

基于双起点十字搜索模型的自适应搜索算法

王双双, 端木春江, 周冬辉, 刘 菡

(浙江师范大学数理与信息工程学院, 金华 321004)

摘要: 提出一种基于双起点十字搜索模型的自适应搜索算法。该算法采用大小十字搜索模型、八边形搜索模型和斜交叉十字搜索模型, 根据不同情况, 使用相应的搜索模型, 同时采用中值预测及提前跳出策略, 减少不必要的搜索。序列测试结果表明, 与 UMHexagonS 算法相比, 新算法在保持相当的峰值信噪比和比特率的情况下, 运动估计时间平均节省了 15.88% 和 15.94%。

关键词: 运动估计; 大小十字搜索; UMHexagonS 算法

Adaptive Search Algorithm Based on Double-initial-cross Search Pattern

WANG Shuang-shuang, DUANMU Chun-jiang, ZHOU Dong-hui, LIU Han

(College of Mathematics Physics and Information Engineering, Zhejiang Normal University, Jinhua 321004)

【Abstract】 This paper proposes an adaptive search algorithm based on double-initial-cross. The algorithm employs big-small cross search pattern, octagon and quasi-cross search pattern for adapting to different block content. Median prediction and early skip strategy are used to reduce unnecessary search. Compared with UMHexagonS, simulation results of the sequences present that this algorithm can save motion estimation time of average 15.88% and 15.94% while obtaining similar *PSNR* and *Bitrate*.

【Key words】 motion estimation; big-small cross search; UMHexagonS algorithm

1 概述

良好的运动估计算法是提高 H.264/AVC 编码效率最重要的环节。运动估计通过减少视频序列中帧内的空间和帧间的时间相关性, 从而达到较高的压缩比。近年来, 各种各样的运动估计算法先后涌现, 全搜索算法(FS)具有较好的图像质量, 但是其巨大的耗电量和高强度的计算复杂度使它不具备实用性。随之出现的 TSS, 4SS, DS 和 HEXBS^[1]等在保持与 FS 质量相当的情况下都不同程度地减少了计算复杂度, 但是在大运动块中易陷入局部最优。现在已经被 H.264/AVC 参考软件采纳的 UMHexagonS 算法性能较好, 能获得与 FS 算法非常接近的峰值信噪比(*PSNR*), 其估计时间也大大降低。

根据 H.264/AVC 中块运动的特征以及运动模式判定的原理, 本文提出一种对起始预测点和原点同时进行大小十字搜索的算法, 根据不同位置的点动态改变搜索策略, 在保持精度的同时, 减少了冗余搜索。

2 基于双起点十字搜索模型的自适应搜索算法

本文的算法结合了运动起始点预测、运动搜索模型和半路终止策略等技术, 增强了运动估计算法的鲁棒性和有效性, 主要包括: (1)起始点预测和提前跳出策略; (2)原点(0, 0)和预测起始点同时进行大小双十字模型搜索; (3)多步半路终止策略和混合模型搜索。

2.1 算法步骤

算法步骤如下:

(1)起始点预测及提前终止: 使用中值预测方法确定初始点, 并判断是否满足提前终止策略, 如果满足, 则转入(8), 否则转入(2)。

(2)大小十字搜索: 以上一步的预测点和原点(0, 0)为中心

同时进行大小十字搜索, 转入(3)。

(3)模型判别: 取 2 个大小十字搜索的最小值 *SAD*, 然后断此 SAD_{\min} 是否满足半路终止条件, 满足转入(8), 否则判断此 SAD_{\min} 的位置; 如果在大小十字中心, 则转入(4), 否则转入(5)。

(4)斜交叉十字搜索: 以上一步得到的最佳匹配点为中心, 进行斜交叉十字搜索, 转入(8)。

(5)八边形搜索: 以上一步得到的最小 *SAD* 为中心, 进行八边形搜索^[2], 如果最小 *SAD* 点为中心点, 转入(6), 否则重复此步骤。

(6)大十字搜索: 计算大十字模型上的 4 个点 *SAD* 值, 如果最小 *SAD* 点在仍在中心, 则转入(7), 否则转入(5)。

(7)小十字搜索: 以上一步得到的点为中心计算小十字模型上的 4 个点 *SAD* 值, 如果最小 *SAD* 值在中心, 转入(8), 否则继续此循环。

(8)得到最佳点, 算法结束。

2.2 初始点预测

视频序列中运动矢量具有较强的时空相关性, 且一帧中所有的块都是按光栅顺序存放, 因此, 当前块的矢量与左方 A、上方 B 和右上方 C 3 个宏块的矢量最为密切, 利用这些特点对当前宏块的运动矢量进行初始点预测。当块运动较大时, 初始点的预测可以提前找到匹配块, 缩减搜索步骤和时间。此算法使用当前块左方、上方和右上方的块矢量来预测当前块的运动矢量, 其预测方法与 UMHexagonS 一样仍然使

作者简介: 王双双(1984-), 女, 硕士研究生, 主研方向: 视频压缩编码; 端木春江, 副教授; 周冬辉、刘 菡, 硕士研究生

收稿日期: 2009-07-09 **E-mail:** wangzlyz@163.com

用中值预测, 当前块及相邻块位置如图 1 所示, 其中, MBc 为当前匹配块。

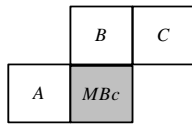


图 1 当前块及相邻块位置

2.3 搜索终止策略

由文献[3]可知, 图像中静止块或准静止块占主导地位, 则大部分宏块的运动矢量都接近零运动矢量。且对于大小为 16×16 的静止块的平均 SAD 值经常在 $600 \sim 1300$ 间波动^[4], 当阈值 $T=512$ 时, 重建后的图像峰值信噪比与 $T=0$ 时相当。因此, 设定一个阈值 T , 在搜索过程中, 如果某点的 $SAD < T$, 则判定此点为最佳匹配, 停止搜索; 如果 $SAD > T$, 则继续进行相应的模式搜索。算法中的 T 设为 512。

2.4 自适应搜索模板

小十字是最简单的搜索模型, 已被应用于许多的快速匹配算法中, 但其不利于大运动矢量的搜索, 本文使用大小十字相结合的方法, 首先进行原点和起始预测点的搜索, 判断其最小 SAD 点的位置是否在任一搜索中心, 如果是, 则认为此块为小运动块, 再进行一次斜交叉十字搜索即可; 反之, 则认为此块为中等或大运动块, 采用八边形、大十字和小十字模型循环搜索, 算法使用八边形模型来代替六边形, 其更多的搜索点数可以提高搜索的精度和速度。搜索模型见图 2。

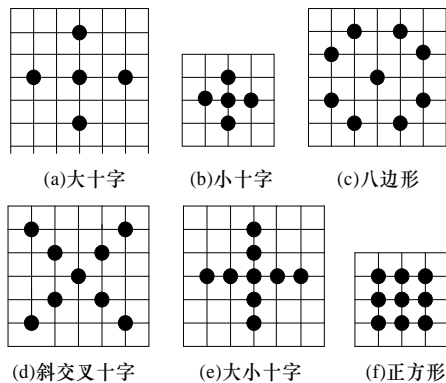


图 2 本文算法采用的搜索模型

2.5 正方形一次搜索算法

在算法的第 7 步, 如果将小十字搜索模板改成正方形搜索, 只要进行 1 次, 无需多次循环, 即可减少搜索的复杂度, 但图像质量稍有下降。

3 实验结果与分析

本文算法在 H.264/AVC 参考模型 JM12.2^[5] 上实现, 并与 FS 和 UMHEX 算法进行比较, 实验使用了 6 个较有代表性的 QCIF 序列, 分别为 Akiyo(298 帧)、Salesman(447 帧)、Highway(1 998 帧)、Carphone(380 帧)、Coastguard(300 帧)和 Mobile(298 帧)。实验采用测试序列的前 100 帧, 帧率为 30 Hz, 搜索窗口为 $[-16, +16]$, 量化参数均为 28。算法从重建图像的平均 PSNR、比特率(Bitrate)和运动估计时间 3 个方面与 FS 和 UMHEX 进行了比较。

不同算法在不同序列中的 PSNR 比较如表 1 所示。不同算法在不同序列中所用运动估计时间的比较如表 2 所示。其中, QCIF 序列大小为 177×144 ; PPS 表示小十字搜索算法; PPS1 表示正方形一次搜索算法。

表 1 不同算法在不同序列中的 PSNR 比较

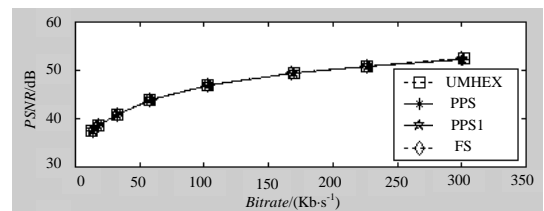
序列	PSNR/dB			
	FS	UMHexagonS	PPS	PPS1
Akiyo	38.54	38.51(-0.03)	38.54(-0.00)	38.54(-0.00)
Salesman	35.76	35.75(-0.01)	37.75(-0.01)	35.74(-0.02)
Highway	37.87	37.83(-0.05)	37.83(-0.05)	37.80(-0.07)
Carphone	37.61	37.56(-0.05)	37.51(-0.10)	37.52(-0.09)
Coastguard	35.07	35.06(-0.01)	35.07(-0.00)	35.06(-0.01)
Mobile	34.34	34.34(-0.00)	34.33(-0.01)	34.33(-0.01)

表 2 不同算法在不同序列中所用运动估计时间的比较

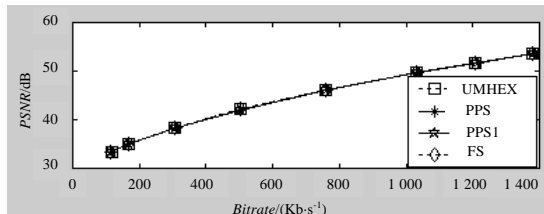
序列	运动估计时间/ms			运动估计时间增量(%)	
	UMHexagonS	PPS	PPS1	PPS	PPS1
Akiyo	22.575	18.986	19.062	15.90	15.56
Salesman	30.396	26.534	26.480	12.71	12.88
Highway	31.940	28.707	28.588	10.21	10.49
Carphone	38.493	32.603	31.596	15.30	17.91
Coastguard	51.924	41.533	42.159	20.01	18.80
Mobile	54.467	43.543	43.560	21.23	20.02

当 $QP=28$ 时, 本文算法对于运动较小的序列如 Akiyo, Coastguard 具有较高的质量, 只有序列 Carphone 质量略有下降, 其余的均与 UMHEX 算法相当。从表 2 可以看出, 本文算法在保持与 UMHEX 算法质量相当的情况下, 其运动估计搜索时间比 UNHexagonS 算法优越许多, 最多节省时间可以达到 20% 之多, 平均节省时间分别为 15.88% 和 15.94%。

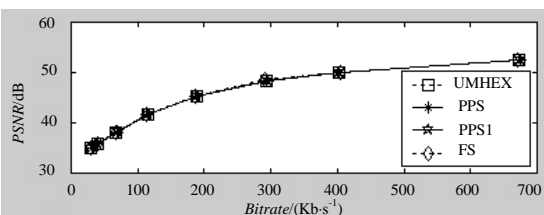
不同序列在 4 种算法中的 RDO 曲线比较如图 3 所示, 通过改变 QP 来改变 PSNR 和 Bitrate, 从而反映压缩后图像的质量好坏。为了节省空间, 仅列出了 4 种序列的 PSNR 和 Bitrate 的变化。



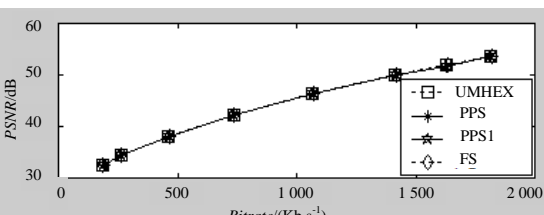
(a)Akiyo



(b)Coastguard



(c)Salesman



(d)Mobile

图 3 不同序列在 4 种算法中的 RDO 曲线比较

(下转第 231 页)