

基于 H. 264 编码模式的快速判决算法

廖 怡 郭宝龙

(西安电子科技大学机电工程学院 ICE 研究所, 西安 710071)

摘 要 H. 264 在运动估计中引入了多种不同的编码模式, 使得模式选择过程异常复杂。针对这一问题, 本文提出了一个新概念——编码模式参考集, 并给出了一种基于区域预测的编码模式参考集选择方法。该方法可以大大减少候选参考模式的数目, 简化模式选择过程。并根据所选模式的特点给出了一种基于模式选择的自适应搜索策略。实验结果表明, 本文提出的新算法在重建图像质量没有明显下降的前提下, 编码速度有了显著提高。

关键词 H. 264 视频编码 运动估计 模式选择 NMDA

中图分类号: TN919.81 文献标识码: A 文章编号: 1006-8961(2008)01-0034-06

A Novel Mode Decision Algorithm Based on H. 264

LIAO Yi GUO Bao-long

(ICE Institute, School of Electromechanical Engineering, Xidian University, Xi'an 710071)

Abstract Variable coding modes are used in H. 264 which makes the mode selection a time-consuming component. To improve this situation, we first the concept of mode reference aggregate a mode reference aggregate decision algorithm is proposed based on region forecast. This method can significantly reduce the number of candidate mode, thus simplifying the procedure of mode decision. Experimental results show that the algorithm can increase the encoding speed significantly with negligible PSNR drop and bit-rate increase.

Keywords H. 264 video coding motion estimation mode decision NMDA

1 引言

H. 264^[1,2]是由国际电信联盟 (international telecommunication union, ITU) 和国际标准化组织 (international organization for standardization, ISO) 联合制订的新一代视频编码标准, 与以往的编码标准相比, H. 264 在编码性能上有了很大的提高, 其中一个关键因素是 H. 264 在运动估计中引入了多种不同的编码模式。但随着模式选择的增多, 运动估计算法的复杂度成倍增加。H. 264 目前采用的编码模式是编码器必须对每一个宏块所有的编码模式分别进行检验, 以确定最优模式。采用全部模式的全搜索运动估计占整个计算量的 80% 以上^[3], 这种全搜索方法给编码器带来了沉重的计算负担, 耗时相当大,

难以达到实时通信的要求。

近些年来, 针对模式选择问题提出了许多快速算法。文献 [4] 基于率失真代价对宏块进行分类, 并按一定的顺序对每个模式类型进行测试, 如果某个模式的编码代价小于阈值则停止对剩余模式的测试, 这种方法没有考虑相邻宏块模式之间的相关性。文献 [5] 使用当前帧和前一帧中已编码宏块的模式预测当前宏块的模式, 当前宏块的模式取代价为最小的相邻宏块的模式, 这种方法在时间相邻宏块中仅选用了前一帧中的一个宏块, 在这种情况下预测的模式不一定是全局最优, 从而因模式误判可能降低视频编码质量。

本文通过对 H. 264 标准中模式选择复杂度高的问题以及对文献 [4] 和文献 [5] 中算法的分析, 提出了基于 H. 264 编码模式的快速判决算法 (a novel

基金项目: 国家自然科学基金资助项目 (60572152)

收稿日期: 2006-02-28 改回日期: 2006-09-25

第一作者简介: 廖怡 (1982~), 女, 西安电子科技大学硕士研究生。研究方向为视频压缩、多媒体通信。E-mail: yigu0091@163.com

mode decision algorithm based on H. 264 NMDA)。新算法提出了编码模式参考集的概念, 并给出了一种基于区域预测的模式参考集选择方法, 同时通过分析编码模式的特点, 设计了一组适用于各种模式的搜索模板。实验结果表明, 该算法能很好地解决模式选择过程复杂的问题, 大大简化模式选择的过程。

2 H. 264 中的编码模式及模式选择方式

H. 264 在编码过程中, 首先将图像分割成若干个宏块, 然后对每一个宏块选择一种合适的编码模式进行编码。在 H. 264 中有多种不同的编码模式, 主要分为帧内预测和帧间预测两大类。

帧内预测是利用视频序列图像的空间相关性, 根据相邻块的系数进行预测以得到当前块预测值的一种方法。帧内预测模式包括 4×4 (记为 Intra4) 和 16×16 (记为 Intra16) 两种亮度尺寸。帧间预测主要是利用图像序列帧与帧之间的相关性, 通过运动补偿来消除视频图像的时间冗余。H. 264 标准中进行帧间预测时, 使用可变块运动估计技术, 每个宏块具有 4 种分割模式即 16×16 、 8×16 、 16×8 和 8×8 。若选择了 8×8 方式, 每个 8×8 块又具有 4 种分割模式即 8×8 、 8×4 、 4×8 和 4×4 。这种灵活而细微的模块划分一方面更切合图像中实际运动物体的形状, 能有效提高运动估计的准确性。另一方面, 细化的运动补偿所得到的图像块效应不明显, 具有优良的视觉效果。另外, 除了 7 种帧间编码模式之外, H. 264 在 P 帧还支持 skip 模式。skip 模式是一种特殊的 16×16 编码模式, 它有两个显著的特点, 一是其运动矢量为 $(0, 0)$ 或者为预测运动矢量; 二是其编码比特率为 0。

H. 264 采用了率失真优化^[6]策略对模式选择进行判定, 率失真代价函数 J 定义如式 (1)。在模式选择的过程中, 计算并比较所有可选模式的率失真代价, 代价最小的模式即为最优模式。

$$J_{\text{mode}}(s \text{ cl } Q, \lambda_{\text{mode}}) = \text{SSD}(s \text{ cl } Q) + \lambda_{\text{mode}} R(s \text{ cl } Q) \quad (1)$$

式中, s 是当前块像素值, c 为重建块像素值, R 包括对宏块编码的所有比特, Q 为量化步长, λ_{mode} 为拉格朗日乘法器, SSD (sum of square error) 为当前块与重建块差值的平方和。

3 基于区域预测的编码模式参考集选择方法

H. 264 在编码的过程中, 需要对每一个宏块所有可能的模式进行检验, 然后通过比较率失真代价来确定最优编码模式。这种基于率失真的模式判决运算量相当大, 降低其运算量的有效途径之一就是减少候选模式的数目。而本文就是基于这一思想提出了编码模式参考集的概念, 将所有可能的编码模式分成若干个模式参考集, 编码过程中只对某一个选中参考集中的编码模式进行检验, 从而减少了候选编码模式的数目。另外考虑到编码模式的选择和图像运动特性有很大的关系: 静止和运动简单的区域采用较大块模式进行编码, 而运动复杂的区域则选择较小的块模式, 基于这一点本文提出了基于区域预测的模式参考集选择方法。

3.1 编码模式参考集

表 1 列出了 4 种代表性序列编码模式的分布情况。从表中可以看出, 在纹理简单, 背景变化不大的图像序列中, 如 Container 和 Foreman skip 模式和较大的块模式所占的比率较大; 在纹理细腻, 变化明显的序列 Bus 和 Mobile 中, 较小块模式所占比例最大达 40% 以上; 并且在帧间预测中采用帧内模式的概率很小。因此, 根据分布情况以及模式之间的相近程度, 可以将模式集合划分成 4 种模式类型 $S_1 \sim S_4$, 即上面提到的编码模式参考集: skip 模式运算简单且不需要编码, 可单独作为一个参考集, 即 $S_1 = \{\text{skip}\}$; 将 16×16 、 16×8 和 8×16 归纳为第 2 个模式参考集 S_2 , 称为宏块类, $S_2 = \{16 \times 16, 16 \times 8, 8 \times 16\}$; 第 3 个参考集 S_3 包括子块类中的 8×8 、 8×4 、 4×8 和 4×4 4 种模式, 即 $S_3 = \{8 \times 8, 8 \times 4, 4 \times 8, 4 \times 4\}$; 帧内模式作为一类, 记为第 4 参考集 S_4 , $S_4 = \{\text{Intra4}, \text{Intra16}\}$ 。根据各种模式计算的难易程度以及被选中使用的概率, 将这 4 个模式参考集按处理优先级排列成为 S_1 、 S_2 、 S_3 和 S_4 。

表 1 不同序列的编码模式所占百分比

Tab 1 Percent of selected modes in different sequences

序列	skip	16×16	16×8	8×16	8×8	Intra16	Intra4
Container	75.8	10.4	3.5	2.7	7.3	0.3	0
Foreman	23.7	39.9	39.9	7.3	7.6	7.3	9.3
Bus	4.5	22.0	12.1	14.4	40.4	1.0	5.5
Mobile	5.5	31.3	7.1	6.1	49.7	0	0.3

在提出编码模式参考集的概念之后, 模式选择问题就转变成首先从 4 个候选参考集中确定一个模式参考集, 然后只对所选参考集内的可能模式进行检验。那么如何正确选择参考集将成为一个至关重要的步骤。接下来将提出一种通过判断宏块所处区域来确定模式参考集的方法。

3.2 基于区域预测的编码模式参考集选择方法

由上面的分析可知, 图像采用的编码模式与其内容有很大关系, 根据它们之间存在的关系, 本文将图像分为静止区域、平滑区域以及动态区域 3 个区域。由于视频图像序列在时间和空间上都存在较大的相关性, 特别是对于背景简单, 低速运动的序列(如头肩序列), 这种时空相关性尤为明显。因此利用时空相关性对宏块所属区域进行预测, 进而确定当前宏块的模式参考集。

3.2.1 利用时间相关性进行区域预测

设 $B_n(x, y)$ 为第 n 帧图像中的一个待编码宏块, 该宏块左上角的像素位于图像的 (x, y) 处, $B_{n-1}(x, y)$ 为第 $n-1$ 帧中相同位置的宏块。由于视频序列在时间上具有很强的相关性, 即 $B_n(x, y)$ 的图像特征与 $B_{n-1}(x, y)$ 有很大的关系, 因此可以利用参考帧中对应位置的宏块来预测当前宏块所属的图像区域。为此在参考帧中为当前宏块 $B_n(x, y)$ 定义一个参考区域:

$$f_n = \{B_{n-1}(x, y) \mid |x| < \hat{v}_x^{\text{motion}}, |y| < \hat{v}_y^{\text{motion}}\} \quad (2)$$

式中, $\hat{v}_x^{\text{motion}}$ 和 $\hat{v}_y^{\text{motion}}$ 分别为 x 方向和 y 方向上的预测运动矢量分量。

如果参考区域 f_n 中的所有宏块均采用 16×16 的帧间编码模式, 则称区域 f_n 为静止区域, 那么当前宏块 $B_n(x, y)$ 也可以认为属于静止区域, 此时略过其他的编码模式, 只对该宏块采用 16×16 的帧间编码模式。

3.2.2 利用空间相关性进行区域预测

经过上面对静止区域的判定, 可以比较精确的将运动平缓的背景区域区分出来, 然后对它们采用 16×16 的帧间模式进行编码, 这样不仅能保证很高的精度, 而且还可以避免对一些不必要的编码模式进行检测。但是如果参考区域 f_n 中有一个或多个宏块不属于静止区域, 那么将无法判断当前宏块 $B_n(x, y)$ 是否属于静止区域。针对这一问题, 本文提出了一种利用序列图像的空间相关性进行预测的方法。在这种方法中, 利用 16×16 帧间编码模式下的 SAD(sum of absolute difference)来判断 $B_n(x, y)$ 属于静止区域、平滑区域还是动态区域。对于整幅

图像来说, SAD 与宏块属于哪个区域并没有直接的关系, 但是在局部的小区域, 属于静止区域的宏块的 SAD 值明显小于属于动态区域宏块的 SAD 值, 应用这一点对宏块所属区域进行判定。定义 $B_n(x, y)$ 在 16×16 帧间编码模式下的 SAD 为

$$SAD(B_n(x, y)) = \sum_{i=0}^{15} \sum_{j=0}^{15} |Y(x+i, y+j) - \hat{Y}(x+i+u, y+j+v)| \quad (3)$$

式中, Y 和 \hat{Y} 分别为当前图像和参考图像的灰度值, (u, v) 为宏块的运动矢量。

设 $B_n(x, y)$ 为当前待编码宏块, A, B, C, D 为与 $B_n(x, y)$ 相邻的已编码宏块。为衡量当前编码块的运动情况, 定义一个新的变量:

$$SAD_{\text{AVG}} = \frac{1}{n} \sum_{i=0}^{n-1} SAD(X_i) \quad (4)$$

其中, n 为 A, B, C, D 中采用 16×16 帧间编码模式的宏块个数, $X_i \in \{A, B, C, D\}$ 然后将 $SAD(B_n(x, y))$ 与 SAD_{AVG} 进行比较, 如果 $SAD(B_n(x, y)) < SAD_{\text{AVG}}$, 认为宏块 $B_n(x, y)$ 属于平滑区域; 如果 $SAD(B_n(x, y)) > SAD_{\text{AVG}}$, 则认为宏块属于动态区域。如果 A, B, C, D 中没有宏块采用 16×16 的编码模式, 那么在对当前宏块 $B_n(x, y)$ 的编码过程中仍需考虑所有可能的帧间编码模式以确定最优帧间编码模式。

3.2.3 基于区域预测的模式选择

在确定了宏块所属区域之后, 根据图像区域与编码模式之间的关系, 可以为当前宏块选择一个与之相对应的编码模式参考集。若宏块属于静止区域, 则只对当前宏块采用 16×16 的帧间编码模式; 若宏块属于平滑区域, 说明当前宏块的运动较平缓、纹理较简单, 此时应采用较大的块模式进行编码, 因此选用宏块类 S_2 作为编码模式参考集; 若宏块属于动态区域, 说明该宏块纹理较细腻, 包含较多的运动细节, 应采用较小的块模式进行运动估计, 因此选择子块类 S_3 中的编码模式。选择合适的模式参考集之后, 只对所选参考集内的模式进行检验以确定最优模式。具体方法按照式(1)计算所选编码模式参考集中各编码模式的率失真代价, 代价最小的模式即为所选的最优帧间模式。

4 基于模式选择的自适应搜索

4.1 NMDA 算法的搜索模板

运动估计时间的长短取决于搜索点数的多少,

目前对运动估计算法的优化有很大一部分都是基于“减少搜索区域内待匹配点数”这一原理,即在保证搜索精度的前提下,减少搜索点数以提高搜索速度。目前比较典型的算法有:菱形搜索法(DS)、六边形搜索法(HEXBS)等。这些改进的搜索算法与 H. 264 的全搜索算法相比,可以大幅度减少搜索点数,提高运动估计的效率。但是这些算法并不是针对 H. 264 的运动估计提出的,对于 H. 264 不同于其他视频编码标准的一些新特点,这些算法存在着一些欠缺。

在 H. 264 编码标准中,为了提高运动估计的精度,采用了 7 种帧间编码模式。DS 搜索和 HEXBS 搜索法对于形状为正方形的块模式(16×16 、 8×8 、 4×4)覆盖效果比较好,因为它们的搜索范围接近于一个圆。但对于 16×8 、 8×16 、 8×4 、 4×8 的块模式,由于块的长宽不一致,覆盖效果并不是很好。

针对这一情况,本文设计了一种基于模式选择的自适应六边形搜索模板,如图 1 所示,采用自适应的六边形搜索模板作为大搜索模式,不受限的小菱形作为小搜索模式进行精确搜索;同时六边形可根据所选模式的不同形状和大小自适应地改变形状和大小,以达到最佳的搜索效果。具体搜索步骤如下:对于形状为正方形的块(16×16 、 8×8 、 4×4)采用菱形搜索模板(HSP);对于采用 16×8 或 8×4 编码模式的块采用水平搜索模板(HHSP);对于 8×16 或 4×8 的块则采用垂直搜索模板(VHSP);当搜索达到一定精度时改用小菱形模板(SHSP)进行精确搜索。

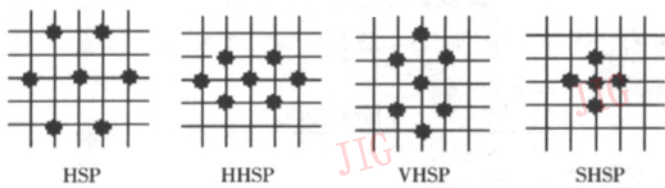


图 1 自适应六边形搜索模板

Fig 1 Selfadaptive Hexagon Search Pattern

4.2 NMDA 算法的搜索策略

基于区域预测的模式参考集选择方法已经大大减少了候选模式的数量,但是它只针对帧间编码模式的选择。NMDA 算法考虑所有可能的编码模式,根据 4 个模式参考集的优先级,首先进行 skip 模式的检验,然后对不能采用 skip 模式的宏块利用基于区域预测的模式参考集选择方法确定最优帧间编码模式,如果必要,最后再检验帧内模式所在的参考集

S_4 , 从而最终确定最优编码模式。

综上所述,整个 NMDA 算法可以概括为以下几个步骤:

(1) 检验 skip 模式,计算 skip 模式下的 MSE_{skip} (mean square error)。判断 MSE_{skip} 是否等于 Q 若是,则 skip 模式为最优编码模式,跳转至步骤 (7); 否则进入步骤 (2);

(2) 检验 16×16 的帧间模式,并计算 $MSE_{16 \times 16}$ 。比较 MSE_{skip} 与 $MSE_{16 \times 16}$ 的大小,若 $MSE_{16 \times 16} > MSE_{skip}$, 选择 skip 为最优编码模式,跳转至步骤 (7); 否则进入步骤 (3);

(3) 根据图像序列的时间相关性进行静止区域的预测,若宏块属于静止区域,选择 16×16 作为最优帧间编码模式,跳转至步骤 (5); 否则进入步骤 (4);

(4) 利用空间相关性进行平滑区域的预测,若宏块属于平滑区域,选择宏块类 S_2 作为编码模式参考集;若宏块不属于平滑区域,采用较小的块模式进行预测,选择子块类 S_3 作为编码模式参考集;然后按照率失真优化公式计算所选参考集中各种编码模式下的率失真代价,代价最小的即为最优帧间编码模式,进入步骤 (5);

(5) 最优帧间编码模式已确定。判断 MSE_{skip} 是否大于阈值 T , 如果是进入步骤 (6); 否则跳过帧内编码模式的检测,步骤 (4) 中确定的最优帧间编码模式即为最终所选的编码模式,然后跳转至步骤 (7);

(6) 比较帧间编码模式的绝对误差和 SAD_{inter} 与帧内编码模式的绝对误差和 SAD_{intra} , 若 $SAD_{inter} < SAD_{intra}$, 上面确定的帧间模式为最优编码模式; 否则最优编码模式为帧内模式。然后进入步骤 (7);

(7) 最优编码模式已确定。然后根据所选的编码模式动态选择搜索模板进行搜索。若编码模式为 16×16 、 8×8 、 4×4 中的一种,采用 HSP 进行搜索;若编码模式为 16×8 或 8×4 采用 HHSP 进行搜索;若编码模式为 8×16 或 4×8 采用 VHSP 搜索;当最小率失真代价点位于上述 3 种模板的中心时,转入步骤 (8); 否则循环执行步骤 (7);

(8) 改用 SHSP 进行精确搜索,直至最小率失真点位于小菱形模板中心处时停止搜索,该中心点即为最优匹配点,它所对应的运动矢量为最优运动矢量。

5 实验结果

本文的实验采用了 JVT (joint video team) 的 JM90 (joint model 90) 作为实验平台, 主要的编码参数设置如下: 搜索范围为 16, 采用 Hadamard 变换, 使用 CABAC 编码, 图像序列以 30 帧 / s 的速度进行编码, 量化步长 QP 值分别选取 24, 28 和 32, 采用 5 个参考帧, 使用率失真优化策略。为了检验新算法

的性能, 选取全搜索算法 (Full) 与 NMDA 算法进行性能比对。实验选用 3 个 QCIF (quarter common intermediate format) 图像序列: Carphone (380 帧), Silent (298 帧), Foreman (389 帧) 和一个 CIF (common intermediate format) 图像序列: Mobile (298 帧)。表 2 是不同的图像序列采用两种不同算法在不同 QP 值时的峰值信噪比 (PSNR), 平均每帧的运动估计时间以及平均比特率的性能比对。

表 2 FULL 与 NMDA 的性能比对
Tab 2 Performance comparison of Full and NMDA

序列	QP	PSNR (dB)			时间 (ms)			比特率 (kbps)		
		Full	NMDA	$\Delta(F)$	Full	NMDA	(N/F)%	Full	NMDA	$\Delta(F)$ %
Carphone	24	39.776	39.743	-0.033	845.860	69.678	8.22	10502.8	10484.6	+0.76
	28	36.859	36.807	-0.052	610.566	51.278	8.46	6336.0	6239.5	+0.80
	32	33.962	33.870	-0.092	637.004	56.646	8.86	3633.0	3662.8	+1.36
Silent	24	38.970	38.950	-0.026	69.888	36.026	7.83	5774.3	5673.7	+0.81
	28	35.914	35.903	-0.011	52.278	41.383	8.61	3582.0	3521.0	+1.09
	32	33.094	33.070	-0.024	56.730	41.500	8.48	2129.6	2150.9	+1.00
Foreman	24	38.275	38.190	-0.085	630.162	62.769	9.91	10469.0	10672.8	+0.97
	28	35.672	35.650	-0.022	622.467	62.977	10.10	5929.5	5975.1	+0.94
	32	32.999	32.916	-0.083	612.744	67.296	10.97	3297.5	3277.1	+0.58
Mobile	24	37.246	37.242	-0.004	1799.070	195.576	10.83	93396.7	93326.4	-0.08
	28	33.907	33.900	-0.007	1796.731	197.100	10.95	54857.4	54860.7	+0.01
	32	30.641	30.615	-0.026	1840.549	207.980	11.13	28835.3	28836.8	+0.01

从表 2 可以看出, NMDA 算法的 PSNR 值与全搜索法相比下降特别小 (平均降低 0.038), 说明重建图像的质量与原图像非常接近, 同时比特率增加也特别小 (平均增加 0.70%), 但是用于运动估计的时间大大减少了, 速度有了显著的提升, 平均节省约 90% 的时间。分析上面的数据可知, 对于运动较小的低比特率图像序列, 如 Silent 和 Carphone, 速度提高的幅度尤为明显, 因为低速运动的图像序列较多的采用了编码速度较快的 skip 模式以及较大的块模式进行运动估计; 而对于细节较丰富, 纹理较细腻的 Mobile 序列, 由于采用了较小的块模式进行编码, 因此运动估计的精度较高, 得到的重建图像的质量也就较好, 表现在表 2 中就是运动估计后的 PSNR 值几乎与用全搜索法进行搜索后的 PSNR 值相等。图 2 和图 3 分别为 NMDA 算法与全搜索 Full 算法预测精度和预测速度的一个比较。从图 2 中可以看

到代表两种不同算法 PSNR 值的两条曲线是相互交织的, 说明采用 NMDA 算法进行预测的精度与使用

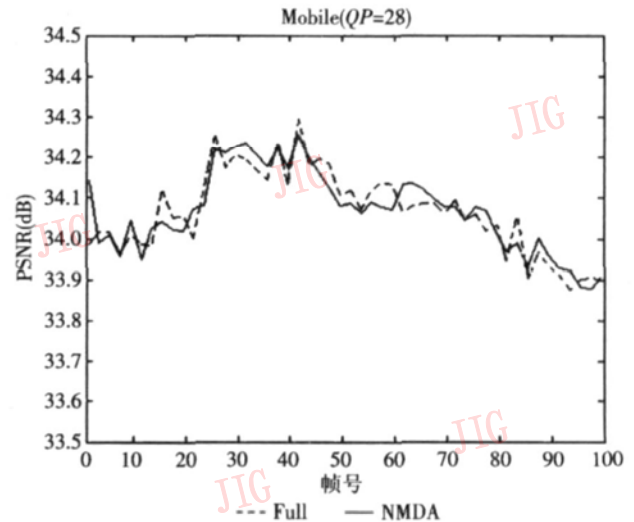


图 2 两种算法的预测精度比较

Fig 2 Prediction precision comparison of full and NMDA

全搜索算法的预测精度是不相上下的。而图 3 则反映了 NMDA 算法在编码速度上的明显优势。

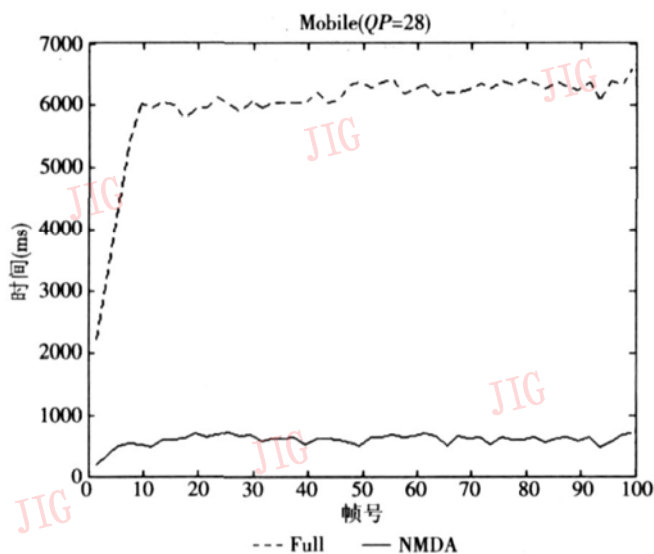


图 3 两种算法的预测速度比较

Fig 3 Prediction speed comparison of full and NMDA

6 结 论

本文在研究现有 H. 264 编码模式选择算法的基础上, 提出了编码模式候选集的概念以及一种新的模式判决算法 (NMDA), 新算法大大减少了候选参考模式的数目, 解决了以往算法容易陷入局部最优的问题。实验结果表明, 该算法在保证编码质量

的同时, 运动估计速度有了显著的提高, 更有利于实时应用, 是一种综合性能优良的快速运动估计算法。而本文在研究中提出来的模式参考集的概念以及对图像区域划分的方法对于这类算法具有一定的实际意义。

参考文献 (References)

- 1 ITU-T Recommend H. 264 / ISO / IEC 11496-10 Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (Advanced Video Coding) [S], 2003
- 2 Thomas Wiegand Overview of the H. 264 video coding standard [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13(7): 1~19
- 3 Lappalainen Hailapuro A, Hanalainen T Optimization of emerging H. 26L video encoder [A]. In Proceeding of IEEE Work on Signal Processing Systems Design and Implementation [C], Antwerp Belgium: WSES Press, 2001, 9: 26~28, 406~415
- 4 Jeyun Lee, Byeungwoo Jeon Fast mode decision for H. 264 with variable motion block sizes [A]. In Proceedings of International Symposium on Computer and Information Sciences [C], Antalya, Turkey, 2003: 723~730
- 5 Ahmad A. Selection of variable block sizes in H. 264 [A]. In Proceedings of International Conference on Acoustics, Speech, and Signal Processing [C], Quebec, Canada, 2004: 319~321
- 6 Wiegand T, Schwarz H, Joch A, et al Rate constrained coder control and comparison of video coding standards [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2003, 13: 688~703