

Novel Motion Estimation Algorithms Based on Quadratic Prediction

Shengfu Dong, Longfei Gao

Peking University Shenzhen Graduation School, Shenzhen Guangdong
Email: dongsf@pkusz.edu.cn, gaolf@sz.pku.edu.cn

Received: Jan. 29th, 2016; accepted: Feb. 12th, 2016; published: Feb. 19th, 2016

Copyright © 2016 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper presents a motion estimation (ME) algorithm for High Efficiency Video Coding (HEVC), which provides a strategy to speed up the search process significantly, while yielding the same quality performance as the Test Zone Search (TZSearch) scheme in HEVC Test Model (HM). Compared with the H.264/AVC, HEVC employs a more complex hybrid coding architecture and a larger size search window, leading to great computational complexity. In order to reduce the complexity, a novel motion estimation algorithm is proposed, in which some limited pixels of certain position in the current search window are utilized to build a quadratic model, and then shrink the search range repeatedly by analyzing the sum of absolute difference (SAD) distribution until the best motion vector (MV) is obtained. The proposed algorithm can be applied to various encoding conditions. Experimental results show that our method can save 40% of computations compared with the HM, with negligible decrease of coding quality.

Keywords

Video Encode, Motion Estimation, Fast Algorithm, Quadratic Prediction

基于二次模型的运动估计快速算法设计

董胜富, 高龙飞

北京大学深圳研究生院, 广东 深圳
Email: dongsf@pkusz.edu.cn, gaolf@sz.pku.edu.cn

收稿日期: 2016年1月29日; 录用日期: 2016年2月12日; 发布日期: 2016年2月19日

摘要

本文提出了一种HEVC视频编码的运动估计快速算法, 该算法可以显著加速搜索过程, 编码效率与HEVC参考软件(Test Model)中的Test Zone Search (TZSearch)平分秋色。较之H.264/AVC视频标准, HEVC引入了更为复杂的混合编码框架和更大尺寸的搜索窗口, 从而带来了更高的计算复杂度。为了降低复杂度, 我们提出了一种新的运动估计算法, 该算法仅利用搜索窗中所选取的少量几个点建立二次曲面模型, 通过对像素的绝对差值和(SAD)分布的迭代分析, 逐步缩小搜索范围, 直到得出最优的运动矢量。该算法可被用于不同的编码环境中, 实验结果表明, 与HM相比, 该算法可以节约大约40%的运动估计算法时间, 仅引起轻微的性能损失。

关键词

视频编码, 运动估计, 快速算法, 二次模型

1. 引言

视频编码技术是通过分析视频数据的特性, 去除视频本身所包含的冗余信息, 以达到数据压缩的一项数字媒体处理的关键性技术。HEVC [1]作为高效视频标准(High Efficiency Video Coding), 由ITU-T的VCEG和ISO/IEC的MPEG组织联合协作制定, 是当今世界最为先进的视频编码标准之一, 较之传统的H.264/AVC, 编码效率提高一倍[2]。

未经压缩的视频数据量是非常庞大的, 给视频信息的存储与传输带来了极大的不便。近几十年来, 随着人们对视频压缩技术的持续研究, 相继推出了H.261、MPEG-1、MPEG-2、H.263、H.264等一系列优秀的编码标准, 使得视频编码器的编码效率不断提高。为了满足人们对于视频内容分辨率、色彩饱和度不断提高的需求, HEVC、AVS2等新一代视频编码标准应运而生, 针对高清视频的压缩性能尤为出色。

新一代视频编码标准HEVC采用了多项新的编码技术, 在视频编码性能上有显著提高, 但同时也引入了极高的计算复杂度, 这十分不利于HEVC的实际应用, 特别是实时性要求较高的场合。因此, 研究新一代视频标准的快速算法, 在保持性能的前提下, 降低算法复杂度具有十分重要的研究意义。

运动估计作为HEVC编码器的关键技术之一, 占用将近一半的编码时间[3]。在运动估计的研究历程中, 国内外学者提出了很多优秀算法, 基本的算法如全搜索(Full Search, FS)算法, 通过遍历搜索窗内的每一个点进行代价函数计算, 选取代价最小的块为最佳匹配块, 该方法匹配效果最好, 但时间复杂度过高。其它方法如三步搜索法(Three Step Search, TSS)、四步搜索法(Four Step Search, FSS) [4]、菱形搜索法(Diamond Search, DS) [5]、六边形搜索法(Hexagon-Based Search, HEXBS) [6]等算法大多是基于模板图案的搜索方法, 相对全搜索可以节省90%以上的编码时间, 但并不适应搜索窗尺寸较大的编码标准, 且易陷入局部最优。为此, 在HEXBS算法的基础上, UMHexagonS (非对称十字搜索与六边形网格搜索)算法[7]对此加以改进, 被广泛运用于H.264/AVC、AVS等视频编码标准中。

2. 相关工作

2.1. 视频编码中的运动估计技术

典型的视频编码系统, 包含预测、变换、量化、熵编码等关键技术。预测技术利用视频内容的时域和空域相关性实现对当前编码块的预测, 仅对当前块和预测块的残差信息、运动矢量等数据作进一步的

处理,剔除了大量冗余信息,从而可以实现对视频内容的大幅度压缩。编码预测主要包括帧内预测和帧间预测两种方法。运动估计(Motion Estimation, ME)是帧间预测的重要组成部分,是编码系统中消除时域冗余、提高压缩比的最基本和最重要的关键技术,其中块匹配法(Block Match Algorithm, BMA)由于算法简单且易于硬件实现,被广泛应用于各视频编码标准中。块匹配法的基本思想是先将图像划分为许多子块,然后对当前帧中的每一块根据一定的匹配准则在前后多个参考帧的一定搜索范围内寻找出当前块的最佳匹配块,由此得到两者的相对位移,即当前块的运动矢量。在 HEVC 标准中,图像序列的当前帧被划分成互不重叠的最大为 64×64 大小的子块,每个子块还可以划分为更小的子块。对于当前块来说,只有运动估计所得运动矢量及当前块和预测块之间的残差信息,经过后续处理后才会被输出到码流中。由此可见,运动估计算法越完善,运动矢量估计越准确,预测误差越小,输出的比特流数据就越小,因此,高质量的运动估计算法是高效视频编码的前提和基础。此外,快速运动估计算法往往会受到搜索范围的影响,当搜索范围越小时,运动向量的预测越容易陷入局部最优,当搜索范围越大时,算法复杂度越高。寻找一个兼顾运算复杂度与精确度、且又不易于陷入局部最优的运动估计算法显得非常重要。

2.2. HEVC 参考软件 TZSearch 算法

在 HEVC 标准的参考软件 HM 中,采用的运动估计方法是 TZSearch [8] (如图 1 所示),TZSearch 的起始点选择有三种可选方案,分别是:(1) 中值预测方案;(2) 当前预测块的左、上、右上三种模式的预测方案;(3) 零运动矢量的预测方案。具体的搜索方法包括两遍菱形搜索和一遍光栅搜索。其中,第一遍菱形搜索从步长为 1 开始,不断以 2 的倍数扩大,步长较小时采用 4 点的菱形搜索,步长较大时采用 8 点的菱形搜索,光栅搜索则按照步长为 5 的固定值,按照光栅的 Z 字型顺序搜索寻找匹配代价最小的块。

2.3. 代价函数模型

目前,衡量块匹配效果好坏的准则主要有平均绝对差(Mean Absolute Difference, MAD)、绝对误差和(Sum of Absolute Difference, SAD)、最小均方误差(Mean Squared Difference, MSD)、差值平方和(Sum of Squared Difference, SSD)、归一化相关函数(Normalized Cross-Correction Function, NCCF) 等几种。选择一个既能精确描述块之间的匹配程度,又能快速得到结果的匹配准则,将会极大地提高搜索速度,改进算法性能。

计算两个块大小相同的像素单元的平均绝对差 MAD 如公式(1)所示,是通过累加两个块中每个点的对应坐标像素值之差的绝对值并求取其平均值而得到的。式中 M, N 是匹配块的高度与宽度尺寸, $f_k(m, n)$ 表示当前 (m, n) 坐标下的像素值, (i, j) 表示位移矢量。MAD 的值越小,则匹配的效果越好,说明两个块的相关性越高。

$$\text{MAD}(i, j) = \frac{1}{M * N} \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)| \quad (1)$$

由于 MAD 平均值的计算涉及乘除法运算,通常用绝对误差和 SAD 代替 MAD,但不会影响匹配效果。SAD 的计算方法见公式(2)。

$$\text{SAD}(i, j) = \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)| \quad (2)$$

最小均方误差 MSD 的计算方法类似 MAD,见公式(3),差别在于 MSD 所求的是方差的绝对值的平均值。MSD 的值越小,则匹配的效果越好,说明两个块的相关性越高。

$$\text{MSD}(i, j) = \frac{1}{M * N} \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)|^2 \quad (3)$$

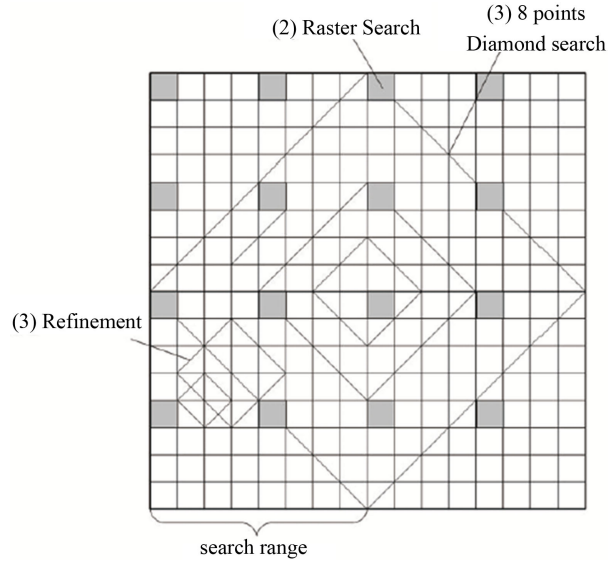


Figure 1. The TZSearch algorithm in HM

图 1. HM 中的 TZSearch 算法

同样地, 由于 MSD 平均值的计算也涉及乘除法运算, 通常用差值平方和 SSD 来代替 MSD, 但不会影响匹配效果。SSD 的计算方法见公式(4)。

$$SSD(i, j) = \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)|^2 \quad (4)$$

归一化相关函数 NCCF 对两个块的对应位置的像素值进行归一化, 方法如公式(5)。归一化的值越大, 匹配效果越好, 说明两个块的相关性越高。NCCF 的计算复杂度较高, 实用性较差, 仅应用于研究中。

$$NCCF(i, j) = \frac{\sum_{m=1}^M \sum_{n=1}^N f_k(m, n) * f_{k-1}(m+i, n+j)}{\left[\sum_{m=1}^M \sum_{n=1}^N f_k^2(m, n) \right]^{\frac{1}{2}} * \left[\sum_{m=1}^M \sum_{n=1}^N f_{k-1}^2(m+i, n+j) \right]^{\frac{1}{2}}} \quad (5)$$

一般而言, 在进行运动估计时, 离最优匹配点越远, 匹配误差值 SAD 越大, 现有的整像素运动估计算法大都采用 SAD 来进行块匹配的计算。

3. 基于二次模型的运动估计快速算法

3.1. 模型介绍

对 HEVC 诸多测试序列的代价分布进行统计, 整个搜索窗的代价分布接近一个连续的曲面。距离曲面的最小值, 即最优点的距离越近, 代价值越小。假设该曲面为一个二次曲面, 利用数学方法求解二次曲面的极值点即为最优结果, 该 SAD 的二次曲面可以用 $F(x, y) = \alpha x^2 + \beta y^2 + \gamma xy + \delta x + \varepsilon y + \zeta$ 来近似估计。

在搜索窗内, 以当前搜索范围的一半为初始步长 R , 选取围绕当前点的 8 个点, 如图 2 所示, 并分别为其分配虚拟坐标。其中当前点的虚拟坐标为 $O(0,0)$, 水平与垂直方向上的四个点分别是 $A(-1,0)$ 、 $B(0,1)$ 、 $C(1,0)$ 、 $D(0,-1)$, 对角线方向上的四个点分别是 $E(-1,1)$ 、 $F(1,1)$ 、 $G(1,-1)$ 、 $H(-1,-1)$, 计算此 9 个点的 SAD 值。

将上述 9 个点的虚拟坐标以及 SAD 值带入公式 $F(x, y) = \alpha x^2 + \beta y^2 + \gamma xy + \delta x + \varepsilon y + \zeta$ 中, 则可得到如下所示的结果:

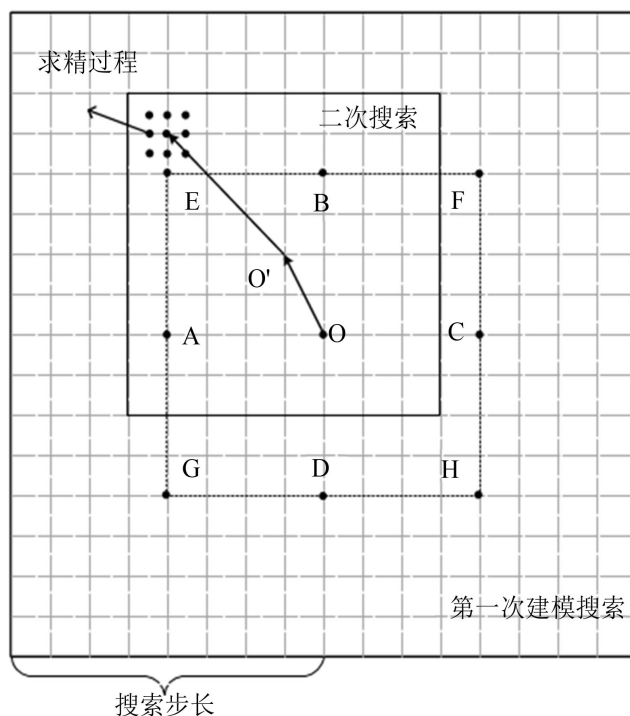


Figure 2. Graphic of quadratic model

图 2. 二次模型示意图

$$\begin{cases}
 F(A) = F(-1,0) = \alpha - \delta + \zeta \\
 F(B) = F(0,1) = \beta + \varepsilon + \zeta \\
 F(C) = F(1,0) = \alpha + \delta + \zeta \\
 F(D) = F(0,-1) = \beta - \varepsilon + \zeta \\
 F(E) = F(-1,1) = \alpha + \beta - \gamma - \delta + \varepsilon + \zeta \\
 F(F) = F(1,1) = \alpha + \beta + \gamma + \delta + \varepsilon + \zeta \\
 F(G) = F(1,-1) = \alpha + \beta - \gamma + \delta - \varepsilon + \zeta \\
 F(H) = F(-1,-1) = \alpha + \beta + \gamma - \delta + \varepsilon + \zeta \\
 F(O) = F(0,0) = \zeta
 \end{cases} \quad (6)$$

因此, 利用 O 、 A 、 B 、 C 、 D 、 E 、 F 、 G 、 H 九个点及其代价 SAD 值, 即可计算出模型函数的所有参数, 计算方法如下式所示:

$$\begin{cases}
 \alpha = -F(O) + \frac{1}{2}(F(A) + F(C)) \\
 \beta = -F(O) + \frac{1}{2}(F(B) + F(D)) \\
 \delta = \frac{1}{2}(F(A) - F(C)) \\
 \varepsilon = \frac{1}{2}(F(B) - F(D)) \\
 \zeta = F(O) \\
 \gamma = \arg \min_{P=E,F,G,H} \sum_{Q=E,F,G,H} |F(Q) - \hat{F}_P(Q)|
 \end{cases} \quad (7)$$

对方程 $F(x, y) = \alpha x^2 + \beta y^2 + \gamma xy + \delta x + \varepsilon y + \zeta$ 进行微分操作, 当 $F(x, y)$ 的 x 分量与 y 分量的偏导数都为 0 时, 所得的点 (x_b, y_b) 即为极值点。当 $\alpha > 0$ 时, 所得曲面为一个凹曲面, 极值点即为极小值点, 否则, 为极大值点, 具体的判断过程如下式所示:

$$(x_b, y_b) = \begin{cases} \left(\frac{\gamma * \varepsilon - 2\beta * \delta}{4\alpha * \beta - \gamma^2}, \frac{\gamma * \delta - 2\alpha * \varepsilon}{4\alpha * \beta - \gamma^2} \right), & \text{if } \alpha > 0 \\ \{(x_i, y_i) | \min_{P=O, A-H} F(P)\}, & \text{if } \alpha \leq 0 \end{cases} \quad (8)$$

由上式可知, 极值点为极大值点时, 丢弃该点, 选择 $O, A-H$ 中 SAD 值最小的点作为搜索结果 (x_b, y_b) 。以上所得极值点的坐标乘以 R 得 $O'(R_{x_b}, R_{y_b})$, 该坐标为真实坐标。

如图 2 中的二次搜索所示, 以 O' 为中心, 搜索步长缩小一半, 即 $R = R/2$, 重复上述步骤, 更新 (R_{x_b}, R_{y_b}) 的值。

如果步长小于等于 8, 停止建模过程, 直接选取周围相邻的八个点, 计算代价 SAD, 选取 SAD 最小的点作为最终的匹配点, 匹配点与起始点的向量即为所求运动向量。

至此, 模型的极值点 (x_b, y_b) 得以求出。

3.2. 整像素与分像素的完整计算过程

3.2.1. 起始点的选择

在搜索窗中选取起始点, 利用中值算法, 选取与当前块相邻的上方、左方、右上方块的运动矢量, 取三个矢量的中值作为起始搜索点。

本算法中放弃了 HM 算法中的零向量起始点和直接选取相邻块向量的可选方法。实验证明, 其它两种方法的准确性较差, 且 HM 中虽实现了另外两种方法, 但并未使用。

3.2.2. 整像素搜索的计算方法

按照上节中的建模方法, 从起始点开始, 初始搜索窗大小选取 64×64 , 以步长 32 选取起始点及其周围的 9 个点建立模型 $F(x, y) = \alpha x^2 + \beta y^2 + \gamma xy + \delta x + \varepsilon y + \zeta$, 通过极值点的计算获取第一次搜索的最优值。然后, 在当前点的周围进行多次搜索, 直到搜索步长缩小至 8, 停止建模, 比较相邻的 9 点选取最终结果。

3.2.3. 分像素搜索的计算方法

分像素的运动矢量计算方法与整像素类似, 利用建模的方法省去为每一个分像素位置插值的过程。

以上一节中的整像素结果作为分像素搜索的起点, 选取当前点和与其相邻的 9 个点建立模型 $F(x, y) = \alpha x^2 + \beta y^2 + \gamma xy + \delta x + \varepsilon y + \zeta$, 通过极值点的计算获取分像素运动矢量的估计值。其中, 极值的计算结果是一个小数, 根据小数的分度确定分像素的具体位置 (x_b, y_b) , 分度大小为 0.25。

$$x_b = \begin{cases} -0.75, & \text{if } x_p < -0.675 \\ -0.5, & \text{if } -0.625 < x_p < -0.375 \\ -0.25, & \text{if } -0.375 < x_p < -0.125 \\ 0, & \text{if } -0.125 < x_p < 0.125 \\ 0.25, & \text{if } 0.125 < x_p < 0.375 \\ 0.5, & \text{if } 0.375 < x_p < 0.625 \\ 0.625, & \text{if } 0.625 < x_p \end{cases} \quad (9)$$

$$y_b = \begin{cases} -0.75, & \text{if } y_p < -0.675 \\ -0.5, & \text{if } -0.625 < y_p < -0.375 \\ -0.25, & \text{if } -0.375 < y_p < -0.125 \\ 0, & \text{if } -0.125 < y_p < 0.125 \\ 0.25, & \text{if } 0.125 < y_p < 0.375 \\ 0.5, & \text{if } -0.375 < y_p < -0.625 \\ 0.625, & \text{if } 0.625 < y_p \end{cases} \quad (10)$$

4. 实验结果及评价

4.1. 测试序列

HEVC 参考软件提供了一系列的测试序列, 涵盖了在分辨率、剧烈运动程度、亮度对比度等方面不同的视频测试文件。该测试序列包括 Class A(2K)、Class B(1080p)、Class C(480p)、Class D(240p)、Class E(720p)等五种分辨率的视频分类, 每一个分类包含不同的视频文件。

4.2. 评价指标与实验对比

视频编码器有可控的量化参数(QP)来控制视频压缩的码率, 为了充分验证算法对于编码器的改进效果, 我们对同一个测试序列进行实验时, 分别测试其量化参数为 22、27、32、37 四种情况, 并对测试结果求平均值。

为了表征快速算法的速度提升, 我们用公式(11)来表示提出的算法较之参考软件的时间节省。

$$TS = \frac{1}{4} \sum_{i=1}^4 \frac{\text{Time}_{\text{HM16.0}}(QP_i) - \text{Time}_{\text{proposed}}(QP_i)}{\text{Time}_{\text{HM16.0}}(QP_i)} \times 100\% \quad (11)$$

其中 $\text{Time}_{\text{HM16.0}}(QP_i)$ 是参考软件 HM16.0 对该序列的当前量化参数 QP_i 的编码时间, $\text{Time}_{\text{proposed}}(QP_i)$ 是使用新提出的运动估计快速算法之后的编码时间。

为了表征编码视频的性能, 最直观的指标是视频编码后的码率(RATE)和峰值信噪比(PSNR)。码率越小, 则压缩后的码流文件越小, 占用的信道带宽越小, 编码器的压缩性能越好; PSNR 是原视频与压缩后视频之间的均方误差相对于 $(2^n - 1)^2$ 的对数值, PSNR 值越大, 则说明压缩客观质量越好, 与原视频差别越小。

由于码率和 PSNR 之间的关系是非线性的, 对于检验不同编码器对于同一测试视频的压缩与质量性能, 直接比较码率和 PSNR 值的大小并不能总是可以得到准确的结果。由于压缩性能与编码质量具有非线性相关性, 为了简单起见, 我们可以假定在相同编码质量下对压缩性能, 或者在相同压缩性能下对编码质量进行比较, 由此得到的判断是合理的、精确的。然而, 不同编码器编码后的码率和 PSNR 值, 很难出现相同的情况。为此, 我们采用 Bjontegaard 提出的算法[9]: 首先, 对于某个具体的视频测试序列, 分别使用两种编码方案, 在四个不同的量化参数 QP 下进行编码, 分别获得其码率与 PSNR 值; 然后, 在不同的编码方案之下, 通过四个不同的量化参数 QP 的码率和 PSNR, 各自拟合出一条码率和 PSNR 的 RD 曲线; 最后, 通过两条拟合曲线差值的积分, 在整个编码码率的跨度上, 得到 PSNR 值的变化量 BD-PSNR (Bjontegaard Delta Peak Signal-to-Noise Rate, 单位为 dB), 在整个 PSNR 值的跨度上, 得到编码码率的变化率 BDBR (Bjontegaard Delta Bit Rate, 用百分率表示)。其中, BDBR 表示在相同的客观质量下, 两种方法的码率压缩情况, BDBR 越小, 表示当前编码器的压缩性能越佳; BD-PSNR 表示在给定的同等码率下, 两种方法的 PSNR-Y 的差异, BD-PSNR 的值越大, 表示当前编码器的质量损失越小。

4.3. 实验结果

按照上节评价指标, 分别测试 HEVC 的参考软件 HM16.0 与本文提出的算法, 按照 Bjontegaard 给出的方法, 得到 HM 的运动估计算法与本文算法的对比结果, 如表 1 所示。

由表 1 可以看出, 该算法可以带来大约 31%~50%的时间节省, 平均码率上升为 1.0%, 平均的时间节省为 41.5%, 亮度分量的质量损失约为 0.005dB。编码效率的损失非常微弱, 以至于可以忽略不计, 但由此带来的时间节省是非常可观的。

通过比较两种编码算法的 RD 曲线图, 即码率与损失的分布曲线图, 可以衡量这两种算法的编码结果是否吻合。如图 3 所示, 通过比较本文算法与参考软件的编码 RD 曲线, 二者吻合程度较好, 该算法并未改变源视频序列的内容。

Table 1. The coding quality and complexity reduction compared with the HM

表 1. 提出算法与参考软件各项测试结果对比

类型	序列名	Δ PSNR-Y (dB)	BDBR (%)	BD-PSNR (dB)	时间节省(%)
A	Traffic	-0.001	0.9	-0.05	52.3
	PeopleOnStreet	-0.004	0.9	-0.05	47.7
B	Cactus	-0.003	1.0	-0.02	32.2
	ParkScene	-0.001	0.7	-0.06	39.1
	Kimono	-0.013	1.3	-0.05	36.1
C	BQMall	-0.004	0.8	-0.08	42.2
	PartyScene	-0.005	0.7	-0.06	43.3
	BasketballDrill	-0.004	1.3	-0.04	46.6
D	RaceHorses	-0.008	1.0	-0.06	38.3
	BasketballPass	-0.004	1.2	-0.09	48.5
	BQSquare	-0.007	1.2	-0.07	42.7
E	Johnny	0.003	1.3	0.06	43.7
	Vodyo3	-0.003	0.9	-0.04	38.8
	FourPeople	-0.005	1.1	-0.06	31.3
平均值		-0.005	1.0	-0.048	41.5

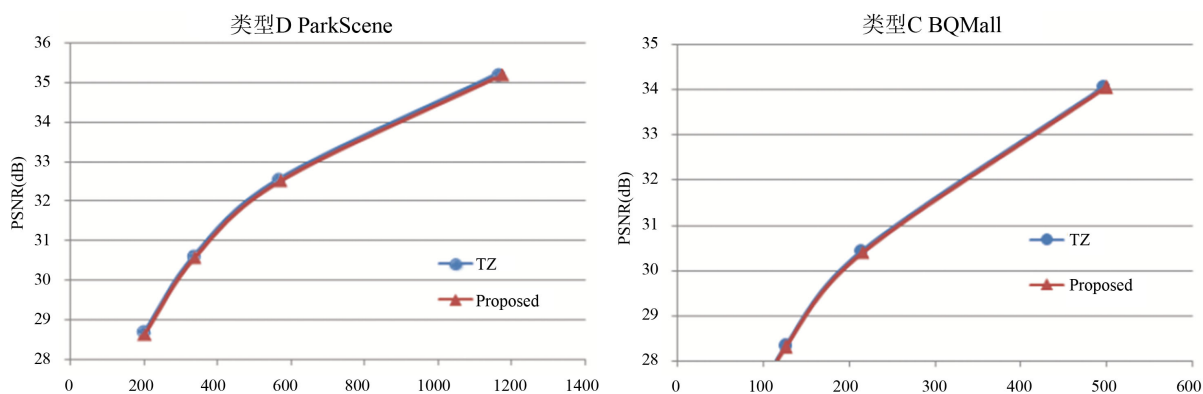


Figure 3. The RD curve of the proposed algorithm and the TZSearch

图 3. 提出算法与参考软件的 RD 曲线图

5. 结论

运动估计作为视频编码系统中的关键技术之一, 占用着接近一半的计算复杂度, 如何在保证运动估计准确性的前提下有效降低算法的计算复杂度, 一直是视频编码技术研究的重要课题。本文提出了一种快速运动估计算法, 摒弃传统的块匹配方法, 选择分析代价分布, 利用有限的几个点, 通过极值的微分计算方法, 获得最佳的整像素运动估计与亚像素运动估计结果。实验结果证明, 该算法在保证编码效率基本不变(大约 1% 的轻微质量损失)的前提下可以节省超过 40% 的计算时间, 算法的加速效果较为明显。

随着人们对视频质量的视觉需求不断提高, 视频编码技术发展也会日新月异。对于高清、超高清视频序列, 在一定程度上意味着采用的搜索窗口会越来越大, 编码单元的大小也会越来越大, 传统的基于模板匹配的运动估计算法的计算复杂度也会随之大幅提高, 本文提出的建模方法有效地利用已得点的代价相关性, 大幅提高了视频编码器的整体运行速度; 同时, 对于如 H.264/AVC、AVS2 等其它视频编码标准, 具有很好的适用性。

致谢

在本研究过程中, 得到了北京大学深圳研究生院信息工程学院院长王文敏教授的大力支持和帮助, 在此表示衷心的感谢。

基金项目

本项目由深圳市海外高层次人才资金资助(20130408-183003656)。

参考文献 (References)

- [1] JCT-VC (2013) High Efficiency Video Coding (HEVC) Text Specification Draft 10 (for FDIS & Consent). JCTVC-L1003_v132, JCT-VC Meeting, Jan. 2013.
- [2] Sullivan, G.J., Ohm, J.-R., Han, W.-J., Wiegand, T. and Wiegand, T. (2012) Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, **22**, 1649-1668.
- [3] Koga, T., Iinuma, K., Hirano, A., Lijima, Y. and Ishiguro, T. (1981) Motion Compensated Inter Frame Coding for Videoconferencing. *Proceedings of National Telecommunications Conference*, New Orleans, November 1981, G5.3.1-G5.3.5.
- [4] Po, L.M. and Ma, W.C. (1996) A Novel Four-Step Search Algorithm for Fast Block Motion Estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, **6**, 313-317. <http://dx.doi.org/10.1109/76.499840>
- [5] Zhu, C., Lin, X. and Chau, L.-P. (2002) Hexagon-Based Search Pattern for Fast Block Motion Estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, **12**, 349-355. <http://dx.doi.org/10.1109/TCSVT.2002.1003474>
- [6] Ma, K.-K. and Zhu, S. (1997) A New Diamond Search Algorithm for Fast Block Matching Motion Estimation. *IEEE Transactions on Image Processing*, **9**, 287-290.
- [7] Chen, Z.B., Xu, J.F., Zhou, P. and He, Y. (2003) Hybrid Unsymmetrical-Cross Multi-Hexagon-Grid Search Strategy for Integer Pel Motion Estimation in H.264. *Proceedings of PCS*, St-Malo, April 2003, 17-22.
- [8] Pan, Z.Q., Zhang, Y., Kwong, S., Wang, X. and Xu, L. (2005) Early Termination for TZSearch in HEVC Motion Estimation. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 26-31 May 2013, 1389-1393.
- [9] Bjontegaad, G. (2001) Calculation of Average PSNR Differences between RD Curves. Document VCEG-M33, Apr.