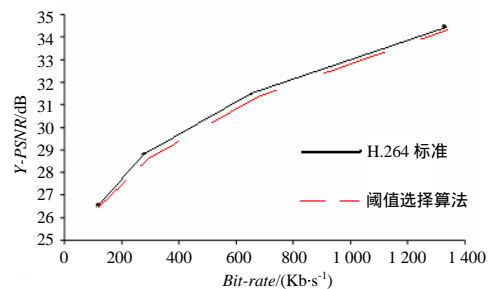
序列	$QP=28$			$QP=32$		
	$P$ /dB	$B$ /%	$T$ /%	$P$ /dB	$B$ /%	$T$ /%
Akiyo	-0.05	0.04	-94.12	-0.03	1.03	-93.39
News	-0.06	1.33	-83.30	-0.06	1.26	-83.48
Tempete	-0.05	1.72	-87.67	-0.06	2.63	-86.37
Foreman	-0.12	0.60	-81.76	-0.13	1.43	-80.61
Mobile	-0.14	1.58	-83.26	-0.14	1.64	-82.45
Football	-0.16	1.23	-84.96	-0.17	2.61	-83.84
Bus	-0.20	0.72	-80.65	-0.23	1.55	-79.78

由表 4 可以看出,与快速算法 UMHexagonS 相比,阈值选择算法在运动状况不同的各种序列中均表现出优异的性能,编码时间显著缩短,缩短幅度平均达到 84%,而峰值信噪比( $PSNR$ )的下降和比特率的上升极小,可以忽略不计,在极大地提高编码效率的同时,确保视频质量无明显下降。

图 2 给出了 2 组典型序列(Akiyo 序列和 Foreman 序列)的率失真(RD)曲线。



(a)Akiyo 序列



(b)Foreman 序列

图 2 2 组典型序列的 RD 曲线

由图 2 可以看出,在运动状况不同的 Akiyo 序列和 Foreman 序列中,阈值选择算法的 RD 曲线与 H.264 标准算法的 RD 曲线几乎重合, $PSNR$  指标相当,表明阈值选择算法不仅效率高,而且适应性强,视频质量得到保证。

(下转第 181 页)