

基于自适应搜索模板的快速运动估计算法

张凯舟 刘 鹏 姚庆栋

(浙江大学信息与电子工程学系, 杭州 310027)

摘 要 为了降低快速运动估计的计算复杂度,避免小菱形搜索算法带来的局部最优点问题,提出了自适应搜索模板的估计算法。该算法在搜索时根据 SAD 值的变化快慢和相邻帧之间时间相关性自适应选择搜索模板。实验表明,使用该算法编码,码流大小与使用菱形搜索算法和六边形搜索算法差距为 $\pm 0.6\%$,搜索点数为菱形算法的 72% ~ 77%,六边形算法的 83% ~ 86%。在减少搜索点数的前提下有效地避免了局部最优点问题。

关键词 运动估计 自适应 搜索模板

中图法分类号: TN919.8 文献标识码: A 文章编号: 1006-8961(2009)11-2191-07

A Fast Motion Estimation Algorithm Based on Adaptive Search Pattern

ZHANG Kai-zhou, LIU Peng, YAO Qing-dong

(Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027)

Abstract In order to reduce the computational complexity of fast motion estimation, and avoid the local minimum problem caused by small diamond search, a motion estimation algorithm based on adaptive search pattern was proposed. This algorithm chose search pattern adaptively depending on SAD changing speed and the temporal relativity in video sequence. Experimental results demonstrated that by using this algorithm, the encoded bit-rate changes within $\pm 0.6\%$ compared with diamond search and hexagon-based search; searching points were reduced to 72% ~ 77% and 83% ~ 86% compared with the diamond search and the hexagon-based search respectively. In the condition of reducing searching points this algorithm avoids local minimum problems.

Keywords motion estimation, adaptive, search pattern

1 引 言

对于视频序列图像,由于相邻帧之间存在很大的时间相关性,通过减少时间冗余,可以提高视频编码的效率。基于块匹配的运动估计算法是一种有效的方法,已被许多视频编码标准所采纳。而运动估计算法占整个编码器 50% ~ 70% 的运算量,很大程度上决定了编码器的速度,编码质量和编码大小。在块匹配运动估计算法中,全搜索算法精度最高,但

运算量大。为解决大运算量问题,产生了很多快速搜索算法,如 3 步法^[1],菱形搜索算法^[2],六边形搜索算法^[3]和改进的小菱形算法^[4]。这些快速算法大大提高了运动估计的速度,但同时也有搜索模板的非自适应问题。菱形搜索算法和六边形搜索算法为了避免局部最优点问题采用了大的搜索模板,但带来了搜索点数的大量增加;而小菱形搜索算法采用了小菱形搜索模板减少了搜索点数相比菱形和六边形搜索算法,但带来了局部最优点问题。

为了解决上述问题,提出了采用自适应搜索模板的运动估计算法。它自适应地选择从菱形算法模

收稿日期:2008-02-19;改回日期:2008-09-05

第一作者简介:张凯舟(1984 ~),男,2008 年于浙江大学获信息与通信工程专业硕士学位。主要从事视频编码和视频信号处理研究。

E-mail: zgz19842@yahoo.com.cn

通讯作者:刘 鹏, E-mail: liupeng@isee.zju.edu.cn

板中分割出来的 3 种模板,在尽可能减少搜索点数的情况下避免局部最优点问题。

2 常见算法的优缺点

在传统的使用搜索模板的算法中,一般采用了一大一小两种模板,用大模板初定位,用小模板精确定位。比如菱形搜索算法中使用了图 1 的两种搜索模板。六边形搜索算法使用了图 2 的两种搜索模板。

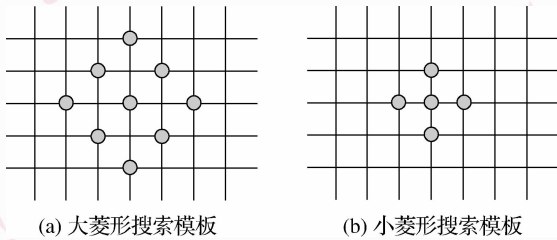


图 1 菱形搜索模板

Fig. 1 Diamond search pattern

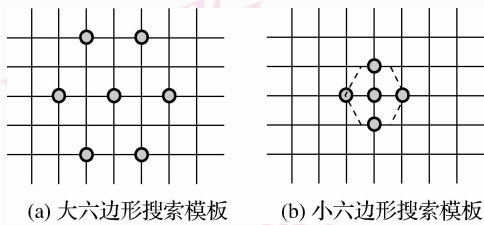


图 2 六边形搜索模板

Fig. 2 Hexagon-based search pattern

可见两种算法使用的小搜索模板是一样的。两种算法的搜索步骤也是一样的,先使用大模板,以第 3.2 节中提到的初始搜索点为中心搜索,找到块匹配绝对误差(SAD)最小点,如最小点不在模板中心就以最小点为中心再使用大模板搜索直到 SAD 最小点在模板中心,然后改为用小模板搜索,直到 SAD 最小点在小模板中心,这个点就为最佳匹配点。文献[3]表明六边形算法与菱形算法相比搜索点数更少。

这两种算法虽然在运动估计中有良好的表现,但也有自身的问题。(1)大部分情况下初始搜索点就为最佳匹配点。只用小模板搜索就可以了,如使用大模板再用小模板,有的搜索点就显得多余了。(2)当使用大模板搜索时,有时虽然 SAD 最小点不在中心,但不同搜索点之间的 SAD 值差已不大了,

说明最小点已经很近,这时没必要再用大模板搜索了,使用小模板更好。

针对上面提到的第 1 个问题,文献[4]和文献[5]中采用了只使用小搜索模板的方法即小菱形搜索算法。实验表明只使用小搜索模板可以将搜索点数减少到菱形算法搜索点数的 50% ~ 60%,但因为小模板的搜索半径很小,而且不能使搜索路径斜向移动,这会带来局部最优点问题,即终止搜索点并不是全局上的最佳匹配点,只是一个局部的最佳点。这种情况在有较大运动量的图像序列中特别明显。所以为了避免局部最优点问题,具有大搜索半径的搜索模板还是有必要使用的。

为了解决上面提到的 2 个问题,同时尽可能避免出现局部最优点,提出一种自适应地选择搜索模板的算法。

3 自适应搜索模板算法分析

3.1 自适应搜索模板算法中用到的模板

模板从菱形算法模板中拆分出来如图 3 所示,有 3 种。

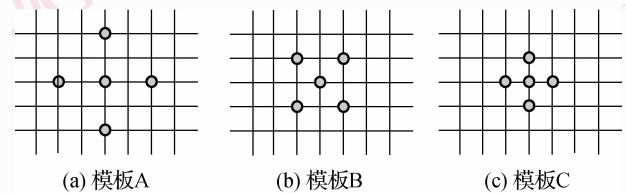


图 3 本算法中用到的模板

Fig. 3 Search patterns used in this algorithm

3 个模板的搜索半径变化平稳,分别为 $2, \sqrt{2}, 1$ 。采用模板 A 是为了有较大的搜索步长,避免在大运动量图像序列中搜索时陷入局部最优点问题;采用模板 B 是为了在小范围搜索时能使搜索路径斜向移动,同时避免从模板 A 直接向模板 C 过渡。如果模板 C 中点不为模板 A 中点时可能带来的搜索不精确问题,采用模板 C 搜索进行最后的精确定位。

这个算法通过 SAD 值的变化大小决定使用哪个模板,认为 SAD 值变化大时图像的运动也相应地大。所以当 SAD 值变化大时使用大的模板,反之用小的模板。

文献[4]提出了在搜索完一个模板的点后,当最小 SAD 值 SAD_{\min} 点,第二小 SAD 值 $SAD_{\text{sec}_{\min}}$ 点和它们连线的中点都不为搜索模板的中点时搜索它们

连线的中点的方法。原因在于 SAD_{min} 点与 SAD_{sec_min} 点连线中点的 SAD 值有很大的可能性小于 SAD_{min} , 并把这种算法用到了只使用了模板 C 的小菱形算法中。考虑到模板 B 虽然可以使搜索路径斜向移动, 但没法水平和垂直移动, 如果运用这种方法就可以增加一个水平和垂直方向的搜索点, 这样可以使得搜索路径定向更加有方向性, 更快地逼近最佳匹配点, 同时减少错过搜索最佳匹配点的可能性。不在模板 A 中使用这个算法的理由在于模板 A 是用于大步长的初步定向而不是用于最后的定位。如果搜索连线中点, 并在下一步基于该点使用模板 B 或 C 搜索, 可能陷入局部最优点问题。不在模板 C 中使用该方法的原因在于模板 C 在这个算法中是用于最后精确的定位, 添加这个搜索点反而多余了。所以模板 B 在实际搜索时, 添加了预测搜索点后出现了除原模板外的 4 种变形, 如图 4 所示。

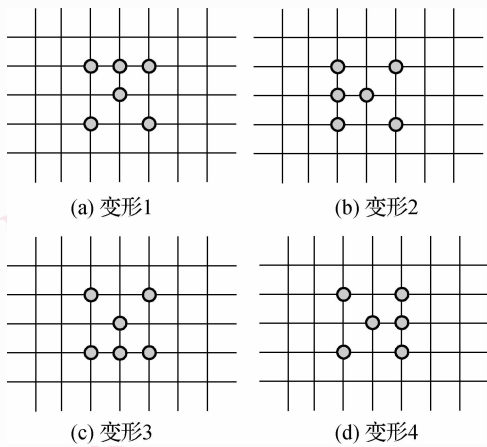


图 4 模板 B 的 4 种变形

Fig. 4 4 kinds of modified pattern of pattern B

3.2 初始搜索点的预测

为了减少搜索点数和克服局部最小 SAD 值问题, 一般不会直接以块的中心点开始搜索, 而是先计算中心点 (0,0) 预测的 SAD 值, 然后和一个阈值 T 比较。如果小于 T 就认为当前块的运动矢量为 (0, 0) 停止搜索, 不然则计算几个预测运动矢量对应点的 SAD 值, 选其中的最小值。如小于 T 则认为当前块的运动矢量为这个最小 SAD 值的对应矢量, 停止搜索。否则对比 (0,0) 点 SAD 和这个最小 SAD, 选小的一个, 其对应的点为初始搜索点。预测运动矢量一般为当前块的左, 左上和正上方的块的运动矢量。文献[6]中的研究表明对于 16×16 的静止块的 SAD 值一般在 600 ~ 1 300 之间, 所以阈值 T 一般

的对 16×16 块取 512, 其他大小块按比例缩小, 比如 16×8 块的 T 为 256, 实验表明这样取的 T 值和直接取 T 值为 0 编码效果基本一样。

3.3 判断 SAD 值的变化大小以确定使用的搜索模板

在确定起始搜索点后, 就要选择一个搜索模板进行搜索, 这时出现了一个问题。只知道搜索起始点的 SAD 值的情况下, 无法判断 SAD 值变化的大小, 从而无法判断运动矢量的大小, 进而无法选择搜索模板。在这种情况下需要一个作为可以和起始搜索点 SAD 值比较的一个参考值, 从而确定运动矢量的大小。文献[7]提到了利用连续帧的相关性来建立自适应模型的思想, 参考这种思想, 进一步考虑到连续帧中同一位置的块具有时间相关性, 采用如图 5 的自适应模型计算参考值 SAD_{pre-n} 。

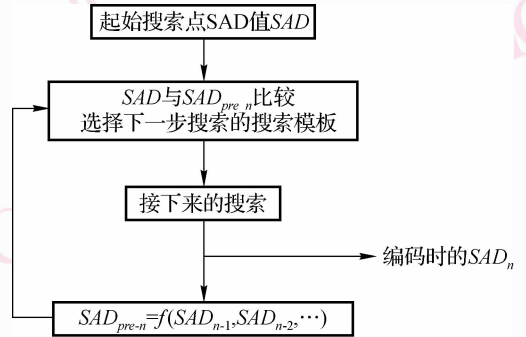


图 5 SAD_{pre-n} 的自适应计算模型

Fig. 5 The adaptive calculate mode of SAD_{pre-n}

在这个模型中第 n 帧 (编号按编码顺序递增) 中的某块用到的参考值 SAD_{pre-n} 为当前块 SAD 值的预测值, 是前面的已编码帧的对应块编码时的 SAD 值的函数。注意这里的连续帧要除去 I 帧, 因为 I 帧没有运动估计。

这里提出一种利用当前帧的前 4 帧对应块的 SAD 值计算 SAD_{pre-n} 的方法,

$$weight_table[4] = \{0.5, 0.2, 0.2, 0.1\}$$

$$SAD_{pre-n} = \sum_{i=1}^4 SAD_{n-i} \times weight_table[i-1] \quad (1)$$

式中, $weight_table$ 为权重系数数组, SAD_n 为第 n 帧编码时的 SAD 值 (都不考虑 I 帧), 当前帧为第 n 帧, 编号按编码顺序递增。使用这个计算方法主要是考虑到离当前编码帧编号比较远的帧与当前编码帧的相关性小, 参考意义不大, 可不用于计算参考值 SAD_{pre-n} , 同时考虑到编号越接近当前编码帧的帧与

当前编码帧的相关性越大,参考价值越大,权重系数就越大,所以这里采用了对第 $n-1$ 帧使用 0.5 的权重系数,对 $n-2$ 和 $n-3$ 帧使用 0.2 的权重系数对 $n-3$ 帧采用 0.1 的权重系数,而更远的其他帧不参加参考值 SAD_{pre_n} 的计算的方法。对一个图像序列,开始编码时前 4 帧计算需要的已编码帧的 SAD_n 有一部分不存在,这时对于一个 16×16 的块可设 SAD_n 为 512。

由于 SAD_{pre_n} 是通过图像的时间相关性估计出来的,同当前块最优点的 SAD 值应该有很大的相关性。但是如果当前块相对前面的对应块变化过大时,则 SAD_{pre_n} 与当前块最优点的 SAD 值的相关性就会变小,参考价值也会变小。基于如上的考虑,同时注意到第 1 次搜索时的最小 SAD 值一般比当前块最优点的 SAD 值大,通过对多个不同图像序列的实验,认为起始搜索点的 SAD 值在 SAD_{pre_n} 的 1.4 到 2 倍之间时 SAD_{pre_n} 的参考价值较大,在这种情况下 SAD_{pre_n} 与起始搜索点的 SAD 值相差越大则与最优距离越远,当使用大的搜索模板,反之亦然,试验表明当起始搜索点的 SAD 值在 SAD_{pre_n} 的 1.4 到 1.8 倍之间时使用模板 B 搜索,1.8 到 2 倍之间时用模板 A 搜索,可以在码流大小在 0.1% ~ 0.5% 变化情况下减少搜索点数。如果起始搜索点的 SAD 值不在 SAD_{pre_n} 的 1.4 到 2 倍之间, SAD_{pre_n} 的参考价值小,为了避免局部最优问题应使用模板 A 搜索。综上所述,在第 1 次搜索时,当起始搜索点的 SAD 值在 SAD_{pre_n} 的 1.4 到 1.8 倍之间时使用模板 B 搜索,否则用模板 A 搜索。

在接下来的搜索中,也可以采用预测的方式选择模板搜索。记录上次搜索的最小值点的值 SAD_{past} ,即本次搜索的中心点的 SAD 值,设本次搜索的 SAD 最小值 SAD_{min} 点为下次搜索的中心点。

如果 SAD_{min} 点为中心点的话考虑 3 种情况。(1) 本次用的搜索模板为模板 C,则搜索结束。(2) 本次搜索用的搜索模板为模板 B,则下次搜索用模板 C。(3) 本次搜索用的搜索模板为模板 A,则下次搜索用模板 C。

如果本次搜索的 SAD 最小值点不为中心点的话也考虑 3 种情况。(1) 本次用的搜索模板为模板 C,下次用模板 C 搜索。(2) 本次用的搜索模板为模板 B,如果 SAD_{min} 小于 1.3 倍的 SAD_{past} ,则下次用模板 C 搜索,否则下次用模板 B 搜索。(3) 本次用的搜索模板为模板 A,如果 SAD_{min} 在 1.4 倍到 1.8 倍

的 SAD_{past} 之间,则下次用模板 B 搜索,否则以下次用模板 A 搜索。

采用 SAD_{past} 与 SAD_{min} 比较,是因为一般情况下两值相差越小则离最优点越近,使用的搜索模板就可以越小。在用模板 A 搜索完后,如果 SAD_{min} 点不为中心点,当 SAD_{past} 与 SAD_{min} 相差比较小时不一定说明全局最优点就在附近,很可能是因为搜索点刚好接近一个局部最优点造成的,这种情况下需要再进行一次以 SAD_{min} 点为中心的模板 A 搜索,判断是否到达全局最优点附近,而在用模板 B 搜索时可以不考虑这种问题,不单是因为模板 B 和 C 搜索范围相差不大,还因为模板 B 和 C 都是用于最后的小范围搜索和精确定位局部最优点出现可能性很小,而门限值用 1.3 倍,1.4 倍和 2 倍的 SAD_{past} 也是通过多次试验分析得到的。

从上面的算法可以看到,当使用模板 A 搜索后,只有当满足一定条件的情况下才使用模板 B 搜索,这样做的目的在于当使用传统的大菱形模板进行大范围搜索时,大菱形模板中模板 B 的部分在很多情况下有些多余,但在小范围搜索时模板 B 比模板 C 搜索范围更广,能更快接近全局最优点,很有必要使用。

4 算法及其描述

通过分析可以得到算法的具体步骤如下:

(1) 按上面的方法进行初始搜索点预测,如果这时判断搜索终结跳到第 6 步,否则设定相应的初始搜索点进入第 2 步。

(2) 计算参考值 SAD_{pre_n} ,与初始搜索点 SAD 值对比如果初始搜索点的 SAD 值在 SAD_{pre_n} 的 1.4 倍到 1.8 倍之间跳入第 4 步,否则跳入第 3 步。

(3) 使用模板 A 搜索,记中心点 SAD 值为 SAD_{past} ,设 SAD 最小值 SAD_{min} 点为下次搜索中心。如果 SAD_{min} 点为中心点跳到第 5 步;如果 SAD_{min} 点不为中心点,当 SAD_{min} 在 1.4 倍到 1.8 倍的 SAD_{past} 之间跳到第 4 步,否则跳到第 3 步。

(4) 使用模板 B 搜索,记中心点 SAD 值为 SAD_{past} ,当最小 SAD 值 SAD_{min} 点,第二小 SAD 值 SAD_{sec_min} 点和它们连线的中点都不为搜索面板的中点时,搜索它们连线的中点,设 SAD 最小值 SAD_{min} 点为下次搜索中心。如果 SAD_{min} 点为中心点跳到第 5 步;如果 SAD_{min} 点不为中心点,当 SAD_{min} 小于 1.3

倍的 SAD_{pass} 跳到第 5 步, 否则跳到第 4 步;

(5) 用模板 C 搜索, 设 SAD 最小值 SAD_{min} 点为下次搜索中心, 如果 SAD_{min} 对应点为模板 C 中心点, 则跳到第 6 步, 否则跳到第 5 步。

(6) 搜索结束。

5 实验结果和分析

为了测试本文算法的性能, 在 H. 264 的测试模型 T. 264 - 0. 14^[8] 实现了菱形, 小菱形, 六边形搜索算法和本文算法。测试序列采用了 CIF 格式的 foreman, football, news, tabletennis, silence, 帧率为

30, QP 为 30, I 帧间隔为 15, 其余为 P 帧, 搜索半径为 16, 运动搜索时使用的块的大小为 $16 \times 16, 16 \times 8, 8 \times 16$, 参考帧数为 1 帧, 熵编码使用 CAVLC 编码。参考值 SAD_{pre_n} 的计算以宏块 (macro_block) 为单位, 如果用 16×8 或 8×16 的块搜索时, 使用参考值 SAD_{pre_n} 为所属宏块的 SAD_{pre_n} 的 $1/2$

本文算法和六边形, 菱形, 小菱形算法对比, 六边形, 菱形, 小菱形算法都先经过了上述的初始搜索点预测过程。对比编码后的 PSNR, 总的编码比特数, 总的搜索点数, 结果如表 1 所示 (注意这里的搜索点数包含了初始搜索点预测过程中计算过 SAD 值的点数)。

表 1 4 种搜索算法的性能对比

Tab. 1 Performance comparison between 4 algorithms

图像序列	编码帧数	算法	PSNR	码流长度 (byte)/百分比 (%)	搜索点数/百分比 (%)
foreman	250	菱形搜索	35. 15	481 061/100. 00	7 184 209/100. 00
		六边形搜索	35. 13	486 388/101. 11	6 396 984/89. 04
		小菱形搜索	35. 13	488 650/101. 57	3 914 404/54. 49
		本文算法	35. 15	484 084/100. 63	5 444 676/75. 79
football	80	菱形搜索	35. 38	340 177/100. 00	2 663 339/100. 00
		六边形搜索	35. 38	338 662/99. 55	2 363 219/88. 73
		小菱形搜索	35. 33	359 866/105. 78	1 552 944/58. 31
		本文算法	35. 38	337 661/99. 26	2 034 655/76. 39
tabletennis	250	菱形搜索	34. 22	562 925/100. 00	6 577 174/100. 00
		六边形搜索	34. 22	562 151/99. 86	5 785 008/87. 96
		小菱形搜索	34. 24	568 353/100. 96	3 234 868/49. 18
		本文算法	34. 22	563 144/100. 04	4 867 329/74. 00
news	250	菱形搜索	36. 50	292 919/100. 00	4 633 612/100. 00
		六边形搜索	36. 50	292 579/99. 88	4 029 322/86. 96
		小菱形搜索	36. 49	292 911/100. 00	2 151 033/46. 42
		本文算法	36. 50	292 916/100. 00	3 378 359/72. 90
silence	250	菱形搜索	34. 44	357 266/100. 00	6 171 194/100. 00
		六边形搜索	34. 43	357 267/100. 00	5 360 133/86. 86
		小菱形搜索	34. 43	358 772/100. 42	2 882 952/46. 72
		本文算法	34. 44	357 766/100. 14	4 477 508/72. 55

表 1 中, 码流长度和搜索点数都以菱形搜索算法的结果为参考, 给出其他算法相对菱形搜索算法结果的相对百分比。

可发现在 news 和 silence 测试序列中, 运动量

都比较小, 4 种算法编码后的码流长度, PSNR 都相差不多, 这种情况下小菱形搜索就相当适合, 其搜索点数只有菱形搜索的 46% ~ 47%, 本文算法搜索点数只有菱形算法的 72% ~ 73%, 也比六边形算法少。

表 2 列出了在 news 和 silence 测试序列中菱形搜索的大菱形模板和小菱形模板的使用数。

表 3 列出了在 news 和 silence 测试序列中用本文算法时模板 A, B, C 的使用数。

表 2 在 news 和 silence 测试序列
中大菱形和小菱形模板的使用数

Tab. 2 Usage of large and small diamond search patterns in news and silence test sequences

图像序列	编码帧数	大菱形模板数	小菱形模板数
news	250	337 154	336 629
silence	250	473 802	455 408

表 3 模板 A, B, C 在 news 和 silence
测试序列中的使用数

Tab. 3 Usage of A, B, C search patterns in news and silence test sequences

图像序列	编码帧数	模板 A 数	模板 B 数	模板 C 数
news	250	301 629	25 732	344 474
silence	250	421 058	27 740	474 175

可见大菱形模板使用数量很多,导致了搜索点数的上升,而这种低运动量图像,一般大量的运动估计点都在开始搜索时的中心点附近,根据情况减少大模板的使用数量以小模板代替反而能更快搜索到最优点。

在 foreman, tabletennis, football 序列中,小菱形搜索算法的局部最优点问题就暴露了,foreman 图像序列开始的运动量中等,中间有一段镜头移动很大导致运动量很大,tabletennis 中背景运动不多但局部运动量很大,而 football 中全程的运动量都很大。在这种情况下小菱形搜索算法的局部最优点问题使得相对于菱形搜索算法编码码流变大,在 foreman 中码流变大了 1.57%,在 tabletennis 中变大了 0.96%,而在 football 中变大了 5.78%,而使用本文算法可以在搜索点数只有菱形算法 74% ~ 77% 的情况下,相对于菱形搜索算法在 foreman 中码流变大 0.63%,在 tabletennis 中变大了 0.04%,而在 football 中反而减小了 0.74%。

在 football 序列中六边形搜索算法和本算法编码码流反比菱形搜索算法少的主要原因在于,菱形搜索算法的大菱形模板为本算法模板 A 和 B 的复合,在某些大运动量的搜索中很可能在某个局部最优点附近大菱形模板中模板 B 的部分出现最小

SAD 值点,误导搜索方向向局部最优点方向前进。而本文算法和六边形搜索算法大搜索模板中每次的跨度都为 2,能更好地避免搜索方向向局部最优点方向前进问题。

表 4 中列出了对每个序列中每幅图像的每个块分别采用 4 种算法搜索比较,选搜索结束时 SAD 值最小的算法为该块的编码算法,列出每种算法计算的 SAD 值为该最小 SAD 值的可能性。

表 4 4 种算法能搜索到最小 SAD 值点的可能性

Table. 4 The possibility of reaching the minimum SAD point use 4 different algorithms 单位:%

	菱形搜索	六边形搜索	小菱形搜索	本文算法
foreman	98.14	92.50	90.97	93.20
football	96.32	91.88	84.35	91.15
tabletennis	99.04	97.80	96.85	97.43
news	99.55	99.04	98.55	98.97
silence	99.11	98.01	96.90	97.65

从表 4 中可以看出,菱形搜索算法容易搜索到全局最优点,图像运动越剧烈,小菱形搜索算法就越难搜索到全局最优点,而本文算法不论图像运动剧烈与否都能很好地搜索到全局最优点,搜索效果和六边形搜索算法相当,且通过表 1 看出六边形搜索算法的搜索点数为菱形搜索算法的 86% ~ 90%,而本文算法的搜索点数为菱形搜索算法的 72% ~ 77%,优于六边形搜索算法。

6 结 论

自适应搜索模板算法根据 SAD 值的变化快慢和相邻帧之间时间相关性选择大小不同的搜索模板,可以有效地减少运动估计中的搜索点数,同时避免只采用小搜索模板带来的局部最优点问题。该算法在整像素最优点的搜索上有较好的表现。

参考文献 (References)

- Kim J N, Choi T S. A fast three-step search algorithm with minimum checking points using unimodal error surface assumption [J]. IEEE Transactions on Consumer Electronics, 1998, 44(3): 638-648.
- Zhu S, 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.
- Zhu C, Lin X, Chau L P. Hexagon-based search pattern for fast block motion estimation [J]. IEEE Transactions on Circuits and

- System for Video Technology, 2002, **12**(50):349-355.
- 4 Jiang Qiao-wei, Liu Peng. LPPDS: line prediction based pseudo-diamond search algorithm [J]. Journal of Zhejiang University (Engineering Science), 2007, **41**(4):612-615. [江巧微, 刘鹏. LPPDS: 基于线性预测的准菱形搜索算法 [J]. 浙江大学学报(工学版), 2007, **41**(4):612-615.]
 - 5 Gallant M, Cote G, Kossentini F. An efficient computation-constrained block-based motion estimation algorithm for low bit rate video coding [J]. IEEE Transactions on Image Processing, 1999, **12**(8): 1816-1823.
 - 6 Nie Y, Ma K K. Adaptive rood pattern search for fast block-matching motion estimation [J]. IEEE Transactions on Image Processing, 2002, **11**(12):1442-1449.
 - 7 Sorwar G, Murshed M, Dooley L S. A fully adaptive distance-dependent thresholding search (FADTS) algorithm for performance-management motion estimation [J]. IEEE Transactions on Circuits and System for Video Technology, 2007, **17**(4):429-440.
 - 8 T264. Software of H. 264[CP/OL]. <http://sourceforge.net/projects/t264>. 2005-8-6.