

一种面向 H.264/AVC 的运动估计算法

王 泉 程嘉利

(西安电子科技大学计算机外部设备研究所, 西安 710071)

摘 要 提出了一种使用具有方向偏向性搜索模板和基于预测模式自适应选择运动估计方案的 EPZS (enhanced predictive zonal search) 改进算法。该算法将运动估计过程与 H.264/AVC 中多参考帧、多预测模式等编码工具紧密结合, 充分利用视频中运动具有高度时空相关性的特点, 综合考虑速度、图像质量及压缩效率等方面的性能。实验结果表明, 该算法以极小的计算量得到了与全搜索方法接近的效果; 与 EPZS 算法相比, 运动估计的处理速度有了较大幅度的提高, 同时保持了与之相当、有时甚至更优的率失真性能。此外, 该算法具有很好的鲁棒性, 并且能够产生平滑的运动矢量场。

关键词 视频编码 H.264/AVC 运动估计 运动矢量 模式选择 率失真优化

中图法分类号: TN919.81 文献标识码: A 文章编号: 1006-8961(2008)06-1054-07

A Fast Motion Estimation Algorithm for H.264/AVC

WANG Quan, CHENG Jia-li

(Research Institute of Computer Peripherals, Xidian University, Xi'an 710071)

Abstract An improved EPZS (enhanced predictive zonal search) algorithm with direction-based search patterns and adaptive motion estimation scheme selection based on prediction block type is presented in this paper. Utilizing high special/temporal correlation of motions in video sequences and focusing on the interactions between motion estimation and new coding tools such as multiple-references and prediction block types, the algorithm can obtain almost the same PSNR as full search at very low computation cost. Compared with EPZS, it can considerably improve the speed of motion estimation, and rate-distortion performance. Experiments verify the superiority of the proposed algorithm. Furthermore, the proposed algorithm is robust and capable of generating a smooth motion vector field.

Keywords video coding, H.264/AVC, motion estimation, motion vector, mode decision, rate-distortion optimization

1 引 言

块匹配运动估计方法 (block-matching motion estimation) 被广泛应用于 MPEG-1/2/4、ITU-T H.261/263/263+ 以及 JVT H.264/AVC 等多个视频编码标准中。全搜索法 (FS) 虽然能够获得非常高的搜索精度, 但是其巨大的运算量令人难以接受。因此, 快速运动估计成为了多年来视频编码领域内的研究热点, 许多算法被相继提出, 其中比较典型的

有: 三步搜索法 (TSS)^[1]、2 维对数法 (LOGS)^[2]、十字型搜索法 (CS)^[3]、新三步搜索法 (NTSS)^[4]、四步搜索法 (FSS)^[5]、梯度下降法 (BBGDS)^[6]、钻石搜索法 (DS)^[7] 和六边型搜索法 (HEXBS)^[8] 等。这些方法虽然在搜索速度方面比 FS 有了很大的提高, 但是普遍存在以下几个问题: (1) 搜索时容易陷入局部最优; (2) 图像质量与 FS 相比仍不够好; (3) 通过最小化匹配误差来得到运动矢量, 运动矢量的搜索独立于后续的 DCT 变换、量化及熵编码等过程, 特别是没有把编码运动矢量本身需要的比特数列入考

基金项目: 陕西省自然科学基金项目 (2006F48)

收稿日期: 2007-07-13; 改回日期: 2007-12-05

第一作者简介: 王 泉 (1970 ~), 男, 副教授。1997 年于西安电子科技大学获计算机器件与设备专业工学硕士学位。主要研究方向为输入输出技术与设备、图像处理与计算机视觉。E-mail: qwang@xidian.edu.cn

虑范围,这显然不能达到编码器的优化;(4)这些方法(同时也包括FS)因为在搜索运动矢量时没有考虑到邻近块的信息,导致产生的运动场是相当杂乱无章的,而杂乱的运动场不利于数据传输时的误差防护。

新一代视频编码标准 H.264/AVC 引入了许多新的编码工具,除了采用多参考帧之外,对每一个宏块的帧间预测编码可以采用多达7种不同大小的预测块类型,每种预测块类型分别对应一种帧间预测模式,如图1所示。

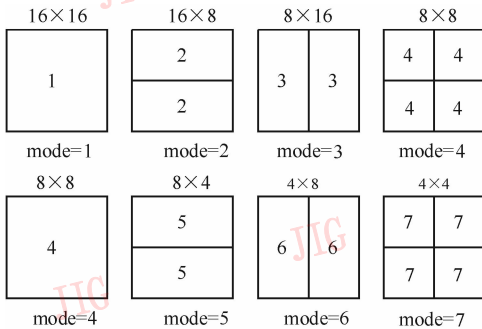


图1 H.264/AVC 中的7种帧间预测模式

Fig.1 Seven inter prediction modes in H.264/AVC

编码器对待编码宏块的每一种帧间模式都进行运动估计,并在运动估计和模式选择(mode decision)过程中引入拉格朗日率失真优化方法^[9](Lagrangian rate distortion optimization)以实现码率/编码器控制(bit-rate/coder control)。特别地,通过求取下式的最小值来得到各模式的运动矢量:

$$J(\mathbf{m}, \lambda_{\text{motion}}) = SAD(s, c(\mathbf{m})) + \lambda_{\text{motion}} \cdot R(\mathbf{m} - \mathbf{p}) \quad (1)$$

式中, $\mathbf{m} = (m_x, m_y)^T$ 为候选的运动矢量, λ_{motion} 为拉格朗日算子(Lagrange multiplier), $R(\mathbf{m} - \mathbf{p})$ 代表了对运动矢量进行差分编码可能消耗的比特数。 $SAD(s, c(\mathbf{m}))$ 为源视频信号 s 和编码视频信号 c 之间的绝对误差和,用来衡量块匹配误差,其计算如下式:

$$SAD(s, c(\mathbf{m})) = \sum_{x=1, y=1}^{B_1, B_2} |s[x, y] - c[x - m_x, y - m_y]| \quad (2)$$

$B_1, B_2 = 16, 8 \text{ 或 } 4$

这些新工具、新方法使得 H.264/AVC 较之前的标准极大地提高了编码效率,但也大大增加了运动估计的复杂度,并对运动估计算法的性能提出了更高的要求。目前已被 H.264/AVC 所采纳的运动估计算法有:EPZS 算法^[10]、UMHexagonS 算

法^[11]及其改进算法 Simplified UMHexagonS。这些算法具有优良的率失真性能,且大大提高了运动估计速度,它们代表了当今块匹配运动估计的最新研究成果。在研究各种运动估计算法尤其是 EPZS 的基础上,提出一种使用具有方向偏向性搜索模板和基于预测模式自适应选择运动估计方案的 EPZS 改进算法。

2 EPZS 简介及本文算法的主要思想

2.1 EPZS 算法简介

EPZS 算法首先利用视频运动中存在的高度时/空相关性,对当前编码块的运动矢量进行预测:选择一系列高度相关的、可靠的(reliable)已编码块的运动矢量作为当前运动矢量的候选预测,通过最小化代价函数来获得最优预测值。其次,利用可靠的自适应提前退出准则实现搜索过程的提前退出以提高运动估计速度。如果没有提前退出,最后再利用简单、规则的搜索模板在最优预测位置周围进行进一步的运动矢量精确定位。

2.2 本文算法主要思想

本文在研究和实验的基础上提出以下几方面的算法思想:

(1)运动估计与新编码工具的结合 本文算法关注模式选择过程与运动估计的相互影响,考虑如何更好地与多参考帧、多预测模式相结合以及如何利用各参考帧、各模式的运动矢量间存在的相关性。之前很多算法中将运动估计与多参考帧、多模式的处理分开来考虑的做法是不合适的,综合考虑这些方面会使编码的复杂度/性能均有更大的获益。

(2)基于预测模式自适应选择运动估计方案之前的大部分运动估计算法(包括 EPZS、UMHexagonS)都是视频/图像层面上的,即对于各种预测块类型采用大致相同的处理方法。本文算法则是基于块层面的,即对不同的预测块类型采用不同的处理方案:如针对不同的块类型采用不尽相同的候选预测选择方法、不同形状和大小搜索模板等。这样的做法对各预测模式的不同特点予以考虑,使得算法更加灵活、高效、有针对性。

(3)通过预测,自适应选择水平/垂直偏向搜索模板 大部分已有算法(包括 EPZS)的搜索模板在水平、垂直方向上的搜索点分布是均匀的,即认为待

搜索的运动矢量代表的运动是偏重于水平方向还是偏重于垂直方向的可能性是相等的。如果当前运动矢量的预测值(即当前块的相邻、相关块的运动矢量)所表示的运动是水平偏重的,那么当前块的运动很有可能也是水平偏重的;反之如果预测值所表示的运动是垂直偏重的,那么当前块的运动很有可能也是垂直偏重的。因此为了能够更快地搜索到运动矢量,同时也为了有助于产生平滑连续的运动矢量场,这里提出 4 种具有方向偏向性的搜索模板,并根据运动矢量预测值的方向来预测当前运动矢量偏重于哪个方向,自适应地选择偏重于水平/垂直运动方向的模板。

3 算法具体描述

3.1 候选预测选择方案

算法中首先定义一个候选预测值的集合:

$$S = \{MV_1, MV_2, \dots, MV_n\} \quad (3)$$

并按重要性把这个候选预测集划分为多个子集。各模式通用的预测子集包括:由中值预测单独组成的子集 S_1 、由空间预测值组成的子集 S_2 、由空

间内存消耗预测值^[10]组成的子集 S_3 及由时域预测值组成的子集 S_4 。

此外,可以针对不同模式的特点选择一些有些模式采用而另一些模式不采用的预测子集,也可以利用其他参考帧、其他模式的信息来对当前参考帧、当前模式的运动矢量进行预测。本文算法采取以下几个选择方案:

(1) 尺寸较大的预测块包含的像素比较多,相比小尺寸块的情况,这些像素处于同一运动的概率一般来说小很多,而且由于更多的像素参与到匹配误差的计算中,误差值也相对大很多。所以在搜索这些块的运动矢量时,比小尺寸块时的情况更加“担心”是否能搜索到最优位置、是否会陷入局部最优。为此对大尺寸预测块采用更多一些的预测信息,搜索范围也更广一些。这里,对它们采用在搜索窗内分布范围比较广的一些固定位置作为候选预测子集 S_5 。

(2) 利用 H. 264/AVC 以从模式 1 到模式 7 的等级顺序进行各预测模式的运动估计的特点,可以选择上层块如模式 5 或 6 是模式 7 的上层;模式 4 是模式 5 和 6 的上层等)的运动矢量作为下层块运动矢量的候选预测值,如图 2^[11] 所示。

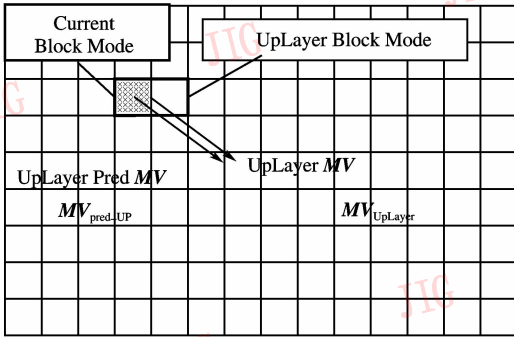
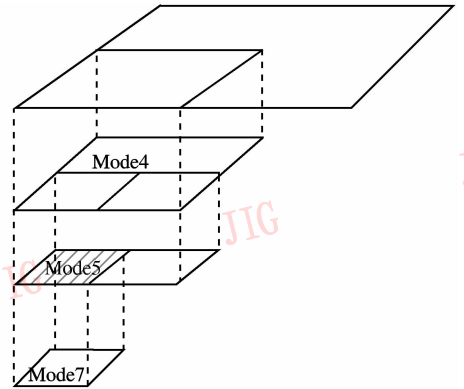


图 2 上层块的运动矢量作为下层块运动矢量的候选预测值

Fig. 2 Spatial uplayer prediction of motion vector



(3) 利用之前检测过的参考帧 0 和 ref_idx-1 上的运动矢量在时域上缩放后的值作为当前参考帧 ref_idx 上运动矢量的候选预测值^[10]。选择方案 2、3 候选预测值组成子集 S_6 。

当然,根据当前块及邻近块的情况以及应用对于图像质量及复杂度等方面的需求,可以发掘其他方面的相关性,增加一些候选预测;也可以去掉或修改一些上述的候选预测以达到降低复杂度或节省内存等目的。

3.2 提前退出策略

首先引入两个自适应的阈值 T_1 、 T_2 。

$$T_1 = 2^{bitdepth-8} \times N_p \quad (4)$$

式中, N_p 为当前预测块中像素的数目, $bitdepth$ 代表当前序列的位深。

$$T_2 = a \times \min(\max(\min(J_A, J_B, J_C), T_{\min}), T_{\max}) + b \quad (5)$$

式中, a 、 b 为固定值, J_A 、 J_B 、 J_C 为当前块的邻近块(左、上、右上)根据式(1)计算出的代价值,

$$T_{\min} = c \times 2^{\text{bitdepth}-8} \times N_p \quad (6)$$

$$T_{\max} = d \times 2^{\text{bitdepth}-8} \times N_p \quad (7)$$

c, d 均为常量。 T_1, T_2 可以灵活选取, 这里仅给出一种参考。

算法首先检测候选预测子集 S_1 , 根据式(1)计算这个中值预测的运动估计代价, 如果代价值小于阈值 T_1 , 则立即结束整个整像素运动估计过程, 而不再检测任何其他候选预测子集。此时, 中值预测就是当前模式的最终整像素运动矢量。如果不满足阈值为 T_1 时的提前退出条件, 与 EPZS 不同, 本文算法按子集 S_2, S_3, \dots, S_6 的顺序依次对各子集进行检测, 在一个子集中的所有预测值都检测之后进行判断, 如果此时的最小代价已小于阈值 T_2 , 则提前退出, 不再对后续的各子集进行检测; 否则再继续检测下一个子集。这就是将 S 按重要性划分成各子集的原因。实验结果证明, 这样的做法会在一定程度上提高运动估计速度而不会对预测精度造成影响。因为重要性越高的子集中包含最优预测的可能

性越高, 如果在重要性高的子集检测完毕之后发现此时的最小代价已小于 T_2 , 则认为与该最小代价对应的预测值已经是对当前运动矢量的足够准确的预测, 那么再对后续重要性较低子集进行检测以期找到“更优”预测值的希望是比较小的。

3.3 运动矢量精确定位过程

如果没有提前退出, 运动矢量需要被进一步精确定位。这个精确定位过程即模板迭代搜索过程, 是很多传统的非预测型运动估计算法的整个过程, 在预测型算法中也是一个重要的组成部分。

3.3.1 本文算法的搜索模板

提出以下 4 种新的搜索模板: 偏重水平方向的大加强六边形模板 (LEH-HBP)、偏重垂直方向的大加强六边形模板 (LEH-VBP)、偏重水平方向的小加强六边形模板 (SEH-HBP) 以及偏重垂直方向的小加强六边形模板 (SEH-VBP), 连同 3 种已有的无方向偏向的模板, 本文算法共使用如图 3 所示的 7 种搜索模板。

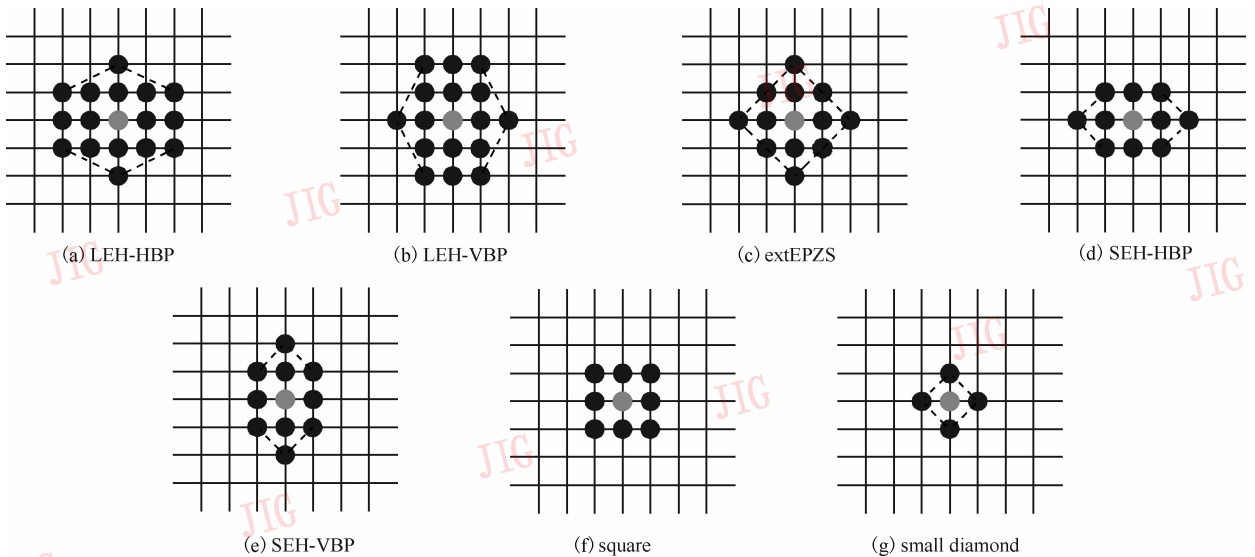


图 3 本文算法所使用的 7 种模板

Fig. 3 Seven search patterns used in the proposed algorithm

模板的选择准则如下:

(1) 如果当前进行运动估计的预测块大小是 16×16 像素、 16×8 像素或 8×16 像素, 因为它们尺寸比较大, 基于 3.1 节所述的原因, 对它们采用较大覆盖范围的模板。接下来判断最优预测值的水平和垂直分量的大小关系: 如果水平分量大于垂直分量, 则选用 LEH-HBP 模板; 如果水平分量小于垂直分量, 则选用 LEH-VBP 模板; 如果二者相等, 则选用

无偏向的 extEPZS 模板。

(2) 如果当前的预测块大小是 8×8 像素、 8×4 像素或 4×8 像素, 则对它们采用中等覆盖范围的模板。接着同样需要判断最优预测值的水平和垂直分量的大小关系以确定选用 SEH-HBP、SEH-VBP 以及无偏向的正方形模板中的哪一个。

(3) 如果当前的预测块大小是 4×4 像素, 可以认为之前的预测过程产生的最优预测值已经很精

确,故采用复杂度最低的小钻石模板。

需要说明的是,采用具有方向偏向性的模板仅仅是认为通过预测,运动矢量偏向哪个方向的可能性较高则可以增加 2 个或 4 个偏重于这个方向的搜索点(相比均匀分布的情况),而在另一个方向上的搜索点数并没有比采用均匀模板时有所减少,也是会被充分考虑到的。例如,LEH-HBP 模板共有 16 个搜索点,其中 8 个偏重于水平方向,4 个偏重于垂直方向,其余 4 个无偏重。所以,本文算法的模板设计是经过全面考虑的,虽有偏向,但并不“偏激”,是不会轻易陷入局部最小而影响效果的,后面的实验结果也证明了这一点。

采用各种模板进行搜索的过程是类似的,即首先以最优预测值对应的位置作为模板的中心,此时中心位置处的运动估计代价是当前的最小代价(J_{\min}),中心处也是当前的最优匹配位置。检测模板上的各个搜索位置,根据式(1)计算它们的运动估计代价,这些位置(包括中心)中哪个的代价最小,则这个位置被选为新的最优匹配位置, J_{\min} 也被更新为这个位置所对应的代价。接着模板以这个新的最优匹配位置为新的中心进行搜索。如此重复进行模板的迭代,直到最小代价 J_{\min} 在某次模板搜索的前后都是当前模板的中心对应的代价时,整个迭代过程停止。此时这个中心位置即为(一次)精确定位过程最终得到的最优匹配位置。

这里,本文算法在模板的迭代过程中加入另一个自适应提前退出阈值 T_3 ,令其值等于 T_2 。在一次的模板迭代结束之后,将当前的最优匹配位置对应的代价即当前的 J_{\min} 与 T_3 进行比较:如果小于 T_3 ,则提前退出,不再以这个位置为模板中心进行下一次迭代,也不再进行后面提到的二次定位过程;否则继续进行下一次的模板迭代。

3.3.2 二次定位过程

引入二次定位过程是为了进一步降低陷入局部最小的可能性,这个过程与 3.3.1 节中所述过程完全类似,只不过这个过程搜索的第 1 步以第 2 优预测值作为模板中心。与 EPZS 不同,本文算法为了进一步减少不必要的计算,对是否进行二次定位添加了如下的条件:

(1) 如果第 2 优预测值在水平和垂直方向上与最优预测值的距离都小于 3 个像素,则不进行二次

定位。因为这时二次定位的搜索过程经过的位置很有可能与一次定位过程经过的大部分是重复的,且这两个过程最终确定的最优匹配位置也很可能是相同的。

(2) 基于与条件 1 中所述相同的原因,如果第 2 优预测值在水平和垂直方向上与一次定位过程确定的最优匹配位置的距离都小于 3 个像素,则不进行二次定位。

4 实验结果及性能分析

实验的测试环境为 Intel Pentium 4 3.00GHz;内存为 512MB;操作系统为 Windows XP Professional;参考软件为 JM12.2。

实验中的主要参数设置如下:档次设为 High Profile;参考帧数设为 5;搜索范围设为 32;采用率失真优化方法、全部 7 种帧间预测模式、全部 6 个候选预测子集以及二次定位过程;采用 I-B-P-B-P-B-P... 的编码顺序。

为了对本文算法的性能进行综合评价,选取格式为 CIF 或 QCIF 的多个被各种算法所广泛使用的视频序列进行测试,全面比较了在不同帧率、不同量化参数(QP)下这些视频序列采用本文算法、FS 以及 EPZS 进行运动估计时的图像质量、码率以及运动估计处理速度,其中图像质量采取客观评价准则——峰值信噪比(PSNR, Y 分量)。

本文算法与 FS 及 EPZS 的比较结果如表 1、表 2 所示。表 1 中结果说明,本文算法在大幅度提高运动估计速度的同时获得了与全搜索方法接近的效果,在一些情况下(如 Bus_cif 序列, $QP = 28$ 时)的 PSNR 甚至超过了全搜索方法。

从表 2 结果可以看出,相比 EPZS,本文算法的运动估计处理速度有了显著的提高,对几乎所有测试序列提高都在百分之十几(最长达 17%),但同时保持了与之相当的率失真性能。一些情况下(如 News_qcif, $QP = 28$ 时、Silent_qcif, $QP = 36$ 时、Mobile_cif, $QP = 36$ 时以及 Flower_cif, $QP = 28$ 时)在比 EPZS 获得更好(相同)图像质量的同时使用更少的编码比特数。从大量的测试数据可知,本文算法在处理各种不同内容、不同格式、不同帧率以及不同量化参数的视频序列时都保持了稳定的优良性能。

表 1 本文算法与全搜索算法的比较

Tab.1 Comparison of the proposed algorithm and FS

测试序列	OP	FS			本文算法			ME 增速 (%)	PSNR 增益 (dB)	比特率 节省 (%)
		PSNRY (dB)	比特率 (kbps)	ME 耗 时(s)	PSNRY (dB)	比特率 (kbps)	ME 耗 时(s)			
Container_qcif	28	36.69	13.64	1 050	36.69	13.74	19.984	52	0.00	-0.73
帧率:10.0	32	33.90	7.49	835	33.89	7.56	20.585	40	-0.01	-0.93
测试 100 帧	36	31.15	4.33	618	31.15	4.33	22.364	27	0.00	0.00
Silent_qcif	28	36.17	43.56	2 529	36.17	43.99	48.947	51	0.00	-0.99
帧率:15.0	32	33.09	23.82	2 050	33.09	23.83	50.542	40	0.00	-0.04
测试 200 帧	36	30.57	12.76	1 657	30.56	12.86	52.765	30	-0.01	-0.78
Mobile_cif	28	34.89	2 002.21	26 160	34.88	2 009.68	444.323	58	-0.01	-0.37
帧率:30.0	32	31.00	1 070.86	21 258	30.99	1 074.44	452.900	46	-0.01	-0.33
测试 300 帧	36	27.67	530.31	19 973	27.66	533.48	461.652	42	-0.01	-0.60
Tempete_cif	28	35.44	1 496.34	18 430	35.44	1 501.50	352.394	51	0.00	-0.34
帧率:30.0	32	31.89	722.91	15 323	31.88	724.29	357.219	42	-0.01	-0.19
测试 260 帧	36	28.96	329.58	11 710	28.94	330.06	358.117	32	-0.02	-0.15
Bus_cif	28	35.40	1 559.72	13 237	35.41	1 577.24	245.805	53	0.01	-1.12
帧率:30.0	32	32.01	846.62	10 747	31.99	857.36	251.342	42	-0.02	-1.27
测试 150 帧	36	29.04	444.26	8 699	29.03	452.97	257.069	33	-0.01	-1.96

表 2 本文算法与 EPZS 算法的比较

Tab.2 Comparison of the proposed algorithm and EPZS

测试序列	QP	EPZS			本文算法			ME 增速 (%)	PSNR 增益 (dB)	比特率 节省 (%)
		PSNRY (dB)	比特率 (kbps)	ME 耗 时(s)	PSNRY (dB)	比特率 (kbps)	ME 耗 时(s)			
Foreman_qcif	28	36.30	51.18	37.871	36.31	51.34	33.127	14.3	0.01	-0.31
帧率:10.0	32	33.43	27.11	37.103	33.41	27.39	32.556	14.0	-0.02	-1.03
测试 100 帧	36	30.80	15.13	36.436	30.81	15.23	32.781	11.1	0.01	-0.66
News_qcif	28	37.04	30.93	52.404	37.05	30.92	46.946	11.6	0.01	0.03
帧率:10.0	32	33.68	18.00	55.815	33.66	17.97	47.525	17.4	-0.02	0.17
测试 200 帧	36	30.65	10.02	53.943	30.63	9.99	48.817	10.5	-0.02	0.30
Silent_qcif	28	36.16	43.82	54.519	36.17	43.99	48.947	11.4	0.01	-0.39
帧率:15.0	32	33.11	23.84	54.850	33.09	23.83	50.542	8.5	-0.02	0.04
测试 200 帧	36	30.55	12.87	57.663	30.56	12.86	52.765	9.3	0.01	0.08
Mobile_cif	28	34.89	2 008.47	500.272	34.88	2 009.68	444.323	12.6	-0.01	-0.06
帧率:30.0	32	30.99	1 073.22	506.049	30.99	1 074.44	452.900	11.7	0.00	-0.11
测试 300 帧	36	27.66	534.17	509.265	27.66	533.48	461.652	10.3	0.00	0.13
Tempete_cif	28	35.44	1 499.50	400.232	35.44	1 501.50	352.394	13.6	0.00	-0.13
帧率:30.0	32	31.88	723.95	406.336	31.88	724.29	357.219	13.7	0.00	-0.05
测试 260 帧	36	28.95	329.81	396.230	28.94	330.06	358.117	10.6	-0.01	-0.08
Stefan_cif	28	36.06	1 585.69	141.490	36.07	1 586.66	126.694	11.7	0.01	-0.06
帧率:30.0	32	32.29	817.78	138.810	32.30	821.93	125.855	10.3	0.01	-0.51
测试 90 帧	36	29.17	411.22	141.382	29.15	411.78	126.566	11.7	-0.02	-0.14
Flower_cif	28	35.84	2 026.24	364.462	35.84	2 026.11	327.450	11.3	0.00	0.01
帧率:30.0	32	31.77	1 126.61	375.871	31.78	1 127.32	341.747	10.0	0.01	-0.06
测试 250 帧	36	28.24	590.65	379.980	28.25	592.65	345.678	9.9	0.01	-0.34

5 结 论

在大量研究与实验的基础上,提出了 4 种新的搜索模板和几种新的运动估计策略,并利用它们对 EPZS 算法进行了多方面的改进。实验结果表明,本文算法以非常小的计算量得到了与全搜索方法非常接近的效果;与 EPZS 相比,运动估计的处理速度有了显著的提高,同时保持了与之相当、有时甚至更优的率失真性能。此外,本文算法还具有以下的特点:(1)具有很好的鲁棒性;(2)算法是可升级的,并且可以根据编码环境及应用需求做出调整;(3)能够产生平滑的运动矢量场,因此在对运动矢量进行差分编码时需要较少的比特数,这对数据分割及易错环境中的误差防护是非常有益的。

参考文献 (References)

- 1 Koga T, Iinuma K, Hirano A, *et al.* Motion compensated interframe coding for video conferencing [A]. In: Proceedings of the National Telecommunications Conference [C], New Orleans, LA, USA, 1981: G5.3.1 ~ G5.3.5.
- 2 Jain J R, Jain A K. Displacement measurement and its application in interframe image coding [J]. IEEE Transactions on Communications, 1981, **COM-29**(12): 1799 ~ 1808.
- 3 Ghanbari M. The cross-search algorithm for motion estimation [J]. IEEE Transactions on Communications, 1990, **38**(7): 950 ~ 953.
- 4 Li Ren-xiang, Zeng Bing, Liou Ming L. A new three-step search algorithm for block motion estimation [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1994, **4**(4): 438 ~ 442.
- 5 Po Lai-man, Ma Wing-chung. A novel four-step search algorithm for fast block motion estimation [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1996, **6**(3): 313 ~ 317.
- 6 Liu Lurng-kuo, Feig Ephraim. A block-based gradient descent search algorithm for block motion estimation in video coding [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1996, **6**(4): 419 ~ 422.
- 7 Zhu Shan, Ma Kai-kuang. A new diamond search algorithm for fast block-matching motion estimation [J]. IEEE Transactions on Image Processing, 2000, **9**(2): 287 ~ 290.
- 8 Zhu Ce, Lin Xiao, Chau Lappui, *et al.* Enhanced hexagonal search for fast block motion estimation [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2004, **14**(10): 1210 ~ 1214.
- 9 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**(7): 688 ~ 703.
- 10 Tourapis Alexis Michael, Cheong Hye-Yeon, Topiwala Pankaj. Fast ME in the JM reference software [EB/OL]. http://ftp3.itu.ch/av-arch/jvt-site/2005_07_Poznan/JVT-P026r1.doc, 2005-07-28.
- 11 Chen Zhibo, Zhou Peng, He Yun. Fast motion estimation for JVT [EB/OL]. http://ftp3.itu.ch/av-arch/jvt-site/2003_12_Waikoloa/JVT-J027.doc, 2003-12-05.