

一种面向 H.264 KTA 的快速分数运动估计方法

陆寄远^{1), 2)} 张培钊¹⁾ 朝红阳³⁾

¹⁾ (中山大学信息科学与技术学院, 广州 510275) ²⁾ (广东金融学院计算机系, 广州 510521)

³⁾ (中山大学软件学院, 广州 510275)

摘要 H.264 KTA (key technology area) 在提供更高视频编码性能的同时, 也大大地增加了视频编码的计算复杂度。其中, 1/8 像素运动补偿技术和分数点自适应插值滤波技术的采纳使分数点运动估计占用总体编码时间的比例越来越大。提出一种快速分数点运动估计算法。该算法利用分数点残差曲面的单峰特性和整分数运动向量的相关性, 优化搜索模板和搜索的起始位置。根据在 H.264 KTA 上的实验结果显示, 新算法在大幅提高计算速度的同时, 保持了与全分数点运动估计算法相当的率失真性能。

关键词 视频编码 H.264 KTA 分数点运动估计

中图法分类号: TP391 文献标志码: A 文章编号: 1006-8961(2010)03-367-05

A Fast Fractional Pixel Motion Estimation for H.264 KTA

LU Jiyuan^{1), 2)}, ZHANG Peizhao¹⁾, CHAO Hongyang³⁾

¹⁾ (School of Information Science and Technology, Sun Yat-Sen University, Guangzhou 510275)

²⁾ (Department of Computer Science and Technology, Guangdong University of Finance, Guangzhou 510521)

³⁾ (School of Software, Sun Yat-Sen University, Guangzhou 510275)

Abstract H.264 KTA increases the coding performance at the cost of high computational complexity. Its adoption of adaptive interpolation filter (AIF) and 1/8 pixel MV resolution causes that fractional pixel motion estimation (FPME) exerts more significant influence on the computational complexity of video coding. In this paper we propose an algorithm to speedup FPME by using unimodal characteristics and the MV correlation of fractional pixel motion compensation. According to experiments our algorithm achieves much higher speed than other fast FPME algorithms with neglectable R-D loss compared with FME algorithm.

Keywords video coding H.264 KTA, fractional pixel motion estimation

0 引言

H.264 KTA (key technology area) 是制定下一代编码标准 H.265 的基准模型。KTA 在现有编码标准 H.264 的基础上加入了许多新的视频编码技术^[1]。其中, 分数点自适应插值滤波 (AIF) 和 1/8

像素运动补偿是 KTA 中的关键技术。这两种技术的采用使得 KTA 比起 H.264 JM (joint model), 在相同码率的情况下视频质量有 1~2 dB 的提升, 但同时也令编码的计算复杂度增长了 5~6 倍。计算复杂度的增加主要有下面两个原因。第一个原因是分数点运动估计更为复杂。当采用了 1/8 像素运动补偿技术, 分数点运动估计的候选区域扩充为 15 ×

基金项目: 国家 863 计划项目 (2007AA01Z340); 广东省自然科学基金项目 (07003728)

收稿日期: 2009-07-16 改回日期: 2009-11-22

第一作者简介: 陆寄远 (1976—), 男, 中山大学计算机软件与理论博士研究生。研究方向为视频编码、图像传输。E-mail: dtc005001

@163.com

通讯作者: 朝红阳。E-mail: isschhy@mail.sysu.edu.cn

15 相当于整数点运动估计候选区域的大小。运动估计候选区域的扩大直接导致了分数运动估计时间的增加。另一个原因是分数像素的生成计算更为复杂。1/8 像素和 AIF 自适应插值滤波器的采用, 导致单个分数像素的计算复杂度以及需要计算分数像素的数目都增加了。虽然可以使用不同的插值计算方法减少插值的计算量, 但此计算量的下界和运动估计的搜索点数是成正比的^[2]。因此, 减少分数点运动估计的搜索点数, 不仅可以提高分数运动估计的速度, 而且可以降低分数像素插值的时间。本文针对 KTA 的分数点运动估计, 提出了一种有效的快速算法, 不仅可以用在 1/2 和 1/4 等常用的分数精度中, 而且也可以用在 1/8 分数精度的搜索区域。文中部分结果已经在亚特兰大举办的国际图像处理年会 (IEEE ICIP) 上宣讲^[3]。

视频编码中的运动估计 (ME) 大致上可以分成两个步骤, 分别是整数点运动估计和分数点运动估计。分数点运动估计是整数点运动估计的后续搜索步骤。AIF 和 1/8 像素运动补偿使分数点运动估计会在 16×16 的分数点区域中重复搜索。这对于当前的分数点运动估计方法是一个新的考验。

全分数点运动估计算法 (F 帧 /s) 是最传统的分数点运动估计算法, 能提供最高的搜索精度。在 1/8 像素运动补偿中, F 帧 /s 需要访问 24 个搜索位置才能找到最优点。当使用 AIF 时, F 帧 /s 的搜索点数是 $24 \times n$, 其中 n 是 AIF 需要的迭代次数。因此, 在 KTA 中使用 F 帧 /s 将耗费很多的计算资源。

为了减少 F 帧 /s 的计算复杂度, 目前已经出现了不少快速分数点运动估计方法。这些方法大致可以分为两类。一类是基于模型的分数点快速算法。这类方法通过建立不同的分数点残差曲面模型来预测最优分数运动向量。其中有多项式模型^[4]、1 维和 2 维的抛物线模型^[5]等。这些模型的参数通过最优整数运动向量附近的残差值计算。但实际中的分数点残差曲面大都是无规则的, 难以用简单的数学模型精确描述。通过这些模型预测最优分数运动向量会产生较大的误差, 从而导致了率失真性能损失。

另一类的快速分数点运动估计算法是基于预测的方法。这类方法的思路来源于快速整数点运动估计方法, 借助了相邻宏块运动向量间的相关性, 利用一些简单的搜索模板搜索分数运动向量。CB 帧 /s 方法^[6]使用了中值运动向量作为搜索的起始位置, 然后采用一个钻石型的搜索模板细化搜索。IF 帧 /s^[7]

使用一些简化的技巧加快全分数运动估计算法的搜索速度。EPZ 帧 /s^[8] 根据预测向量与两个最优位置的关系选择不同的搜索模板。倒数分数点运动估计方法 SPME^[2] 利用了运动的相对性搜索分数运动向量。因为这类方法多数是从整数点运动估计方法扩展而来的, 所以缺少了专门针对分数点运动估计的特性而优化的部分。如果利用这些特性进行优化, 分数运动估计算法的速度仍有进一步提升的空间。

本文所提出的快速分数点运动估计方法并不是简单地扩充整数点运动估计方法, 而是在此方法的基础上, 利用分数点运动估计的独有特性构造的一种新方法。本方法的贡献在于: 1) 利用分数点残差曲面的单峰特性, 构造一个能更快向目标位置收敛的动态剪裁模板; 2) 利用当前运动向量整数部分的信息, 准确地预测分数点搜索的起始位置。

1 本文所提出方法

快速运动估计方法中两个重要的部分就是搜索的模板和搜索的起始位置。为了提高分数点运动估计的速度, 针对分数点运动估计, 提出动态剪裁钻石型模板和分数运动向量相关预测方法。

1.1 动态剪裁模板

根据不少文献的统计分析结果^[2,5], 分数点运动向量搜索区域的残差曲面可以认为是一个拥有最小值的单峰曲面。也就是说分数点运动估计的局部最优位置同时可以认为是全局的最优位置。许多的快速分数点搜索方法利用了这一个特性, 采用相对较小的对称模板搜索。目前最经常使用的是钻石型搜索模板。本文其中一部分工作就是进一步简化钻石型模板, 减少其搜索所需的步骤。

为了分析分数点残差曲面, 把单峰曲面简化成 1 维的情况讨论。图 1 是一条单峰曲线 $f(x)$, 横坐标 x 表示位置, 纵坐标表示相应位置上的残差值。其中 x_0 表示分数点区域的最优位置。根据单峰曲线的特性有:

$$\begin{cases} f'(x) < 0 & x < x_0 \\ f'(x) = 0 & x = x_0 \\ f'(x) > 0 & x > x_0 \end{cases} \quad (1)$$

使用钻石型模板搜索, 在 1 维的情况下就相当于搜索当前位置的左右两个搜索点。图 1 是 1 维搜索的一个例子。其中, x_1 是当前搜索位置, x_2 和 x_3 分别是

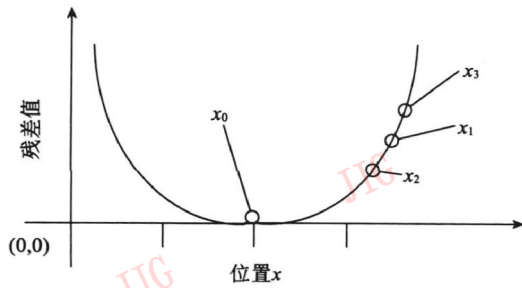


图 1 1 维的单峰残差表面

Fig. 1 One-dimensional cost surface

x_1 左右的两个相邻位置。这 3 点可以出现在曲线的任意位置上, 只要保持它们之间的位置的相对性。那么 x_2 和 x_3 的导数只可能是以下 3 种情况之一:

$$\begin{cases} f'(x_2) < 0, f'(x_3) < 0 & x_2 < x_0, x_3 < x_0 \\ f'(x_2) < 0, f'(x_3) > 0 & x_2 < x_0, x_3 > x_0 \\ f'(x_2) > 0, f'(x_3) > 0 & x_2 > x_0, x_3 > x_0 \end{cases} \quad (2)$$

而不可能出现的情况是:

$$f'(x_2) > 0, f'(x_3) < 0 \quad (3)$$

这是因为 x_2 在 x_1 的左侧, 而 x_3 在 x_1 的右侧。式 (3) 说明 x_2 和 x_3 两个位置的残差值不可能同时比 x_1 小。由此, 推广到 2 维的情况, 则有以下结论: 两个关于中心对称的搜索点的残差值不可能同时比中心点的残差值小。

根据上面的结论可知, 当某个分数点位置的残差值比模板中心的残差值要小时, 那么就不需要再搜索与它相反方向的搜索位置。图 2 是提出的动态剪裁模板的一个例子。搜索的位置和顺序由图中的序号标出。1 表示起始搜索位置。当使用钻石型模板搜索 1 周围的 4 个相邻位置时, 算法检测到 2 和 3 位置的残差值都比 1 的残差值要小, 图中虚线的位置就被剪裁掉不做检测, 直接把模板的中心移动到新的最优位置上。该动态剪裁策略不仅可以应用在钻石型模板上, 而且可以应在任何中心对称的模板上, 加快搜索速度。

由于 KTA 的 $1/8$ 像素通过动态插值产生, 运动估计中访问的 $1/8$ 像素越多, 插值计算的消耗就越大。鉴于此, 上述的动态剪裁模板只应用在 $1/2$ 像素和 $1/4$ 像素上, 跳过所有的 $1/8$ 像素。但是当动态剪裁模板终止的时候, 在最优 $1/4$ 位置周围的 8 个 $1/8$ 位置上再次应用动态剪裁策略搜索, 如图 2 中位置 2 周围的 8 个 $1/8$ 位置。根据本文的实验结果, 这种做法可以在不降低率失真性能的情况下, 提高搜索速度。

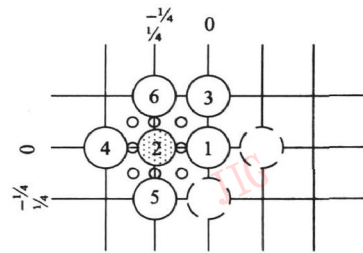


图 2 动态剪裁模板示例

Fig. 2 An example of dynamic pruning search pattern

1.2 分数运动向量预测

除了简化搜索模板外, 优化运动估计的另一个重要的部分是提高预测起始搜索位置的准确度。目前许多整数快速运动估计算法^[8]利用相邻的运动向量作为起始的搜索位置, 即预测点。但分数运动估计仅仅在最优整数运动向量周围的一个像素范围内搜索, 直接使用空域相邻运动向量不能保证预测点都落在该范围内。针对这个问题, 提出了针对分数运动向量的预测方法。

前期工作^[3]指出: 相邻的运动向量如果拥有相同的整数部分, 那么这些运动向量的分数部分也会有很高的相似性。同时, 整数运动估计在分数运动估计之前进行, 所以当前运动向量的整数部分肯定是已知的。根据这两点, 提出使用空域相邻的运动向量和当前运动向量的整数部分预测分数运动向量。分数运动估计的预测点就是与当前运动向量的整数部分高相关的运动向量。

该预测方法分成以下两步。首先, 选择空域相邻的运动向量作为候选预测点的集合。因为运动向量空域相关性强于时域相关性^[8], 所以候选预测点集合只包括上、右上、左上和左相邻运动向量。如果当前运动估计的分块模式不是 16×16 那么上层运动向量^[6]也加入候选预测点集合中。其次, 在候选预测点集合中挑选一些运动向量, 作为真正的预测点。挑选条件是基于候选预测点集合里的运动向量是否在当前分数点运动估计的搜索区域内。满足这个条件的候选预测点就会作为分数点运动估计的预测点。

图 3 是在 $1/4$ 分数点精度下的分数运动估计搜索区域的示例。图中大的实心圆点表示整数点位置, 大的空心圆点表示 $1/2$ 像素的位置, 小的实心圆点则表示 $1/4$ 像素的位置。区域的中心点就是当前最优整数运动向量。其中, 有两个空间相邻的运动向量落在了搜索区域中, 成为预测点。把这两个位

置的残差值比较后, 较小的位置便会成为后续应用动态剪裁钻石型模板搜索的起始位置。

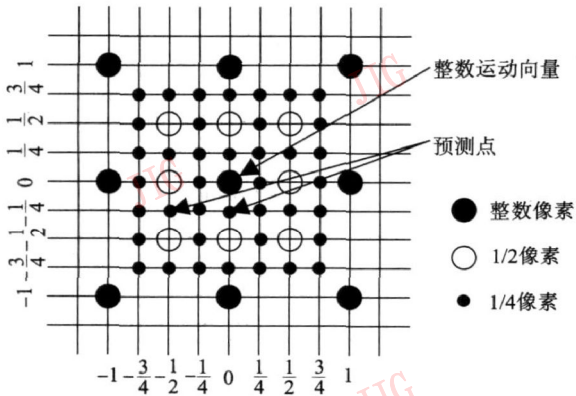


图 3 分数整数运动相关预测示例

Fig. 3 An example of fractional pixel ME predictors

2 实验结果

为了评估本算法的性能, 配合 H.264 KTA 1.8^[1]对 VCEG 定义的标准视频序列^[9]做了大量的实验。因为篇幅有限, 下面只列出高质量 ($QP = 20$) 和低质量 ($QP = 28$) 两种情况下的部分测试结果。其中 Akiyo, Foreman, Mobile, Tempete 和 News 是 CIF 序列, City, Bigships 和 Crew 是 720p 的高清序列。所有序列的编码帧率为 30 帧/s, 编码结构为 IPPP。KTA 中使用分数点自适应插值滤波 (Use Adaptive Filter = 1) 和 1/8 像素运动补偿 (MVR resolution = 1), 采用了高精度的率失真优化选项。

表 1 给出了本算法与 KTA 中两种快速分数运动估计算法: SUMH 和 EPZ 帧/s^[8]在计算复杂度上的比较结果。计算复杂度的量度用相对于 F 帧/s 算法计算复杂度的百分比表示。各算法的计算复杂度等于各自的搜索点数量除以 F 帧/s 算法所消耗的搜索点数。F 帧/s 算法的计算复杂度是 100%。

从表 1 中看出, 本算法在各种情况下 (不同的序列和不同的量化参数) 都拥有最快的速度。本算法与 F 帧/s 算法相比平均提高 6% 的速度, 与 KTA 中的两种快速运动估计算法相比则仍有 24% 和 17% 的提高。分辨率和量化参数 QP 对各快速算法的影响都不明显。而对快速算法性能影响比较明显的因素是运动情况。Akiyo 和 News 是其中较为静止的序列。本算法和 EPZ 帧/s 算法作用在这两个序列上, 算法速度有显著的提高。这是因为这两种算法都采用了较小的搜索模板, 在运动不大的情况

表 1 快速分数点运动估计算法相对于 F 帧/s 的计算速度

Tab. 1 Relative computational complexity of different fast FPME algorithms to F 帧/s

| 序列 | QP | 相对于 F 帧/s 的计算复杂度 % | | |
|----------|----|--------------------|---------|------|
| | | S-UMH | EPZ 帧/s | 本文算法 |
| Akiyo | 20 | 52 | 35 | 26 |
| (CIF) | 28 | 55 | 36 | 23 |
| Foreman | 20 | 54 | 50 | 38 |
| (CIF) | 28 | 57 | 51 | 41 |
| Mobile | 20 | 67 | 64 | 59 |
| (CIF) | 28 | 65 | 63 | 57 |
| Tempete | 20 | 69 | 62 | 51 |
| (CIF) | 28 | 71 | 63 | 48 |
| News | 20 | 53 | 38 | 22 |
| (CIF) | 28 | 51 | 40 | 25 |
| City | 20 | 77 | 79 | 53 |
| (720p) | 28 | 75 | 76 | 52 |
| Bigships | 20 | 64 | 62 | 31 |
| (720p) | 28 | 64 | 65 | 30 |
| Crew | 20 | 51 | 59 | 31 |
| (720p) | 28 | 53 | 59 | 33 |

下减少了搜索点数。而 SUMH 算法则采用了与 F 帧/s 相似的固定搜索模板, 所以对运动情况不敏感。

表 2 给出了本算法相对于 F 帧/s 算法的率失真性能。因为 F 帧/s 算法是理论上最准确的分数运动估计算法, 所以以该算法的率失真性能作为基准。本文在信噪比 (PSNR) 和码率 (Bitrate) 两方面与 F 帧/s 算法比较。表中的 $\Delta PSNR$ 和 $\Delta Bitrate$ 由式 (4) 计算:

表 2 本算法相对于 F 帧/s 的率失真性能

Tab. 2 Relative R-D performance of the proposed algorithm to F 帧/s

| 序列 | QP = 20 | | QP = 28 | |
|----------|-------------------|--------------------|-------------------|--------------------|
| | $\Delta PSNR$ /dB | $\Delta Bitrate$ % | $\Delta PSNR$ /dB | $\Delta Bitrate$ % |
| Akiyo | -0.06 | -0.3 | -0.13 | -0.68 |
| Foreman | 0.01 | 0.2 | -0.04 | -0.80 |
| Mobile | 0.1 | 0.03 | 0.11 | 0.69 |
| Tempete | 0 | 0.06 | -0.02 | -0.23 |
| News | 0.01 | 0.09 | 0.02 | 0.20 |
| City | 0.12 | -0.01 | -0.21 | -0.46 |
| BigShips | 0.09 | -0.09 | 0.2 | 0.88 |
| Crew | 0.04 | -0.72 | 0.09 | 0.14 |

$$\Delta PSNR = PSNR_{\text{proposed}} - PSNR_{\text{帧/s}}$$

$$\Delta \text{Birate} = \frac{\text{Birate}_{\text{proposed}} - \text{Birate}_{\text{帧/s}}}{\text{Birate}_{\text{帧/s}}} \quad (4)$$

式中, $PSNR_{\text{proposed}}$, $\text{Birate}_{\text{proposed}}$ 和 $PSNR_{\text{帧/s}}$, $\text{Birate}_{\text{帧/s}}$ 分别是本文算法和 F 帧 /s 算法的信噪比和码率。

根据表 2 中的数据, 本文算法无论是在 PSNR 或是在码率上, 都与 F 帧 /s 算法有着相当接近的性能。在 PSNR 上的差别是 + 0.12 dB 与 - 0.21 dB。在码率上的差别仅为 + 0.88% 和 - 0.8%。使用较低质量 ($QP = 28$) 编码时, $\Delta PSNR$ 和 ΔBirate 都会相应变小。其原因是当量化参数越大, 分数运动估计算法在精度上的细微差别会被忽略掉。而不同的分辨率, 不同序列的运动和纹理特征, 对视频质量和码率的影响不明显。总的来说, 本文算法与 F 帧 /s 方法相比在率失真性能上的损失是微乎其微的。

3 结 论

提出了一种适合 H.264 KTA 的快速分数点运动估计算法。首先, 动态剪裁模板比传统的搜索模板更快速地向最优位置收敛。其次, 利用了整数和分数运动向量的相关性对分数运动向量预测, 提高了起始搜索点的准确度。通过与其他快速分数点搜索方法相比, 本文算法能进一步减少 KTA 中分数运动估计的计算复杂度。如果配合动态的插值计算方法, 还能大大地减少分数点插值的计算量。特别当 AF 和 1/8 像素运动补偿应用在 H.264 KTA 时, 插值计算所耗费的时间是很庞大的, 本文算法更能体现出优势。

参考文献 (References)

- [1] Jung J Tan T K. KTA 1.8 Software Manual [EB/OL]. (2008-04-22) [2008-07-10]. <http://www.itu.int/mtd/T05-SG16-C-0410/en/C0410.doc>
- [2] Minoo K H, Tuong N Y. Reciprocal subpixel motion estimation Video coding with limited hardware resources [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2007, 17(6): 707-718.
- [3] Chao H Y, Lu J Y. A high accurate predictor based fractional pixel search for H.264 [C] // Proceedings of 2006 IEEE International Conference on Image Processing, Piscataway, USA: IEEE Press 2006 2365-2368.
- [4] Suh J W, Jeone J C. Fast sub-pixel motion estimation techniques having lower computational complexity [J]. IEEE Transactions on Consumer Electronics 2004 50(3): 968.
- [5] Chang J F, Leou J J. A quadratic prediction based fractional-pixel motion estimation algorithm for H.264 [J]. Journal of Visual Communication and Image Representation, 2006 17(5): 1074-1089.
- [6] Chen Z B, Xu J F, He Y, et al. Fast integer-pel and fractional-pel motion estimation for H.264/AVC [J]. Journal of Visual Communication and Image Representation, 2006 17(2): 264.
- [7] Hill P, Chiew T K, Bull D, et al. Interpolation free subpixel accuracy motion estimation [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2006 16(12): 1519-1526.
- [8] Tourapis A, Topiwala P. Fast Subpixel Motion Estimation Support for the Enhanced Predictive Zonal Search Scheme [EB/OL]. (2005-10-15) [2008-07-10]. http://wftp3.itu.int/av-arch/jvt-site/2005_10_Nice/JVT-Q079-L.doc
- [9] Tan T K, Sullivan G, Wedi T. Recommended Simulation Common Conditions for Coding Efficiency Experiments Revision 1 [EB/OL]. (2005-10-06) [2008-07-10]. http://wftp3.itu.int/av-arch/video-site/0510_Nic/VCEG-AA10.doc