

## H. 264 中的运动估计和帧内预测算法研究

李绍滋<sup>1</sup>, 苏松志<sup>1</sup>, 成 运<sup>2</sup>, 孙 岩<sup>1</sup>, 郭 锋<sup>1</sup>

(1. 厦门大学智能科学与技术系, 福建厦门 361005; 2. 湖南人文科技学院通信与控制工程系, 湖南娄底 417000)

**摘 要:** 针对一些算法过早的确定搜索方向, 容易陷入局部最优值缺失搜索准确度的情况, 提出一种新的基于搜索方向预测的运动估计算法. 实验结果表明, 与单一搜索图形相比, 该算法具有更高的搜索精度和搜索速度. 帧内预测技术作为影响帧编码效率的关键, 一直被广泛研究, 为此, 本文提出了一种快速的帧内预测算法, 对宏块划分提出了新的“模板”宏块对比算法思想, 并在已有的抽样算法和搜索窗算法的基础上, 提出了改进方法. 实验结果证明, 算法在不降低图像质量的基础上, 编码速度平均提高 80%.

**关键词:** 运动估计; 帧内预测; H. 264

**中图分类号:** TP393 **文献标识码:** A **文章编号:** 0372-2112 (2008) 12A-175-06

### A New Motion Estimation and Intra-frame Prediction Method for H. 264 Video Coding

LI Shao-zi<sup>1</sup>, SU Song-zhi<sup>1</sup>, CHENG Yun<sup>2</sup>, SUN Yan<sup>1</sup>, GUO Feng<sup>1</sup>

(1. Department of Cognitive Science, Xiamen University, Xiamen, Fujian 361005, China; 2. Department of Communication and Control Engineering, Hunan Institute of Humanities Science and Technology, Loudi, Hunan 417000, China)

**Abstract:** Determinating the search direction too early will make motion estimation fall into a local optimum value in H. 264. An integer pixel search method based on prediction for motion estimation is proposed in this paper. Experimental results show that it can improve search precision and decrease computation time. Intra-frame prediction is another important issue in video coding; we present a fast prediction method based on template-macro-block. Results show that it speeds up coding rate about 80% while not lower image quality.

**Key words:** motion estimation; intra prediction; H. 264

## 1 引言

视频序列在时间上具有很强的相关性, 利用帧间压缩技术可有效的去除图像帧间冗余度, 实现高码率压缩比. 运动估计是帧间压缩技术的重要实现手段, 其具有很大的灵活性, 占整个视频编码计算量较大比重, 因而成为帧间编码的研究热点. 据统计, 采用 1 帧作为参考帧时, 帧间压缩占总编码计算量的 60%, 而采用 5 帧参考图像时更是达到了 90% 左右. 目前, MPEG 和 H. 26x 系列标准普遍采用基于块匹配(BMA)的运动估计法, 其基本思想是将当前帧中待预测的块, 在参考帧中的一定区域内, 搜索与之最接近的块, 这样当前帧中的每个块都可用一个残差块和一个运动矢量来表示, 只需对残差数据和运动矢量进行编码, 从而实现高压缩比.

在 BMA 中, 全搜索算法(FS)精度最高, 但是相应的计算时间和复杂度也最大, 使其不宜实用. 因此研究人

员提出了很多快速搜索算法来替代全搜索算法<sup>[1~4]</sup>. 由 Zhibo Chen<sup>[5]</sup>等人提出的非对称十字型多层次六边形搜索(UMHS)算法, 以搜索准确度高, 运算时间少, 图像质量与全搜索效果相当而闻名, 且已被 H. 264 验证模型 JM 收录采用.

相对以前的编码标准, H. 264/AVC 帧内预测引入了更精细的宏块(16×16)划分, 以亮度为例, 可将亮度宏块划分为 16×16 和 4×4 两种大小的子块, 对于 16×16 块引入 4 种候选预测模式, 对 4×4 子块引入 9 种候选预测模式. 由于块编码模式的增多, 每个块都能在编码模式集中找到最佳的编码模式以达到最佳的编码质量. 然而, 众多的编码模式却使计算复杂度急剧增加, 影响了视频编码实时性的应用<sup>[6]</sup>.

为降低帧内预测的计算复杂度, 同时尽可能保存编码后图像质量, 国内外已经产生了一些快速算法, 取得了很好的效果. 这些快速算法主要可分为两大类: 一是

在宏块划分时,宏块信息做出“预判”,以尽早确定宏块是采用  $16 \times 16$  划分或是  $4 \times 4$  划分,减少“无谓”的计算.二是在已经确定了宏块划分方式后,对候选预测模式进行“过滤”,将小概率预测模式剔除出候选模式集,减少计算;也有一些算法通过分析各个编码模式的后验概率及编码的率失真(RD, rate distortion)信息并结合空间域和变换域的特征,如 SAD(sum of absolute difference)和 SATD(sum of absolute transform coefficient difference),以及相邻块的信息快速选择帧内编码模式<sup>[7]</sup>.第一类算法由于分类方式的多种多样,算法产生的效果也参差不齐;第二类算法是通过减少候选预测模式来加快编码速度的,虽然可大大降低计算复杂度,但由于结果编码模式可能非最优,致使图像失真可能增加,影响后续编码图像质量.

针对一些算法过早的确定搜索方向,容易陷入局部最优点缺失搜索准确度的情况,本文提出一种新的基于搜索方向预测的运动估计算法,与现有的算法相比,该算法在运算时间和计算量方面有着巨大的优势,同时图像质量也达到了与全搜索算法相当的效果.为提高编码速度,本文提出一种新的帧内预测算法,实验结果表明,该算法在图像质量未明显失真的情况下,大大提升了编码速度,是一种可行的算法.

## 2 新的运动估计算法

运动估计算法容易陷入局部最优陷阱而降低搜索精度,为此,本文提出了一种新的运动估计算法,可很大程度上避免局部最优陷阱.算法大致可分为三个步骤:初始搜索点预测,整像素运动搜索,搜索终止策略.

### 2.1 运动矢量起点预测

在运动矢量起点预测中,准确的确定初始搜索点,对后续搜索将起到极其重要的作用.式(1)显示了初始搜索点与初始搜索运动矢量的关系.

$$\begin{aligned} start.x &= origin.x + mv.x \\ start.y &= origin.y + mv.y \end{aligned} \quad (1)$$

其中  $start.x$ ,  $start.y$  是待预测块起始搜索点,  $origin.x$ ,  $origin.y$  是参考图中的块坐标,而  $mv.x$ ,  $mv.y$  是初始预测的运动矢量.

因此,确定初始搜索点也就是确定初始搜索运动矢量.根据大量实验结果显示,待预测块的初始搜索矢量与周围已编码块的运动搜索矢量有很密切的关系,表1显示了周围块对待预测块的影响,各块的位置关系如图1所示.

从表1中数据就可看出,待预测块初始运动矢量平均大约有70%的概率与周围块相关,因此用已编码块的运动矢量来预测初始运动矢量是完全可行的.考虑

到视频图像有时会出现一些极端运动情况,此时采用上述方法预测难免使预测值有些失真,本文采用了一种改进的加权法,分别计算了周围块的权重值,然后交叉与矢量值相乘,这样最大程度的避免了当一个运动矢量极大时,对预测值产生的干扰.式(2)列出了权值计算公式.

表1 A, B, C块对待预测块E的影响

| 测试序列          | A (%) | B (%) | C (%) |
|---------------|-------|-------|-------|
| Table Tennis  | 58.8  | 53.7  | 47.5  |
| Foreman       | 57.8  | 66.5  | 45.6  |
| Capphone      | 66.2  | 72.9  | 59.5  |
| Trevor        | 75.6  | 76.6  | 69.8  |
| Coastguard    | 63.7  | 56.9  | 45.7  |
| Silence       | 87.5  | 87.7  | 84.7  |
| Miss American | 84.5  | 83.2  | 78.0  |
| Average       | 70.5  | 71.0  | 61.5  |

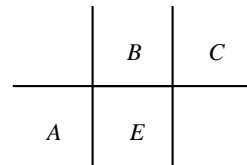


图1 各块的位置关系

$$\begin{aligned} \alpha.A &= \frac{|mv.A|}{|mv.A| + |mv.B| + |mv.C|} \\ \alpha.B &= \frac{|mv.B|}{|mv.A| + |mv.B| + |mv.C|} \\ \alpha.C &= \frac{|mv.C|}{|mv.A| + |mv.B| + |mv.C|} \end{aligned} \quad (2)$$

假定A块的运动矢量出现异常大(出现异常小状况类似),定义初始矢量,计算公式如式(3)

$$\begin{aligned} pred.mv.E &= Median((\alpha.C \times mv.A \\ &\quad + \alpha.B \times mv.B \\ &\quad + \alpha.A \times mv.C), \\ &\quad mv.B, mv.C) \end{aligned} \quad (3)$$

这样做在一定程度上排除了由于预测矢量出现异常所导致的预测不准确问题,消除了候选预测矢量的不均匀,增加了候选预测矢量的可信度.

### 2.2 整像素运动估计策略

为了改变一些快速算法经常会陷入局部最优陷阱的缺点,本文提出了一种带有预测性的整像素搜索策略(PIS, Integer pixel Search algorithm based on Prediction),与前面介绍的算法主要的不同点在于,每次搜索不是直接确定下一轮的搜索中心点,而是先搜索最优点所在的可能方向,然后重点在此方向做更精细搜索,确定下一轮搜索中心点.具体搜索步骤如下:

(1) 对初始点周围做四边形搜索,大小为  $search.range$ ,计算出这4个点的SAD值,与中心点SAD值比



达到了预期的目标。

表2 各种算法 PSNR 值与全搜索对比 (dB)

| 测试序列     | PIS   |       | UMHS  |       | FS    |
|----------|-------|-------|-------|-------|-------|
|          | PSNR  |       | PSNR  |       | PSNR  |
| Parkrun  | 33.54 | 0.00  | 33.53 | -0.01 | 33.54 |
| Stefan   | 36.22 | -0.02 | 36.23 | -0.01 | 36.24 |
| Silient  | 36.23 | 0.00  | 36.23 | 0.00  | 36.23 |
| Salesman | 36.21 | 0.00  | 36.21 | 0.00  | 36.21 |
| Capphone | 37.29 | -0.05 | 37.32 | -0.02 | 37.34 |

表3 各种算法帧间预测时间和全搜索对比 (s)

| 测试序列     | PIS   |        | UMHS  |        | FS    |
|----------|-------|--------|-------|--------|-------|
|          | Time  |        | Time  |        | Time  |
| Stefan   | 1.217 | -2.823 | 0.794 | -4.039 | 4.040 |
| Silient  | 0.830 | -3.508 | 0.329 | -4.009 | 4.338 |
| Salesman | 0.172 | -0.64  | 0.109 | -0.703 | 0.812 |
| Capphone | 0.233 | -0.878 | 0.171 | -0.94  | 1.111 |

从上面的实验结果还可看出, PIS 算法时间优势比较明显, 平均所用时间仅为全搜索的 22.85%, UMHS 算法为全搜索算法的 14.01%, 还是有一定差距, 其原因是 UMHS 算法采用了早期终止策略等优化措施减少了运动估计的时间。图 6 是视频序列 Stefan 在各种码率带宽下采用以上三种算法的 PSNR 值比较。

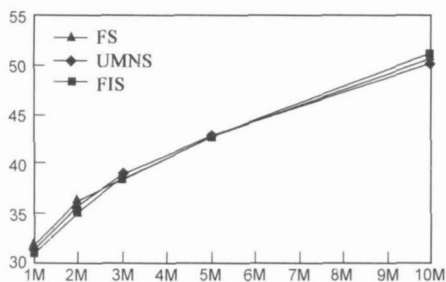


图6 不同带宽下三种算法 PSNR 值的比较

由上面的分析可以看出, 本文提出的 PIS 算法是一种比较成功的算法, 在完整的保持了图像质量的前提下, 使运动估计时间和计算复杂度大幅降低, 而且算法实现比 UMHexagonS 算法简单, 便于硬件实现, 具有很好的应用前景。

### 3 新的帧内预测算法

帧内预测算法大体上分为两个步骤: 首先是判断如何划分宏块, 其次是对划分后的子块如何选择编码模式进行预测。本文针对这两个步骤, 分别提出了改进措施。

#### 3.1 宏块对比划分方式

以前的很多宏块划分算法都是利用周围已编码宏块的信息作为参考, 来预测本宏块的划分方式但却忽略了作为待预测宏块本身所蕴含的信息。本文提出“模

板”块的概念, 利用“模板”块与待预测块做比较的方法, 如果两个宏块相似则说明图像变化平缓, 采用  $16 \times 16$  划分, 反之则说明图像变化剧烈, 需采用  $4 \times 4$  划分。

“模板”块的选取应该代表宏块变化最平缓的情况, 因此本文采用求平均值的方法, 在宏块划分前对该宏块所有的像素取平均值, 作为“模板”块的像素值。然后采用矩阵相似度的有关概念来判断两个矩阵是否相似。

#### 3.1.1 矩阵相似的有关概念

模糊矩阵: 样品  $i$  和样品  $j$  之间的相似程度表示它们之间的贴近程度, 构成模糊关系  $R$ , 可用矩阵表示为:

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & r_{nr} \end{bmatrix}$$

这样产生的矩阵  $R$  中的每个分量都表示对应的两个矩阵关系的紧密程度。

矩阵  $R$  的求取一般可分为两个步骤<sup>[9]</sup>:

首先把各代表点的提取特征数据化, 以便分析和比较, 也称为归一化, 将数据压缩到  $[0, 1]$  闭区间。计算公式很多, 本文采取的是极值标准化公式  $x = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$ , 计算后  $x$  必定落在  $[0, 1]$  内, 其中  $x$  是像素值。

然后算出矩阵  $R$  的各个分量, 常用的方法有: 欧式距离法、数量积法、相关系数法、指数相关系数法、非参数法、最大最小法、算术平均数法等。

为了减少计算量, 适应实时性的需求, 本文选取一种非常简单的计算方式, 首先设定一个阈值, 然后计算两个矩阵对应点之间是否相似, 如果不相似同时累加不相似点的数目, 当数目大于阈值时则认为整个块与“模板”块不相似, 否则相似。

#### 3.2 候选预测模式计算

宏块决定如何划分后, 如何选择预测模式就成了主要问题。H.264 标准对  $4 \times 4$  划分采用了 9 种候选预测模式, 对  $16 \times 16$  划分采用了 4 种候选预测模式。文献 [2] 经过大量实验证明, 一帧内的宏块大多数都是进行  $4 \times 4$  划分的, 因此在计算候选预测模式上, 绝大部分计算集中在  $4 \times 4$  子块的 9 种预测模式上。因此, 如何确定  $4 \times 4$  子块的预测模式是影响编码计算的重要因素之一。本文结合搜索窗算法<sup>[10]</sup>和抽样算法<sup>[11]</sup>的算法思想提出改进的计算方法, 在不明显降低图像质量的前提下, 对  $4 \times 4$  预测模式和  $16 \times 16$  预测模式进行了优化。

当宏块的划分模式已经确定后, 就该确定采用何

种预测模式进行预测. 这里根据块划分的不同, 也分两种情况:

第一种: 当宏块未被划分, 即保持  $16 \times 16$  大小, 采用 4 种预测模式进行预测 - 垂直、水平、直流和平面预测. 在这一步骤, 本文吸收抽样算法的思想, 同样对这 4 种模式进行了抽样计算, 将抽样取出的点组成 4 个  $4 \times 4$  矩阵, 对这 4 个矩

阵分别用上述 4 种模式进行预测, 其中开销 SAD 值最小的模式为最佳预测模式. 这样做, 避免了原算法同一宏块预测 4 次的缺点, 节省了计算空间和计算时间, 同时组成  $4 \times 4$  块的像素点

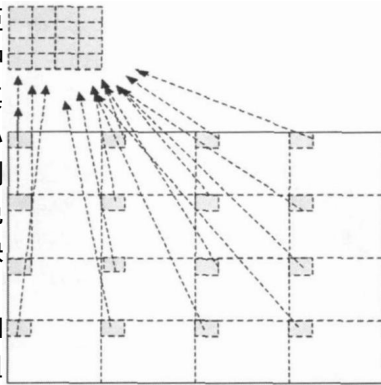


图7  $4 \times 4$  子块抽样组成过程

来自整个宏块的不同区域, 使得计算结果更有信服力. 图 7 显示了其中 1 个  $4 \times 4$  抽样子块的组成过程, 其余子块的组成类似, 只是抽样的位置不同.

第二种: 如果宏块的划分模式是  $4 \times 4$ , 本算法借鉴了文献[3]提出的利用空间域与变换域特点减少候选预测模式的思想, 将 9 种候选预测模式减少为最常用的 3 种 - 垂直预测, 直流预测, 对角线右下预测, 使得计算复杂度大幅下降, 但图像失真未明显增加.

### 3.3 算法分析与实验结果对比

原算法的全搜索策略, 计算每个宏块的最佳划分模式及预测模式需要计算 592 次率失真函数. 本算法根据每个宏块划分的不同, 计算率失真函数的次数也不同, 对  $4 \times 4$  划分, 每个宏块需要计算 12 次, 对  $16 \times 16$  划分需要计算 16 次, 而且计算的像素点数为 16, 与原算法相比, 计算时间大幅下降.

在计算候选预测模式方面, 本文吸取采样算法和搜索窗算法的算法思想, 对  $4 \times 4$  子块预测模式进行了过滤, 对  $16 \times 16$  块预测模式进行了抽样计算, 实验结果证明, 图像质量并未有明显下降, 平均下降 0.45%.

表 4 两种算法 PSNR 值对比

| 视频序列       | 本文提出的算法  |        | 原算法      |
|------------|----------|--------|----------|
|            | PSNR/ dB | / %    | PSNR/ dB |
| Coastguard | 35.39    | - 0.42 | 35.54    |
| Stefan     | 36.67    | - 0.6  | 36.89    |
| M&D        | 37.3     | - 0.4  | 37.45    |
| Carphone   | 38.12    | - 0.39 | 38.27    |

本文采用峰值信噪比(PSNR)作为衡量图像质量的标准, 计算公式如式(5)和式(6)所示. 实验平台采用

windowsXP, VC6.0. 测试模型代码为 JM86, 测试视频序列为 Coastguard( $352 \times 288$ )、Stefan( $352 \times 288$ )、Akiyo( $352 \times 288$ )、M&D( $176 \times 144$ )和 Carphone( $176 \times 144$ ), 编码长度为 50 帧. 表 4 列出了本文提出的算法与原算法的 PSNR 值对比, 表 5 列出了两种算法运行时间对比.

表 5 两种算法预测时间对比

| 视频序列       | 本文提出的算法 |         | 原算法     |
|------------|---------|---------|---------|
|            | Time/ s | / %     | Time/ s |
| Coastguard | 117.346 | - 82.49 | 670.203 |
| Stefan     | 129.17  | - 81.8  | 709.919 |
| M&D        | 24.671  | - 80.8  | 128.546 |
| Carphone   | 24.468  | - 80.72 | 126.922 |

由以上表中数据对比可看出, 本文提出的算法使得编码后图像未明显失真, 平均失真仅为 0.45%, 大幅降低了运行时间, 平均降幅为 81.45%, 这说明本文算法中采取的策略是确实有效的. 图 8 给出了各帧编码时间对比, 编码序列为 Akiyo, 编码 50 帧.

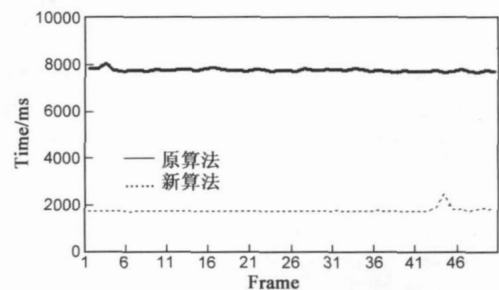


图 8 本文算法与原算法逐帧编码时间对比

由以上实验结果可看出, 本文提出的算法相比原算法编码时间大幅下降, 但图像质量并未明显受损, 是一种可行的算法.

## 4 结语

PIS 搜索算法将四边形搜索和六边形搜索结合起来, 明显的降低了计算量, 加快了编码速度, 同时图像质量得到了很好的保存, 是一种可行的算法, 若在算法中加入类似 UMHS 算法的早期终止策略, 编码效率会进一步提高, 这也将是算法进一步完善的主要突破点.

对于本文提出一种新的宏块划分策略, 将已有算法的思想改进融合进去, 产生了一种新的帧内预测算法. 经实验验证, 本文的算法是行之有效的.

但算法本身也存在不足, 比如在有关宏块划分的阈值确定方面, 只能依靠经验设定, 不一定满足所有视频的特点, 当出现极端情况时, 码率可能会有很大程度的上升, 这也是算法以后需要改进的方向之一. 另外, 由于时间关系, 本文只进行了改进算法与原算法的比较, 没有与其它算法进行比较, 这也是今后需要研究的内容.

## 参考文献:

- [1] 李子印,朱善安.一种快速高效的部分失真块运动估计搜索算法[J].中国图形图像学报,2006,11(4):480-485.
- [2] 李炜,周兵,李波.运动矢量场自适应搜索算法[J].计算机学报,2003,26(2):168-173.
- [3] 杨智勇,郭宝龙,丁贵广.基于预测的方向性菱形快速运动估计算法[J].计算机工程,31(19):180-181,191.
- [4] Cheung C H, Po L M. Novel cross-diamond+hexagonal search algorithm for fast block motion estimation[J]. IEEE Transactions on Multimedia, 2005, 7(1):16-22.
- [5] Chen Z B, Zhou P, He Y. Fast integer pixel and fractional pixel motion estimation for JVT [A]. JVT-F017. doc, 2002, 6th Meeting[C]. Awajisland, Japan :JVT of ISO/IEC MPEG & ITU-T VCEG, 2002. 5-13.
- [6] 冯晓茹,杜建超,周有喜,吴成柯.基于H.264的快速帧内预测算法[J].计算机工程,2008,34(2):229-231.
- [7] 王正宁,彭强,诸昌钤.基于频域特征的H.264/AVC帧内编码模式快速预测[J].西南交通大学学报,2007,42(3):345-351.
- [8] Pao I M, Sun M T. Modeling DCT coefficient for fast video encoding[J]. IEEE Transaction on Circuits and Systems for Video Technology, 1999, 9(4):608-616.
- [9] 杨淑莹.图像模式识别-VC++技术实现[D].北京:北京交通大学出版社,2005.153-159.
- [10] Kim C S, Shih H H, Kuo C J. Fast H.264 intra-prediction mode selection using joint spatial and transform domain features[J]. Elsevier Science, 2004, 17:291-310.
- [11] Meng B J, Oscar C A, Wong C W, et al. Efficient intra-prediction algorithm in H.264[A]. Proceedings of IEEE International Conference on Image Processing[C]. Barcelona, Spain: IEEE Press. 2003. 837-840.

## 作者简介:



李绍滋 男,1963年3月生于湖南常德,现为厦门大学智能科学与技术系主任,教授,博士生导师,主要研究方向为:人工智能与多媒体信息检索、运动目标检测与识别、机器学习与计算机视觉等.

Email :szlig @xmu. edu. cn