

## 一种采用多模式切换的快速运动估计算法

李贺军<sup>\*①②</sup> 李和平<sup>①</sup> 李建雄<sup>①</sup>

<sup>①</sup>(中国科学院电子学研究所 北京 100190)

<sup>②</sup>(中国科学院大学 北京 100039)

**摘要:** 搜索模式对于基于块的运动估计算法(BMA)的搜索速度和编码质量起着重要影响。该文提出一种多模式切换的快速运动估计算法,其采用了早停止和选择性搜索技术来提高编码速度,以小菱形作为起始搜索模式,然后过渡到六边形模式,最后使用正方形搜索模式进行细化,故命名为 DHS(Diamond-Hexagon-Square)算法。实验结果表明, DHS 算法对于各种运动情况(慢速、中等速度和快速)的视频序列具有普适性,在率失真性能未出现显著下降的前提下,其速度明显快于小菱形算法(DIA)、六边形算法(HEX)、穷尽搜索算法(ESA)和非对称十字形多层次六边形格点搜索(UMHexagonS)算法。

**关键词:** 视频编码; 运动估计; 多模式切换; 早停止; 选择性搜索

中图分类号: TN919.81

文献标识码: A

文章编号: 1009-5896(2013)03-0689-07

DOI: 10.3724/SP.J.1146.2012.00811

## A Multi-pattern Switching Algorithm for Fast Motion Estimation

Li He-jun<sup>①②</sup> Li He-ping<sup>①</sup> Li Jian-xiong<sup>①</sup>

<sup>①</sup>(Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China)

<sup>②</sup>(University of Chinese Academy of Sciences, Beijing 100039, China)

**Abstract:** Search patterns have an important influence on both searching speed and encoding quality of Block-based Motion estimation Algorithms (BMA). A fast block-based motion estimation algorithm utilizing a multi-pattern switching method is proposed, during which process a halfway stop technology and a selective search method are used to promote encoding speed. The algorithm is named Diamond-Hexagon-Square (DHS) algorithm, for a small diamond pattern is used as the initial search pattern, followed by a hexagon pattern, and a square search pattern is utilized at last for refinement. Experimental results indicate that the DHS algorithm is suitable for various video sequences with different motion contents (slow, medium and fast). It is faster than the small DIAMOND search algorithm (DIA), HEXagon search algorithm (HEX), a Exhausted Search Algorithm (ESA) and Unsymmetrical-cross Multi-Hexagon-grid Search (UMHexagonS) algorithm with no obvious rate-distortion performance deprivation.

**Key words:** Video encoding; Motion estimation; Multi-pattern switching; Halfway stop; Selective search

### 1 引言

运动估计是视频压缩编码的关键技术,影响着编码的速度和质量,其运算复杂度是整个视频压缩系统中最高的,因此对运动估计算法的优化是视频压缩编码领域的重要研究课题。目前,基于运动补偿、数值变换和量化及熵编码的编码框架已经成为现有视频编码标准的基本框架,国际电信联盟(ITU)制定的所有视频编码标准均建立在基于块的混合编码框架基础上,于是基于块的运动估计算法(BMA)在实际应用中被广泛采纳。本文提出的菱形-六边形-正方形(DHS)算法就是一种基于块的运动估计算

法,确切地说, DHS 算法属于基于中心搜索的运动估计算法。

基于中心搜索的运动估计算法<sup>[1]</sup>是一类重要的运动估计算法。典型的基于中心搜索的算法有 3 步法(TSS)、新 3 步法(NTSS)、4 步法(4SS)、菱形算法(DS)以及六边形算法(HEX)等,其中,菱形模式和六边形模式作为有效的搜索模式,被广泛应用于运动估计算法中。例如: Ranjit 等人<sup>[2]</sup>基于六边形算法和菱形算法,提出了一种六边形-菱形搜索算法(HEXDS)。Ndili 等人<sup>[3]</sup>在传统菱形算法的基础上,提出了一种面向硬件的改进菱形搜索算法(HMDS)。Cheng 等人<sup>[4]</sup>提出的菱形-六边形搜索算法首先采用六边形模式确定最佳匹配块的大致位置,然后使用小菱形模式进行细化搜索。

2012-06-25 收到, 2012-11-12 改回

国家高技术研究发展计划项目(2008AA121804)资助课题

\*通信作者: 李贺军 lihejun3@163.com

可见, 根据视频序列的运动情况, 选择合理的搜索模式, 对于提高运动估计算法的效率至关重要。本文提出的 DHS 算法就是在小菱形搜索算法<sup>[5]</sup>和六边形搜索算法<sup>[6]</sup>的基础上, 参考文献[7]中的选择性搜索思路, 加以改进得到的一种基于中心搜索的运动估计算法。DHS 算法使用小菱形、六边形和正方形等 3 种搜索模式, 并根据视频序列的运动情况灵活地进行搜索模式的切换, 并采用了早停止和选择性搜索技术, 在保证率失真性能未出现显著下降的前提下, 有效地提高了编码速度。

## 2 DHS 算法

### 2.1 DHS 算法的匹配代价函数

为衡量当前编码块和候选搜索块的匹配程度, DHS 算法采用如下匹配代价函数:

$$\text{COST}(\mathbf{MV}, \lambda_{\text{MOTION}}) = \text{SAD}(\mathbf{MV}) + \lambda_{\text{MOTION}} \cdot R(\mathbf{MV} - \mathbf{MVP}) \quad (1)$$

其中  $\text{COST}(\mathbf{MV}, \lambda_{\text{MOTION}})$  是当前编码块和候选搜索块的匹配代价, 在 DHS 算法中, 假设  $\text{COST}(\mathbf{MV}, \lambda_{\text{MOTION}})$  在搜索窗内是一个平滑且单调的曲面, 即匹配代价函数值随着候选运动矢量接近全局最优点而单调减小。下文中, 为了减小计算量以提高搜索速度, 对候选运动矢量进行了基于邻近已搜索点的匹配代价函数值的选择性搜索, 就是基于上述假设。 $\mathbf{MV} = [\text{MV}_x, \text{MV}_y]$  是候选运动矢量;  $\lambda_{\text{MOTION}}$  是拉格朗日乘子, 本算法中取值为 1;  $\mathbf{MVP} = [\text{MVP}_x, \text{MVP}_y]$ , 是根据当前编码块的邻近块的运动矢量信息获得的运动矢量预测值, 并以此确定运动搜索的起点。 $R(\mathbf{MV} - \mathbf{MVP})$  代表对运动矢量进行差分编码所消耗的比特数。 $\text{SAD}(\mathbf{MV})$  是编码块和候选搜索块内所有像素的绝对差值和:

$$\text{SAD}(\mathbf{MV}) = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} |f_C(x, y) - f_R(x - \text{MV}_x, y - \text{MV}_y)| \quad (2)$$

其中  $f_C(x, y)$  是当前编码帧中  $W \times H$  尺寸的分割块内第  $(x, y)$  个点的像素值,  $f_R(x + i, y + j)$  是参考帧中  $W \times H$  尺寸的候选搜索块内第  $(x + i, y + j)$  个点的像素值。

### 2.2 DHS 算法概述

DHS 算法采用图 1 中的 3 种搜索模式。

DHS 算法中, 首先利用运动矢量预测值  $\mathbf{MVP}$  确定搜索起点  $S$ , 然后进行 1 次小菱形搜索, 若  $S$  是最佳匹配点, 停止搜索。若  $S$  不是最佳匹配点, 则在第 1 个菱形最佳顶点  $V_{\text{DIA}}$  的方向上, 对以  $S$  为中心的六边形上的 2 个或 3 个顶点进行外扩搜索(详见 2.3 节), 并将上述顶点的匹配代价值与  $V_{\text{DIA}}$  的匹

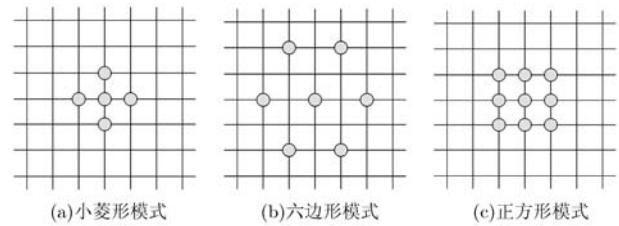


图 1 DHS 算法使用的 3 种搜索模式

配代价值相比较: 若  $V_{\text{DIA}}$  的匹配代价值仍为最小, 则在  $V_{\text{DIA}}$  附近继续进行选择性搜索, 此次搜索可视为第 2 次小菱形搜索, 搜索得到最佳匹配点后, 停止搜索; 否则, 转入六边形搜索模式, 得到最佳匹配六边形。最后, 根据最佳匹配六边形 6 个顶点的匹配代价值的大小情况, 进行正方形选择性搜索。DHS 算法的流程如图 2 所示。

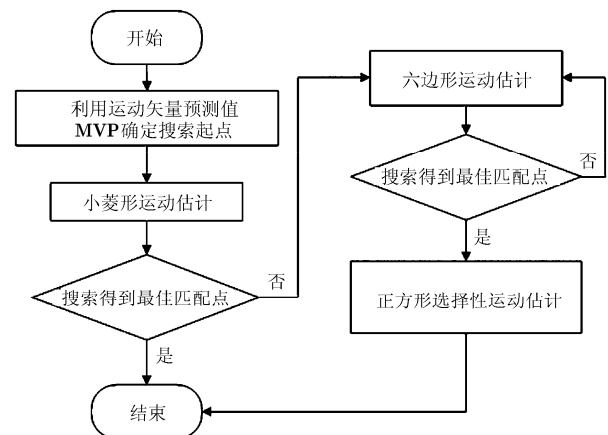


图 2 DHS 算法流程图

### 2.3 DHS 算法的详细步骤

**2.3.1 小菱形搜索模式** 首先, 以当前编码块的运动矢量预测值  $\mathbf{MVP}$  确定的搜索起点  $S$  为中心, 进行 1 次小菱形搜索, 若  $S$  是最佳匹配点, 停止搜索, 如图 3 所示; 否则进入 2.3.2(A) 或 2.3.2(B)。

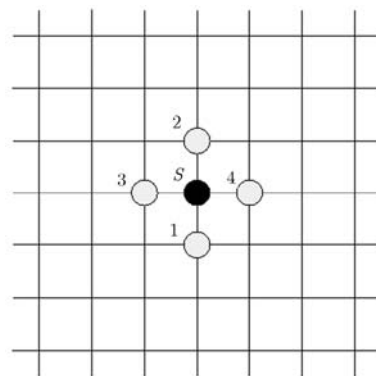


图 3 小菱形搜索模式

### 2.3.2 小菱形到六边形的过渡搜索模式及六边形搜索模式

(A)第 1 个菱形的最佳顶点是 1 或 2 若图 3 中 1 是起始搜索菱形的最佳顶点(2 为最佳顶点时, 处理方法类似), 即 1 是小菱形的 5 个搜索点中匹配代价最小的点, 则在 1 的下侧继续搜索 2 点, 这 2 个点是以  $S$  为中心的六边形上的 2 个顶点, 分别为 5, 6, 见图 4。本文作如下约定:  $cost(P)$  表示  $P$  点处的匹配代价函数值; 匹配代价函数值简称  $cost$  值。然后, 比较  $cost(1)$ ,  $cost(5)$  和  $cost(6)$  :

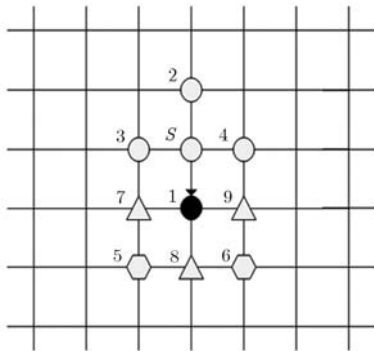


图 4 1 是最佳顶点时的搜索示意图

(1)若  $cost(1)$  最小, 根据匹配代价函数的单调性分布假设, 对 7, 8, 9 三点进行选择搜索。首先统计下面 3 组匹配代价的和值:

$$cost\_sum1=cost(3)+cost(1)+cost(5) \quad (3)$$

$$cost\_sum2=cost(5)+cost(1)+cost(6) \quad (4)$$

$$cost\_sum3=cost(6)+cost(1)+cost(4) \quad (5)$$

然后比较  $cost\_sum1$ ,  $cost\_sum2$  和  $cost\_sum3$ : 若  $cost\_sum1$  最小, 只搜索 7 点;  $cost\_sum2$  最小, 只搜索 8 点;  $cost\_sum3$  最小, 只搜索 9 点。最终, 在 1, 7, 8, 9 中确定出最佳匹配点, 停止搜索。

(2)若  $cost(5)$  或  $cost(6)$  最小, 则转入六边形搜索模式, 以  $cost(6)$  最小为例, 见图 5。

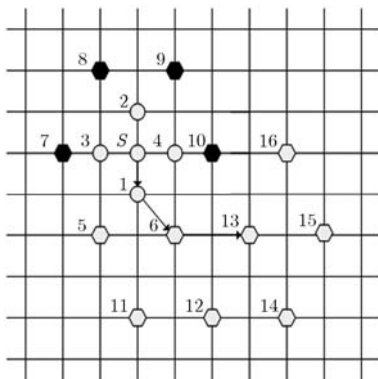


图 5  $cost(6)$  最小时的搜索示意图

首先, 将以搜索起点  $S$  为中心的六边形的 6 个顶点补齐, 添加 7, 8, 9, 10 共 4 个点, 并规定以  $S$  为中心的六边形作为搜索过程的第 1 个六边形。根据匹配代价函数的单调性假设, 具有最小  $cost$  值的点更接近 6, 新添加的 7, 8, 9 点及其附近点极不可能成为最佳匹配点, 故不必对这 3 个点进行搜索。后续的正方形搜索模式可能用到第 1 个六边形顶点的  $cost$  值, 可进行如下设置:

$$cost(7)=cost(8)=cost(9)=MAX=(1<<32)-1 \quad (6)$$

顶点 10 也极不可能成为最佳匹配点, 但其附近的点(如 10 下方的相邻点)可能为最佳匹配点, 为减少计算量, 可用  $cost(4)$  作为  $cost(10)$  的近似值。

然后, 以 6 作为第 2 个六边形的中心点, 沿  $S$  到 6 的延长线方向外扩 3 点, 即对 11, 12, 13 进行搜索。此后, 根据六边形中心点与顶点  $cost$  值的大小关系, 不断进行外扩 3 点的六边形搜索, 直到搜索到最佳匹配六边形; 最后转入正方形细化搜索模式。

图 5 中, 假设第 2 个六边形的中心点 6 不是最佳中心点, 13 是最佳顶点, 故六边形的中心点由 6 移至 13, 并在 6 到 13 的延长线方向上外扩 3 点, 继续进行六边形搜索。

(B)第 1 个菱形的最佳顶点是 3 或 4 若图 3 中 4 是起始搜索菱形的最佳顶点 (3 为最佳顶点时, 处理方法类似), 搜索过程见图 6。

首先, 搜索以  $S$  为中心的六边形上的 3 个顶点, 分别为 5, 6, 7, 并比较  $cost(4)$ ,  $cost(5)$ ,  $cost(6)$  和  $cost(7)$  :

(1)若  $cost(4)$  最小, 则根据  $cost(5)$ ,  $cost(6)$  和  $cost(7)$  的大小关系, 对 8, 9 进行选择搜索: 若  $cost(5)$  最小, 只搜索 8 点;  $cost(6)$  最小, 8, 9 两点均不必搜索, 直接确定 4 作为最佳匹配点;  $cost(7)$  最小, 只搜索 9 点。最终, 在 4, 8, 9 中确定出最佳

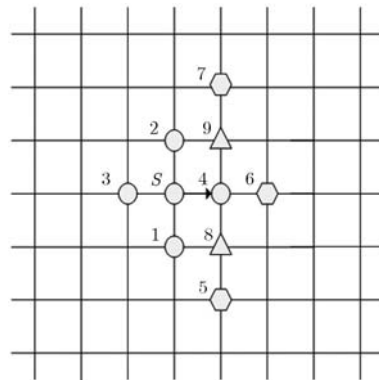


图 6 4 是最佳顶点时的搜索示意图

匹配点, 停止搜索。

(2)若  $\text{cost}(5)$ ,  $\text{cost}(6)$  或  $\text{cost}(7)$  最小, 则转入六边形模式, 以  $\text{cost}(6)$  最小为例, 见图 7。

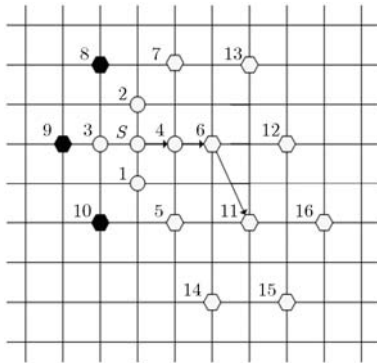


图 7  $\text{cost}(6)$  最小时的搜索过程

首先, 将以  $S$  为中心的六边形上的 6 个顶点补齐, 添加 8, 9, 10 三点, 并规定以  $S$  为中心的六边形作为搜索过程的第 1 个六边形。同样, 根据匹配代价函数的单调性, 具有最小  $\text{cost}$  值的点更接近 6, 新添加的 8, 9, 10 三点及其附近点极不可能成为最佳匹配点, 可进行如下设置:

$$\text{cost}(8) = \text{cost}(9) = \text{cost}(10) = \text{MAX} = (1 << 32) - 1 \quad (7)$$

然后, 以 6 作为第 2 个六边形的中心点, 沿  $S$  到 6 的延长线方向向外扩 3 点, 即对 11, 12, 13 进行搜索。此后, 根据六边形中心点与顶点  $\text{cost}$  值的大小关系, 不断进行外扩 3 点的六边形搜索, 直到搜索到最佳匹配六边形; 最后转入正方形细化搜索模式。

图 7 中, 假设第 2 个六边形的中心点 6 不是最佳中心点, 11 是第 2 个六边形中  $\text{cost}$  值最小的顶点, 故在 6 到 11 的延长线方向上外扩 3 点, 继续进行六边形搜索。

**2.3.3 正方形细化搜索模式** DHS 的正方形搜索模式的理论依据是匹配代价函数值随着候选运动矢量接近全局最优点而单调减小的假设。通过比较图 8 中最佳六边形的 6 个顶点的  $\text{cost}$  值, 在具有最小  $\text{cost}$  值的顶点附近进行选择性细化搜索: (1)  $\text{cost}(0)$  最小, 只搜索 6; (2)  $\text{cost}(1)$  最小, 只搜索 7, 8; (3)  $\text{cost}(2)$  最小, 只搜索 8, 9; (4)  $\text{cost}(3)$  最小, 只搜索 10; (5)  $\text{cost}(4)$  最小, 只搜索 11, 12; (6)  $\text{cost}(5)$  最小, 只搜索 12, 13。通过比较最佳六边形中心点与上述筛选出的候选搜索点的  $\text{cost}$  值, 可获得最佳匹配运动矢量, 搜索停止。

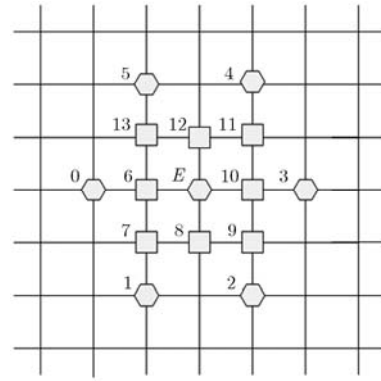


图 8 正方形选择性搜索模式

## 2.4 DHS 算法分析

对于固定大小的像素块, 计算每个运动矢量对应的匹配代价函数值所需的加法、减法等运算量是相同的(见式(1)和式(2))。因此, 可以用获得匹配运动矢量所需的搜索点数衡量运动估计算法的复杂度。

对于现实世界中的视频序列, 绝大多数编码块的运动矢量在包围搜索窗中心的  $5 \times 5$  像素范围内, 故 DHS 算法中首先进行小尺寸的小菱形运动估计是合理的, 有利于快速搜索到最佳运动矢量, 提前终止搜索, 提高编码速度。

接下来切换到六边形运动估计模式, 相比于小菱形运动估计, 六边形运动估计具有更大的搜索步长, 其中, 六边形水平方向上 2 个搜索点的搜索步长是 2, 其余 4 个点的搜索步长  $\sqrt{5}$ 。因此六边形运动估计能减小落入局部最优点的概率; 对于数值较大的运动矢量, 六边形模式的搜索点数明显少于小菱形模式的搜索点数。因此, 当小菱形模式未能搜索到最佳匹配块时, 切换到六边形运动估计模式, 使得本算法对编码快速视频序列具有适应性。

最后, 依据匹配代价函数的单调性, 即匹配代价函数值会随着候选运动矢量接近全局最优点而单调减小的特性, 在小菱形到六边形的过渡搜索模式以及正方形细化搜索模式中, 对候选搜索点进行了选择性搜索, 可有效减少搜索点数。尽管在选择性搜索时, 会增加加法运算和条件判断, 但注意到至少搜索 1 个点, 就能减少 1 个编码块的匹配代价函数值的计算量, 显然后者的运算量更大, 故能明显提高编码速度。

## 3 实验结果及数据分析

### 3.1 实验环境及参数设定

实验采用的硬件环境为一台 PC 机(CPU: Intel(R) Celeron(R) 2.80 GHz, 内存: 1 GB), 使用开源代码 x264<sup>[8]</sup>, 开发环境为 Microsoft Visual

Studio 2008。DHS 算法的性能与 x264 中的 ESA, UMH(摘要中 UMHexagonS 算法的简称), DIA 和 HEX 算法以及参考文献[2]提出的 HEXDS 算法进行对比分析。

帧间模式搜索块的尺寸为  $16 \times 16$  像素, 帧内编码块采用  $16 \times 16$  和  $4 \times 4$  两种尺寸。搜索窗口的尺寸是在水平和垂直方向均为  $\pm 16$  个像素, 使用当前编码帧的前 1 帧作为参考帧。编码帧序列的格式为  $IP \dots PIP \dots PI \dots$ , 其中,  $I$  表示 IDR 帧,  $P$  表示帧间编码帧, 每 30 帧出现 1 个 IDR 帧。熵编码采用 CABAC。编码输出的视频帧率为 25 fps。

### 3.2 实验结果及分析

为了验证 DHS 算法的性能, 选择不同运动情况(慢速, 中等速度和快速)的视频序列进行编码实验。以每个最佳匹配块的平均搜索点数作为搜索速度的评价指标。使用 MSE (均方差准则)对编码质量进行评价, 对像素值的亮度分量的 MSE 进行比较。MSE 准则的计算公式如下:

$$MSE(MV) = \frac{1}{WH} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} [f_C(x, y) - f_R(x - MV_x, y - MV_y)]^2 \quad (8)$$

其中  $MV$ ,  $f_C(x, y)$  和  $f_R(x + i, y + j)$  与式(1), 式(2)中的含义相同。采用率失真特性曲线(码率-PSNR 曲线)评价率失真性能。对于算法的整体编码性能, 用帧率、码率和 PSNR 作为评价指标。其中帧率指每秒编码的帧数, 表示使用特定运动估计算法(即 ESA, UMH, DIA, HEX, HEXDS 和 DHS 等)时编码器的整体编码速度。由于运动估计的运算复杂度是整个编码器系统中最高的, 运动估计过程的耗时占压缩编码总时间的绝大多数, 而且实验中使用的编码器, 除采用的运动估计算法不同外, 其余条件均相同, 故可用编码器的整体编码速度近似衡量运动估计算法的速度, 即可用帧率近似衡量运动估计算法的速度。码率指输出视频帧率为 25 fps 时的数据速率。由于 DHS 算法的实现基于小菱形算法和

六边形算法, 故将着重分析 DHS 算法相比 DIA 和 HEX 算法在性能上的提高。

(1)慢速序列编码实验 采用 Mother-daughter 序列(cif 尺寸)作为慢速代表性序列。第 2-30 帧逐帧的  $16 \times 16$  像素块的平均搜索点数如图 9(a); 每帧图像亮度分量的 MSE 如图 9(b)。编码整个序列的率失真特性曲线如图 9(c), 量化参数 QP 选择了 23, 26, 29, 32, 35 和 38。编码整个序列的帧率、码率和 PSNR 情况见表 1。图 9(a)和图 9(b)及表 1 的实验中 QP = 35。

如图 9(a)所示, 每个像素块的平均搜索点数, DHS 算法是最少的。图 9(b)表明 6 种算法中, DHS, HEXDS, DIA 和 HEX 算法的编码质量稍差, 且这 4 种算法的 MSE 性能大致相同。

从表 1 可知, DHS 的编码速度比 HEXDS 快 11.37%; 比 DIA 快 3.91%, 比 HEX 快 15.18%, 这是因为 Mother-daughter 序列编码块的运动矢量主要分布在数值较小的区域, 而 DHS 和 DIA 算法的起始步骤搜索点数明显少于 HEX 算法。DHS 以小菱形作为起始搜索模式, 对于慢速序列, DHS 算法的执行主要停留在小菱形搜索模式阶段, 故 DHS 和 DIA 算法的性能相差不大。图 9(c)表明 DHS 的率失真性能稍差于 DIA 和 HEX, 这是由于 DHS 算法采用的早停止和选择性搜索方法降低了搜索精度, 导致帧间预测残差增大, 使得 DHS 的 PSNR 略低于 DIA 和 HEX 算法, 同时引起码率略微增大。

(2)中等速度序列编码实验 采用 Crew 序列(cif 尺寸)作为中等速度代表性序列。实验安排与上文 Mother-daughter 序列的编码实验相同; 结果见图 10 及表 2。

由于 Crew 序列存在较快的大范围运动成分, DHS 算法有更高的几率运行到六边形模式和正方形选择性细化阶段, 相比 DIA 和 HEX, 其对搜索点数的减少量更加明显, 如图 10(a)所示; 图 10(b)和图 10(c)表明, DHS, DIA 和 HEX 三者的编码质

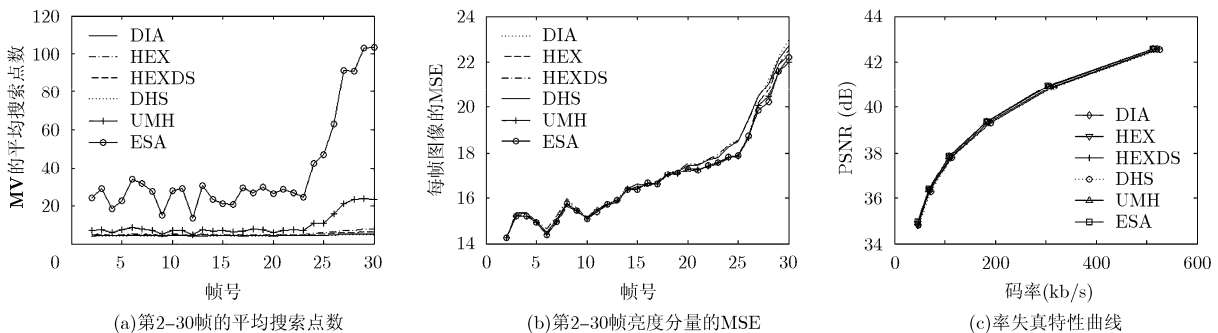


图 9 Mother-daughter 序列编码实验结果

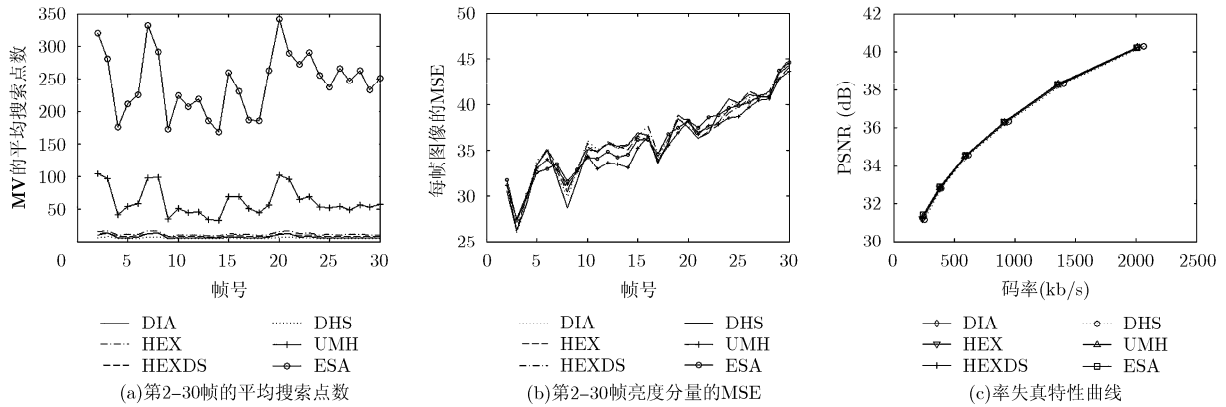


图 10 Crew 序列编码实验结果

表 1 Mother-daughter 序列的编码实验数据

指标	ESA	UMH	DIA	HEX	HEXDS	DHS
帧率(fps)	2.31	6.36	9.71	8.76	9.06	10.09
码率(kb/s)	68.91	68.72	69.89	69.16	70.31	70.38
PSNR(dB)	36.391	36.377	36.305	36.359	36.306	36.296

表 2 Crew 序列的编码实验数据

指标	ESA	UMH	DIA	HEX	HEXDS	DHS
帧率(fps)	1.27	3.71	6.11	5.61	6.14	6.56
码率(kb/s)	384.96	381.98	383.83	384.31	387.16	393.48
PSNR(dB)	32.882	32.849	32.811	32.817	32.810	32.803

量和率失真特性相差不大。从表 2 可知, DHS 算法的编码速度比 DIA 快 7.36%, 比 HEX 快 16.93%, 比 HEXDS 快 6.84%。在码率方面, DHS 仅比 DIA 高 2.51%, 比 HEX 高出 2.38%, 比 HEXDS 高 1.63%。DHS 的 PSNR 比 DIA 和 HEX 算法稍低。这表明, DHS 算法以率失真性能的轻微下降为代价, 获得了编码速度的较大提升。

(3)快速序列编码实验 采用 Soccer 序列(704 × 576 尺寸)作为快速代表性序列, 结果见图 11 及表 3。

如图 11(a), DHS 算法的 MV 平均搜索点数最

少; DHS 算法的 MSE 处于 6 种算法的中间水平, 如图 11(b); 图 11(c)表明 DHS 算法与其它算法保持了一致的率失真特性。从表 3 可知, DHS 的速度比 DIA 提高了 13.48%, 比 HEX 提高了 21.69%, 比 HEXDS 提高了 10.38%; DHS 算法的码率只比 DIA, HEX 和 HEXDS 分别高出 1.71%, 2.91%和 1.50%。

综上, 针对各种运动情况(慢速、中等速度和快速)的视频序列进行实验, 相比 ESA, UMH, DIA, HEX 和 HEXDS 算法, DHS 算法的编码速度有了不同程度的提高; 同时, DHS 的率失真性能轻微下

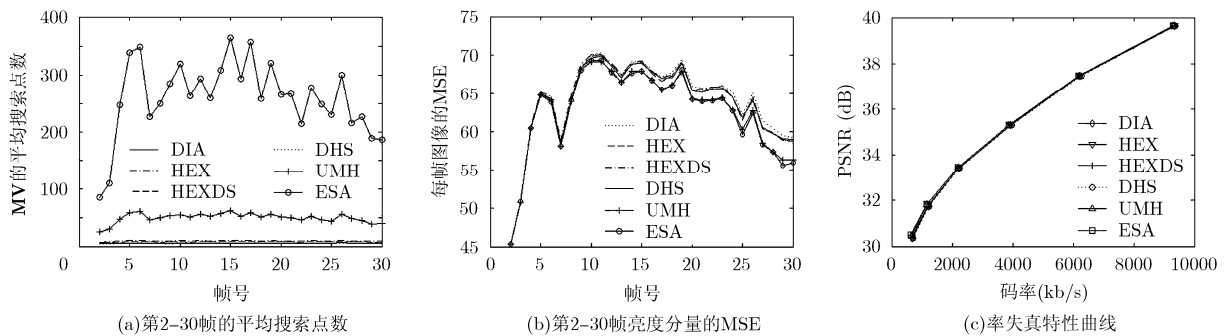


图 11 Soccer 序列编码实验结果

表3 Soccer 序列的编码实验数据

指标	ESA	UMH	DIA	HEX	HEXDS	DHS
帧率(fps)	0.21	0.98	1.78	1.66	1.83	2.02
码率(kb/s)	1171.87	1170.81	1212.79	1198.69	1215.33	1233.55
PSNR(dB)	31.817	31.810	31.718	31.785	31.739	31.738

降, 这是因为 DHS 采用的早停止和选择性搜索技术虽然明显地减少了搜索点数, 但降低了部分搜索精度, 从而导致图像质量的轻微下降和码率的略微上升。

#### 4 结论

本文基于小菱形搜索算法和六边形搜索算法, 提出了一种多种搜索模式灵活切换的基于块的运动估计算法, 即综合采用了小菱形模式、六边形模式和正方形选择性搜索模式的 DHS 算法, 并使用了早停止技术和选择性搜索技术来提高编码速度。实验结果表明, DHS 算法适用于各种运动情况(慢速、中等速度和快速)的视频序列, 以率失真性能略微下降为代价, 换取了视频编码速度的明显提高。

#### 参考文献

- [1] 邓中亮, 段大高, 崔岩松, 等. 基于 H.264 的视频编/解码与控制技术[M]. 北京: 北京邮电大学出版社, 2010: 36-70.  
Deng Z L, Duan D G, Cui Y S, *et al.* Video Coding/Decoding and Control Technology Based on H.264[M]. Beijing: Beijing University of Posts and Telecommunications Press, 2010: 36-70.
- [2] Ranjit S S S, Sim K S, Besar R, *et al.* Estimation of motion vector parameter using hexagon-diamond search algorithm[J]. *Journal of Real Time Image Processing*, 2011, 6(4): 225-234.
- [3] Ndili O and Ogunfunmi T. Hardware-oriented modified diamond search for motion estimation in H.264/AVC[C]. Proceedings of IEEE International Conference on Image Processing(ICIP 2010), Hong Kong, 2010: 749-752.
- [4] Cheng Y, Yang L, Fang Z W, *et al.* A fast motion estimation algorithm based on diamond and hexagon search patterns[C]. Joint Conferences on Pervasive Computing, Taipei, 2009: 595-598.
- [5] Zhu S and Ma K K. A new diamond search algorithm for fast block-matching motion estimation[J]. *IEEE Transactions on Image Processing*, 2000, 9(2): 287-290.
- [6] Zhu C, Lin X, and Chau L P. Hexagon-based search pattern for fast block motion estimation[J]. *IEEE Transactions on Circuits and System for Video Technology*, 2002, 12(5): 349-355.
- [7] Zhu C, Lin X, Chau L P, *et al.* An optimized diamond search algorithm for block motion estimation[C]. IEEE International Symposium on Circuits and Systems, Phoenix and Scottsdale, USA, 2002: 488-491.
- [8] VideoLAN ORGANIZATION. last\_x264.tar.bz2[OL]. <http://www.videolan.org/developers/x264.html>. 2012.2.15.

李贺军: 男, 1987年生, 硕士生, 研究方向为嵌入式系统开发和视频编解码技术。  
李和平: 男, 1976年生, 副研究员, 研究方向为合成孔径雷达信号处理技术。  
李建雄: 男, 1954年生, 研究员, 研究方向为合成孔径雷达系统设计。