

基于运动方向预测 UMHexagonS 算法的改进^①

王艳营

(黑龙江科技学院 电气与信息工程学院, 哈尔滨 150027)

摘要: 为了进一步减少 UMHexagonS 算法的运算量, 在详细分析 UMHexagonS 算法搜索过程的基础上, 对运动方向的预测进行了研究, 并将运动方向预测与 UMHexagonS 算法中的正方形搜索和非均匀多层次六边形搜索相结合, 提出了一种改进算法。利用三个不同的序列进行测试仿真, 测试仿真结果表明, 在搜索精度和码率相似的情况下, 改进后的 UMHexagonS 算法搜索点数明显减少。

关键词: UMHexagonS 算法; 运动方向预测; 正方形搜索; 非均匀多层次六边形搜索

Improved UMHexagonS Algorithm Based on Motion Direction Prediction

WANG Yan-Ying

(Electrical & Information Engineering College, Heilongjiang Institute of Science and Technology, Harbin 150027, China)

Abstract: In order to further reduce the computation in UMHexagonS algorithm, the paper first detailed studies UMHexagonS algorithm search processing, and studies motion direction prediction, and combines motion direction prediction with square search and unsymmetrical multi-hexagon search in UMHexagonS algorithm, an improved algorithm is proposed. The test and simulation by three different sequences are done. The test results show that the improved UMHexagonS algorithm can significantly decrease search points when the search accuracy and rate are similar.

Key words: UMHexagonS algorithm; motion direction prediction; square search; unsymmetrical multi-hexagon search

运动估计是视频编码中的一项核心技术, 它能够有效去除序列图像的帧间冗余, 从而提高视频编码效率, 因此运动估计一直是视频压缩领域的研究热点^[1]。常见的运动估计算法是块匹配法, 如 FS、TSS^[2]、DS^[3]等, FS 精度最高, 但计算复杂, 不利于实时应用, TSS、DS 算法速度快, 搜索点数少, 但易于陷入局部最优, 而 UMHexagonS 算法能较为有效的克服这一缺点。

UMHexagonS 算法作为新一代运动估计算法, 因其优越的性能已被 H.264 标准正式采纳为整像素运动估计算法^[4], 为了进一步减少运动估计时间, 对 UMHexagonS 算法进行了改进, 在保持原算法失真性能基础上, 通过运动方向的预测, 有效地减少搜索点数。

1 UMHexagonS 算法

UMHexagonS (非对称十字形多层次六边形个点搜索) 算法采用混合和分等级的运动搜索策略来搜索最佳匹配点。它的基本思想是采用多种模板进行宏块匹配, 同时利用时空相关性进行运动矢量的预测, 搜索时针对不同的类型采用了大范围粗搜索混合模板, 细搜索采用六边形模板, 精细搜索采用十字形模板^[5]。利用 UMHexagonS 搜索最佳匹配点时, 采用 SAD (绝对误差和) 标准, 其计算公式为:

$$SAD(i, j) = \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} |f_k(x, y) - f_{k-1}(x+i, y+j)|$$

其中: $f_k(x, y)$ 为当前帧, $f_{k-1}(x+i, y+j)$ 为参考帧。UMHexagonS 算法具体搜索步骤如下:

Step1: 确定起始搜索点。利用中值预测、上层预

① 基金项目: 黑龙江省教育厅科学技术研究项目(11551444)

收稿时间: 2011-04-14; 收到修改稿时间: 2011-05-06

测、相应块预测和邻近参考帧预测这 4 种预测方法，选取 SAD 最小的一个作为起始点。若 SAD 很小，转至 Step5；若 SAD 很大，转至 Step4；否则转至 Step2；

Step2: 采用非对称十字形搜索，水平方向的搜索长度为窗口（即搜索范围，在参考帧的一定区域内设定的搜索范围，又称为搜索窗口）宽度，垂直方向的搜索长度为窗口宽度的一半，如图 1 的 Step2 所示，计算 SAD 值，若 SAD 值很小，转至 Step5；若 SAD 很大，转至 Step4；否则转至 Step3；

Step3: 以最小的 SAD 点为中心，在 5x5 区域内进行 25 点的正方形搜索，如图 1 的 Step3-1 所示，并进行 16 点的非均匀多重六边形搜索，利用伸缩系数对六边形进行扩展，如图 1 的 Step3-2 所示；

Step4: 扩展六边形搜索，以最小 SAD 为中心，进行大六边形搜索，如图 1 的 Step4 所示，再计算 SAD 点，若 SAD 点在中心，转至 Step5；

Step5: 以最小 SAD 点为中心，利用 SDSP 进行搜索，若最佳点在中心，停止搜索，确定最终运动矢量。

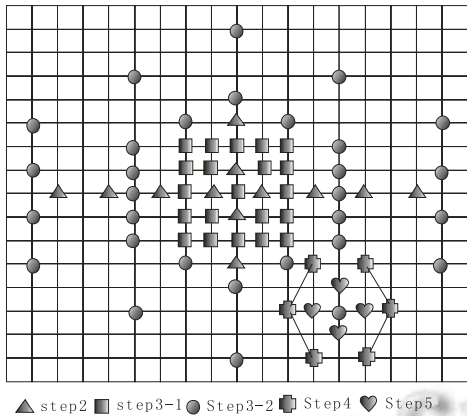


图 1 UMHexasonS 算法搜索过程

2 基于运动方向预测的UMHexasonS算法

从图 1 中可以看出，在 UMHexasonS 算法中，要进行 25 点正方形搜索和多重 16 点非均匀六边形搜索，step2 和 step3 最多搜索点数可达 201 个（图 1 中非对称十字搜索为 12 个点，以得到的最佳点为中心进行正方形搜索，占用非对称十字 4 个点，实际搜索 21 个点，再以当前的最佳点为中心进行非均匀六边形搜索，占用非对称十字 2 个点，实际搜索 14 个点，最多搜索点为 12+21+14x12=201），step4 最多搜索点数可达 192 个（图 1 中非均匀六边形有 32 个点，若在每个点利用

扩展六边形搜索就有最多点数 32x6=192），由此可以看出搜索点数还是很多的，利用运动估计中运动方向的预测可以减少搜索点数。

2.1 运动方向的预测

在运动估计中，运动矢量存在偏置性，也就是矢量偏移性，运动矢量总是高度集中在搜索窗的某个位置附近，称为矢量偏移^[6]。视频序列的运动矢量分布如图 2 所示。

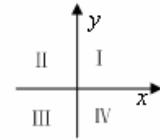


图 2 运动矢量分布

由图中可以看出，运动空间分成了 4 个象限，根据运动矢量的分布，可以事先对视频序列的方向进行预测，来判断视频序列趋于在哪一象限的方向运动（具体预测方法如后面所述），就可以直接搜索此象限内的点，其它象限内的点可以忽略搜索，以此来减少搜索的点数。

假设水平方向的匹配点为 SAD_x ，相对于预测点，匹配点的位置为 hor_x ；垂直方向的匹配点为 SAD_y ，相对于预测点，匹配点的位置为 ver_y ，运动方向的预测根据水平和垂直方向匹配点的位置分为以下情况：

(1) 若 $SAD_x > SAD_y$ ，说明最佳匹配点在水平方向，此情况下运动方向的预测情况为：

① 当 $hor_x > 0, ver_y \geq 0$ 时，运动矢量方向落在第 2 象限，如图 3(a)所示；

② 当 $hor_x > 0, ver_y \leq 0$ 时，运动矢量方向落在第 3 象限，如图 3(b)所示；

③ 当 $hor_x < 0, ver_y \geq 0$ 时，运动矢量方向落在第 1 象限，如图 3(c)所示；

④ 当 $hor_x < 0, ver_y \leq 0$ 时，运动矢量方向落在第 4 象限，如图 3(d)所示；

(2) 若 $SAD_x < SAD_y$ ，说明最佳匹配点在垂直方向，此情况下运动方向的预测情况为：

① 当 $hor_x \geq 0, ver_y > 0$ 时，运动矢量方向落在第 4 象限，如图 3(e)所示；

② 当 $hor_x \geq 0, ver_y < 0$ 时，运动矢量方向落在第 1 象限，如图 3(f)所示；

③ 当 $hor_x \leq 0, ver_y > 0$ 时，运动矢量方向落

在第 3 象限, 如图 3(g)所示;

④ 当 $hor_x \leq 0$, $ver_y < 0$ 时, 运动矢量方向落在第 2 象限, 如图 3(h)所示;

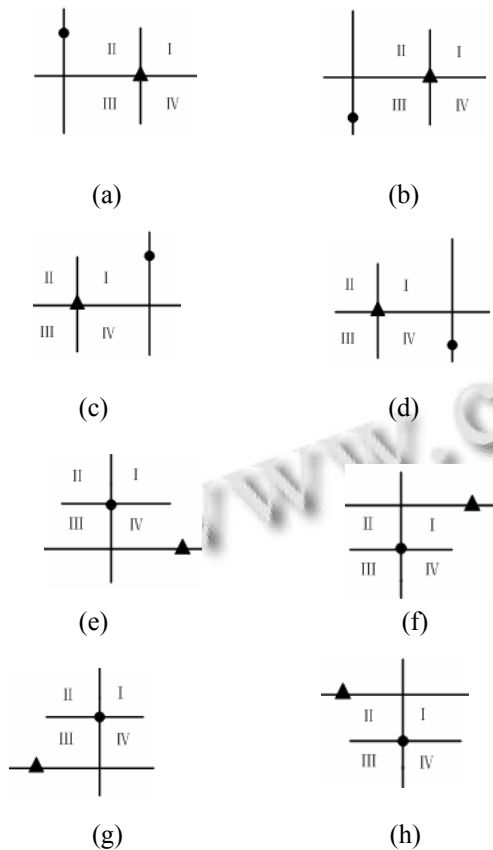


图 3 运动方向预测

其中: ▲为水平方向匹配点, ●为垂直方向匹配点

2.2 改进的 UMHexagonS 算法

利用运动方向的预测方法对 UMHexagonS 算法中的正方形搜索和非均匀多层次六边形搜索进行改进。

根据以上图 3(a)~(h)分析的情况, 改进后的正方形搜索方法如图 4(a)~(h)所示。

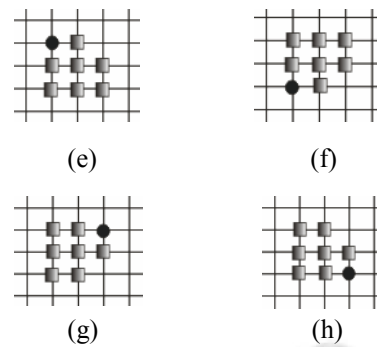


图 4 正方形在不同象限中的搜索方法

根据以上图 3(a)~(h)分析的情况, 改进后的非均匀多层次六边形搜索方法如图 5(a)~(h)所示。

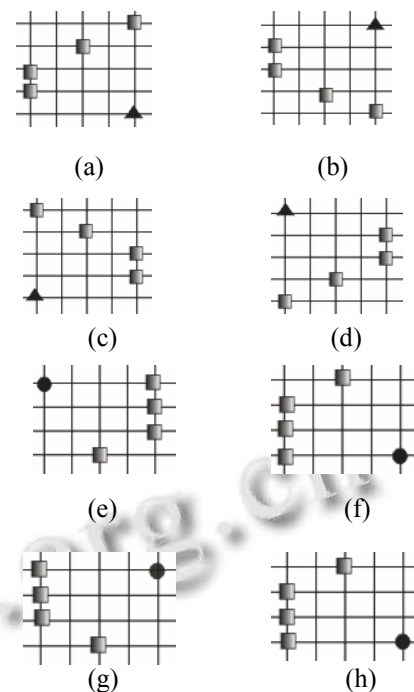
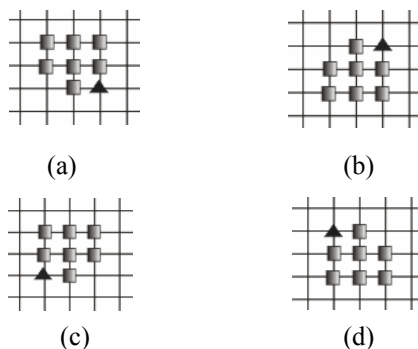


图 5 非均匀多层次六边形在不同象限中的搜索方法
其中, ▲为水平最佳匹配点, ●为垂直最佳匹配点, ■为需要搜索的点。

从图 4 可以看出, 原来需要搜索 25 点, 现在只需搜索 7 点, 从图 5 可以看出, 原来需要搜索 16 点, 现在只需搜索 4 点, 运动估计搜索点数明显减少。

3 测试结果仿真与分析

为了验证算法的结果, 采用 H.264 的参考模型 JM12.2 进行验证, 采用 CIF 视频格式的 silent、foreman 和 football 这 3 种不同运动强度的序列进行



测试, 编码帧数为 60, 参考帧数为 5, 帧率为 30, 编码格式为 IPPPP。各种算法在不.2 量化参数(QP)下的测试结果如表 1 和表 2 所示, 三种测试序列的性能比较如表 3 所示。

表 1 两种算法运动估计时间

| 测试序列 | QP | UMHexagonS 算法 | 改进算法 |
|----------|----|---------------|--------|
| | | 耗时/s | 耗时/s |
| silent | 28 | 20.17 | 15.53 |
| | 32 | 20.56 | 15.68 |
| | 36 | 21.25 | 15.98 |
| | 40 | 21.53 | 16.34 |
| foreman | 28 | 26.78 | 16.87 |
| | 32 | 27.16 | 16.96 |
| | 36 | 27.41 | 17.38 |
| | 40 | 27.68 | 17.54 |
| football | 28 | 234.67 | 132.58 |
| | 32 | 231.85 | 131.67 |
| | 36 | 225.34 | 130.08 |
| | 40 | 223.11 | 128.14 |

表 2 两种算法的 PSNR 值

| 测试序列 | QP | UMHexagonS 算法 | | 改进算法 | |
|----------|----|---------------|---------|-------|---------|
| | | PSNR | 码率 | PSNR | 码率 |
| silent | 28 | 37.13 | 80.24 | 37.14 | 80.13 |
| | 32 | 34.51 | 57.76 | 34.52 | 57.79 |
| | 36 | 31.18 | 36.59 | 31.15 | 36.62 |
| | 40 | 29.03 | 18.14 | 29.03 | 18.17 |
| foreman | 28 | 38.33 | 121.68 | 38.34 | 122.75 |
| | 32 | 35.47 | 98.34 | 35.48 | 98.57 |
| | 36 | 32.89 | 67.17 | 32.89 | 67.01 |
| | 40 | 29.34 | 28.38 | 29.33 | 28.42 |
| football | 28 | 36.83 | 1287.65 | 36.84 | 1288.73 |
| | 32 | 33.69 | 901.23 | 33.67 | 901.57 |
| | 36 | 31.23 | 607.24 | 31.23 | 607.31 |
| | 40 | 29.02 | 416.58 | 29.04 | 416.62 |

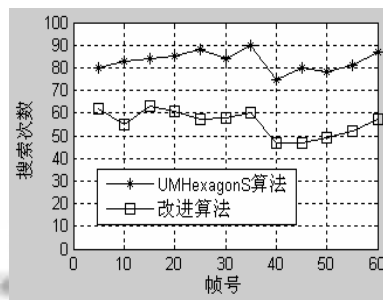
表 3 三种测试序列的性能比较

| 序列 | PSNR 变化/% | 码率变化/% | 节省时间/% |
|----------|-----------|--------|--------|
| silent | -0.04 | 0.16 | 23.91 |
| foreman | 0.03 | 1.01 | 36.95 |
| football | 0.04 | 0.12 | 42.89 |

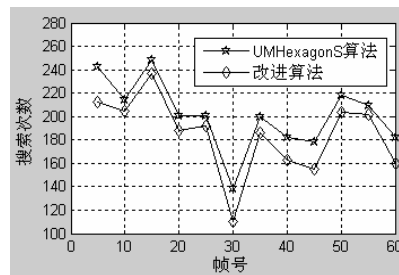
由表 3 可以看出, PSNR 的平均变化率为 0.01%, 码率平均变化率为 0.43%, 由此说明, 改进后的算法与 UMHexagonS 算法的率失真和码率性能保持不变。

测试序列运动估计时间平均节省了 34.58%, 三种不同测试序列节省时间的百分比不同, 运动剧烈的序列节省的时间较多。

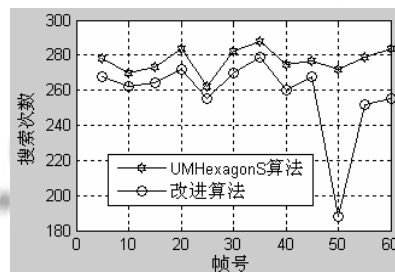
为了进一步说明测试结果, 对两种算法的搜索次数进行了仿真, 仿真图如图 6 所示。



(a) silent 序列搜索次数



(b) foreman 序列搜索次数



(c) football 序列搜索次数

图 6 三种序列的搜索次数仿真图

从图 6 中可以很清晰的看出, 改进后的算法的搜索次数与 UMHexagonS 算法相比, 搜索次数明显减少。

4 结论

该改进算法充分考虑了运动方向的预测性, 改进后的算法在保证原算法率失真和码率性能的基础上, 有效的减少了运动估计的时间, 特别适用于实时性要求高的场合。

(下转第 157 页)

硬件中保存了工程文件,可以从 PLC 编程软件选择上下载工程,在编程软件环境中打开和编辑工程文件。

7 PLC编程变量管理机制设计

在 PLC 编程软件中,所有使用到的变量由变量管理组件进行管理。变量根据使用方式、使用范围、使用意义等分为全局变量和局部变量两类。变量管理组件使用全局变量表类和局部变量表类分别管理全局变量和局部变量。当编译程序时,将所有用到的变量数据传送给编译器组件,作为分配地址的依据。

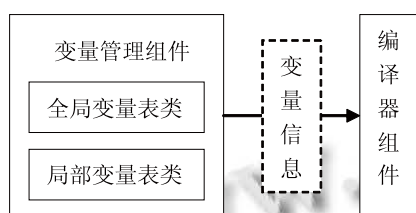


图 6 PLC 编程软件变量管理机制

8 结语

本文针对工业自动化领域应用越来越普遍的 PLC 技术中的重要组成部分——PLC 编程软件的开发提出了一种新的设计实现方式,该方法基于 Eclipse 平台

的插件机制,为实现跨平台的可移植性提供了一个良好的架构,可以降低软件各个组件之间的耦合性,而且扩展性和可维护性更强。该方法可以推广到一些大型基础软件、管理软件的开发过程当中,使软件可以跨平台,加速最终平台无关的趋势实现。

在实际开发过程中,由于 java 语言编译后的字节编码方式容易被反汇编,如果核心处理部分也采用 java 实现,商业安全性方面可能存在风险。采用二进制编码语言开发可以避免这种风险,然而核心功能如果用其他二进制编码语言编译生成动态链接库来实现,会对操作系统有一定程度的依赖。这种情况下需要制作多个平台的版本,但工作量仅限于核心库,而用户界面逻辑无需较大改动。相较而言,其他语言则不具备这种优势,跨平台意味着所有部分都要重写。后者对商业安全和工作量及可跨平台移植性方面取了一个较平衡的折中,综合考虑是一种较好的设计实现方式。

参考文献

- 1 Clayberg E, Rubel D. Eclipse Plug-Ins (third edition) 2009.
- 2 IEC 61131-3 INTERNATIONAL STANDARD. 2nd edition, 2003-01.

(上接第 180 页)

参考文献

- 1 王超,于新波.基于快速运动估计算法 UMHexagonS 的研究.中国新技术新产品,2010,(19):11-12.
- 2 Koga T, Iinuma K, Hirano A. Motion compensated interframe coding for video Conferencing. Proc. of National Telecommunication Conference. New Orleans, LA, 1981: 531-535.
- 3 Zhun S, Ma KK. A new Diamond Search Algorithm for Fast Block-matching Motion Estimation. IEEE Trans. on Circuits Systems for Video Technology, 1996,6(6):313-317.
- 4 Chen ZB, Xu JF, He Y, et al. Fast Integer-pel and Fractional-pel Motion Estimation for H.264/AVC. Journal of Visual Communication and Image Representation, 2000, 17(2):264-290.
- 5 刘云,孟岩,孙芳.JM 模型中 UMHexagonS 改进算法.系统仿真学报,2009,21(1):149-151.
- 6 向友军,吴宗泽,谢胜利.基于运动方向预测的快速运动估计算法.计算机工程,2009,35(24):20-22.