

基于多分辨率的 H.264 多参考帧运动估计算法

范志浩, 徐平, 范影乐, 庞全

(杭州电子科技大学生物医学工程与仪器研究所, 杭州 310018)

摘要: 提出一种基于多分辨率的 H.264 多参考帧运动估计算法, 通过引入多分辨率的思想, 迅速得到第 1 个参考帧的运动矢量, 利用该运动矢量和已建立的临时矢量映射表预测其余参考帧的运动矢量, 无须对多参考帧中的每一帧都进行同等复杂度的运动搜索。在 x264 编码平台下的实验结果表明, 该算法在保持 H.264 编码器编码性能的同时, 可显著提高编码器的编码速度。

关键词: H.264 视频编码标准; 多分辨率; 多参考帧; 运动估计

Multi-reference Frame Motion Estimation Algorithm for H.264 Based on Multi-resolution

FAN Zhi-hao, XU Ping, FAN Ying-le, PANG Quan

(Biomedical Engineering & Instrument Institute, Hangzhou Dianzi University, Hangzhou 310018)

【Abstract】 This paper proposes a multi-reference motion estimation algorithm for H.264 based on multi-resolution. It imports the multi-resolution to get the motion vector rapidly, and utilizes the motion vector and the temporary motion vector map to predict optimal motion vector for a reference frame, which does not need the same search complexity. Experimental results under x264 software platform show that the algorithm can decrease the multi-reference motion estimation time notably, while the quality of video image degrades little.

【Key words】 H.264 video coding standard; multi-resolution; multi-reference frame; motion estimation

1 概述

H.264 支持多参考帧运动估计(Multiple Reference Frame Motion Estimation, MRME), 将运动搜索范围从原来的一个参考帧扩展为多个参考帧, 有助于找到更加精确的匹配块, 从而提高编码效率。但是, H.264 的高性能是以高计算复杂度为代价的, 特别是采用多参考帧后, 极大地增加了处理器的运行负担。因此, 研究基于 H.264 的快速运动估计算法是 H.264 标准走向实时应用的关键。

在整个编码过程中, 运动估计是计算复杂度最高的模块, 也一直是研究的热点。文献[1]提出了一种低计算复杂度的多参考帧快速运动估计算法, 其主要思想是通过保存单帧运动估计结果来预测多帧运动估计结果, 不需要对每个参考帧进行同等复杂度的运动搜索, 从而达到提高多参考帧运动估计速度的目的。但是该算法的第 1 步采用完整的单一参考帧运动估计, 所以, 计算复杂度仍然很大。多分辨率思想可以提高搜索速度, 在单帧运动估计中已有初步的应用^[2]。文献[2]采用多分辨率运动搜索方法, 但是该方法的 1:16 滤波采样比例过大, 导致原始图像细节特征丢失过多, 最终影响到图像编码质量。

本文提出一种基于多分辨率的 H.264 多参考帧快速运动估计算法。先对第 1 个参考帧的运动估计采用多分辨率搜索, 迅速得到单帧运动矢量。再通过保存单帧运动矢量来预测多参考帧运动矢量的算法, 预测当前编码宏块在其余参考帧的运动矢量。接着在运动矢量预测值附近进行小范围的搜索以得到更加精确的运动矢量, 最后得到多参考帧运动估计结果。在 x264 编码平台下的实验结果表明, 与全搜索(Full Search, FS)多参考帧运动估计算法^[3]和文献[1]的多参考帧运动估计算法^[4]相比, 本算法在保持 H.264 编码器编码性能的同时, 运

动估计速度分别提高了 65% 和 37% 左右。

2 基于多分辨率的多参考帧运动估计算法

2.1 第 1 个参考帧的运动估计

本文采用的多分辨率图像金字塔结构如图 1 所示, 2 个相邻级图像之间按 1:4 抽样滤波, 即第 j 级图像的像素值由第 $j-1$ 级图像对应的 4 个像素求平均得到。本文算法的编码帧和参考帧图像都采用金字塔结构存储抽样滤波后的子图像。金字塔每增加一级, 空间分辨率在水平和垂直方向减半, 每级宏块的大小也随着子图像空间分辨率的减半而减半。

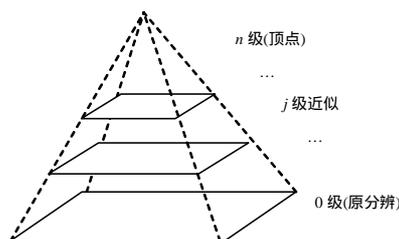


图 1 图像金字塔结构

多分辨率搜索过程如下:

- (1) 在原始分辨率下预测搜索起点。
- (2) 将该搜索起点逐级映射到最低级分辨率下, 得到最低分辨率下的搜索起点。
- (3) 在最低分辨率下搜索得到运动矢量, 并逐级映射回原

基金项目: 浙江省科技计划基金资助重点项目(2006C23047)

作者简介: 范志浩(1982-), 男, 硕士研究生, 主研方向: 模式识别, 生物信息处理, 图像压缩, 视频信号处理与通信; 徐平, 讲师、博士; 范影乐, 教授、博士; 庞全, 教授、硕士

收稿日期: 2008-02-10 **E-mail:** fzh_1982@163.com

始分辨率下。

这里需要注意的是，低一级分辨率下运动矢量指向的一个像素点映射到高一级分辨率下对应为4个像素点，所以，需要在高一级分辨率下对这4个像素点再进行一次搜索，才能得到高一级分辨率下的运动矢量。(4)得到原始分辨率下的运动矢量，再经过分像素搜索得到最终的单帧运动估计结果。

本文算法的第1个参考帧运动估计通过引入多分辨率思想，可以迅速得到第1个参考帧运动估计矢量。由于原始分辨率下一个宏块(16×16)的块分割模式包括7种{16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4}，且4×4块在低分辨率下的运动搜索有意义，因此本文算法采用1级多分辨率搜索。同时保证了不会由于采样滤波比例过大而导致编码质量受影响。

2.2 临时矢量映射表的建立

第1个参考帧运动估计后，需要建立临时矢量映射表。图2最右2幅图是当前帧 $f(t)$ 与其第1个参考帧 $f(t-1)$ 之间建立临时矢量映射表的示意图。临时矢量映射表 $MVMAP_{f(t) \rightarrow f(t-1)}$ 保存 $f(t)$ 中每个块的单帧运动矢量。由于H.264中一个宏块(16×16)的最小分割大小为4×4，因此对一个块搜索得到运动矢量后，块大小(像素)与临时矢量表 $MVMAP_{f(t) \rightarrow f(t-1)}$ 中对应位置元素可以按照空间比例4×4:1(即16:1)进行保存。例如，图2中的运动估计是针对整个宏块(16×16)大小的，宏块映射到临时 $MVMAP_{f(t) \rightarrow f(t-1)}$ 中需要相应位置的16个元素来保存同一个运动矢量值。

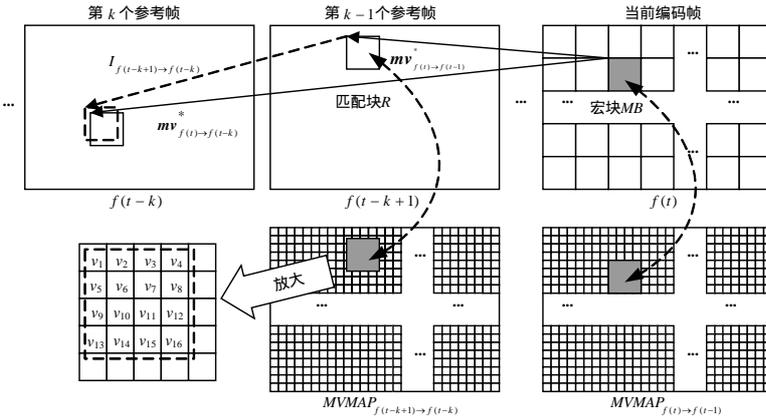


图2 临时矢量映射表的建立和其余各参考帧运动矢量的预测

2.3 其余各参考帧运动矢量的预测

在预测其余各参考帧运动矢量之前，各参考帧在运动估计后都会保存一个临时矢量映射表。按照图2给出的方式将第 $k-1$ 个参考帧中的匹配块 R 覆盖到临时矢量映射表 $MVMAP_{f(t-k+1) \rightarrow f(t-k)}$ 中对应的空间位置。例如，图2中 R 块大小为16×16，则它将覆盖 $MVMAP_{f(t-k+1) \rightarrow f(t-k)}$ 中保存的16个运动矢量元素，记为 $v_1, v_2, \dots, v_{15}, v_{16}$ 。这个覆盖是个近似过程，因为运动矢量 $mv_{f(t) \rightarrow f(t-k+1)}$ 不可能保证总是4的整数倍。各参考帧的运动矢量预测值 $PMV_{f(t) \rightarrow f(t-k)}$ 通过式(1)求得：

$$PMV_{f(t) \rightarrow f(t-k)} = \text{Median}(v_1, v_2, \dots, v_{15}, v_{16}) + mv_{f(t) \rightarrow f(t-k+1)} \quad (1)$$

其中， $k=2, 3, \dots, N$ ， $\text{Median}(v_1, v_2, \dots, v_{15}, v_{16})$ 即求 $v_1, v_2, \dots, v_{15}, v_{16}$ 的平均值。此时，这个预测值还有点误差，需要以该预测值指向的像素点为中心，进行以1个~2个像素为半径的小范围运动搜索，由此得到较精确的当前帧的宏块 MB 在第 $k(k>2)$ 个参考帧中的运动矢量，记为 $mv_{f(t) \rightarrow f(t-k)}$ 。

2.4 算法流程

假设 $f(t)$ 是当前编码帧(t 为当前编码帧的编号)； N 为参考帧的数目，算法的具体步骤如下：

- (1)大循环，选择7种宏块分割类型{16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4}中的一种。
- (2)采用多分辨率搜索，迅速得到第1个参考帧的运动矢量 $mv_{f(t) \rightarrow f(t-k)}$ ，具体过程见2.1节。
- (3)构造临时矢量映射表 $MVMAP_{f(t) \rightarrow f(t-1)}$ (保存各个宏块的 $mv_{f(t) \rightarrow f(t-k)}$)，具体过程见2.2节。
- (4)令 $k=2$ 。
- (5)通过式(1)得到第 k 个参考帧的预测运动矢量 $PMV_{f(t) \rightarrow f(t-k)}$ ，具体过程见2.3节。
- (6)以 $PMV_{f(t) \rightarrow f(t-k)}$ 预测点为中心，在水平和垂直半径分别为 H_{new} 和 W_{new} 的小范围内精确搜索，最后得到第 k 个参考帧的运动矢量 $mv_{f(t) \rightarrow f(t-k)}$ 。
- (7)当 $k=N$ ，到第(8)步，否则， $k=k+1$ ，回到第(5)步。
- (8)从各个参考帧的运动矢量中得到最佳运动矢量值。
- (9)当7种宏块分割类型都循环计算完毕，跳到第(10)步，否则，回到第(1)步选择没有计算过的块分割类型。
- (10)从7种宏块分割类型的运动矢量中选择最佳的运动矢量，完成运动估计。

2.5 算法复杂性分析

本节通过比较各种运动估计算法的搜索整像素点总个数定量分析算法复杂性。设参考帧数为 N ，运动估计搜索区域的水平和垂直半径都为 W ，小范围搜索的水平和垂直半径都为 W_{new} ，引入的多分辨率总级数为 J 。为了方便比较，搜索区域内均采用全搜索的运动搜索方式。统计全搜索多参考帧运动估计算法(记为FS MRME，即每个参考帧采用同等复杂度的运动搜索)、Sung-Eun Kim的多参考帧运动估计算法(记为Kim's MRME)和本文提出的多参考帧运动估计算法(记为New MRME)的运动估计搜索的整像素点总个数，可分别得到式(2)~式(4)：

$$N \cdot (2 \cdot W + 1)^2 \quad (2)$$

$$(2 \cdot W + 1)^2 + (N - 1) \cdot (2 \cdot W_{new} + 1)^2 \quad (3)$$

$$(2 \cdot (W/2^J) + 1)^2 + 4 \cdot J + (N - 1) \cdot (2 \cdot W_{new} + 1)^2 \quad (4)$$

针对QCIF图像，实验参数设置如下：参考帧数为5帧，运动估计搜索区域取半径为16个像素的正方形区域，小范围搜索区域取半径为1个像素的正方形区域，多分辨率取一级。将 $N=5, H=W=16, H_{new}=W_{new}=1, J=1$ ，分别代入式(2)~式(4)，计算结果为5445, 1125和329。由此可知，在3种多参考帧运动估计算法中，本文提出的算法复杂度最低，Kim's MRME算法次之，而FS MRME算法的复杂度最高。所以，本文的算法具有很好的低编码计算复杂度优势。

3 实验结果

实验所采用的硬件环境：AMD Athlon(tm) 64 Processor 3200+2.0 GHz；512 MB 内存；软件环境：Windows XP Professional Service Pack 2；VC++6.0；软件版本采用2007年3月31日发布的x264源码。实验将FS MRME, Kim's MRME和New MRME这3种算法进行了比较。

本文采用2个QCIF(Foreman和Carphone)和2个CIF

(News和Silent)标准测试序列。测试环境是根据JVT制定的编码器测试环境^[4]扩展而来的,参考帧数为5帧,宏块分割类型采用{16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4}。为简便起见,实验采用了IPPP类型的GOP。搜索半径(H, W)都取为16(对于QCIF图像)或者32(对于CIF图像),同时小范围的搜索半径(H_{new}, W_{new})都取为1(对于QCIF图像)或者2(对于CIF图像)。选择不同的帧率和比特率进行压缩性能比较,Foreman QCIF和Carphone QCIF编码时设定为10 Hz和64 Kb/s;News CIF设定为15 Hz和128 Kb/s;Silent CIF设定为30 Hz和256 Kb/s。New MRME算法中的多分辨率搜索在x264平台下实现了一级多分辨率搜索以验证New MRME算法的有效性。

图3给出4个不同测试序列分别采用FS MRME, Kim's MRME和New MRME的率失真性能图。从中可知,在相同比特率下,New MRME的平均PSNR较FS MRME有很少的下降,与Kim's MRME相比,基本不变。

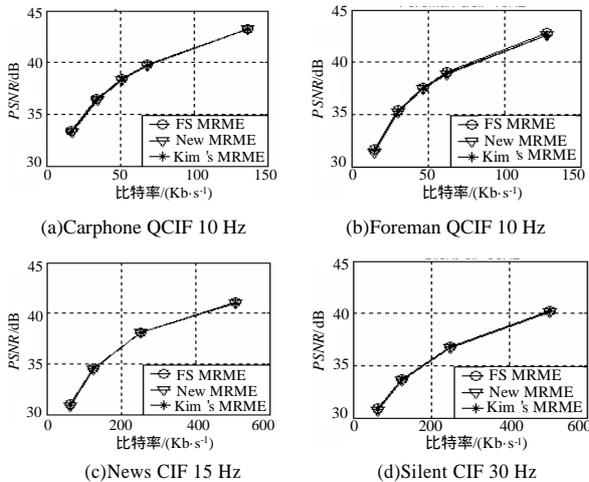


图3 不同图像序列的率失真曲线

表1给出了3种算法编码300帧的运动估计的总CPU耗时。其中,第1列给出4种不同测试序列的名称及其设定的状态;第2~4列统计了3种算法的CPU耗时;第5~6列计算了耗时比, T_{New} , $T_{Kim's}$ 和 T_{FS} 表示分别采用New MRME, Kim's MRME和FS MRME编码300帧的运动估计消耗的CPU总时间。

(上接第234页)

可以看出,由于每个节目长短描述信息的长度是无法预知的,因此采用这种算法相比用数组存储信息节省了大量的空间,又不会产生内存碎片。为了进一步节省内存空间,整块内存大小可以根据实际码流动态选择。从PSI信息中可以得到码流中的频道总数 N 。假设平均每小时一个事件,每个事件的长短描述长度为60 Byte,所有频道的8天EPG需要的内存空间约为: $N \times 24 \times 8 \times 60 = 11.25N$ KB。

可见,上述内存分配方法在解决内存碎片的同时,分配的大小可以动态选择,兼顾了内存空间利用率和灵活性。

5 结束语

本文基于数字广播嵌入式终端在接收数据和动态存储数据时遇到的困难做了针对性研究。采用软件环形缓存策略和双向队列式的内存分配办法,提高系统运行的效率和可靠性,并且通过实验验证了算法的有效性。

表1 运动估计消耗时间比较

标准测试序列	CUP 耗时/s			T_{New}	$T_{Kim's}$
	X264 MRME	Kim's MRME	New MRME		
Carphone QCIF 10 Hz 64 Kb/s 300 帧	27.742	14.800	10.934	0.394 1	0.738 8
Foreman QCIF 10 Hz 64 Kb/s 300 帧	28.169	15.031	11.047	0.392 2	0.734 9
News CIF 15 Hz 128 Kb/s 300 帧	153.710	87.106	46.162	0.300 3	0.530 0
Silent CIF 30 Hz 256 Kb/s 300 帧	220.764	111.615	56.932	0.257 9	0.510 1

综上所述,与全搜索多参考帧运动估计算法和文献[1]的多参考帧运动估计算法相比,本文提出的算法在保持H.264编码器编码性能的同时,运动估计速度分别提高了65%和37%左右。

4 结束语

本文提出的运动估计算法将多分辨率思想引入到多参考帧运动估计中。算法将多分辨率单帧搜索方法和通过保存单帧运动矢量来预测多参考帧运动矢量的方法相结合,达到显著降低多参考帧运动估计计算复杂度的目的。实验结果表明,本文算法可以在PSNR降低很少的情况下,明显降低H.264编码器的运动估计耗时,提高编码效率。

参考文献

- [1] Kim Sung-Eun, Han Jong-Ki, Kim Jae-Gon. An Efficient Scheme for Motion Estimation Using Multireference Frames in H.264/AVC[J]. IEEE Transactions on Multimedia, 2006, 8(3): 457-466.
- [2] Kuo Chih-Hung, Shen Meiyin, Kuo Jay. Fast motion Search with Efficient Inter-prediction Mode Decision for H.264[J]. Journal of Visual Communication and Image Representation, 2006, 17(2): 217-242.
- [3] Joint Video Team(JVT) of ISO/IEC MPEG and ITU-T VCEG. Final Committee Draft of Joint Video Specification[Z]. 2003.
- [4] Sullivan G. Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low-resolution Progressive-scan Source Material[C]//Proc. of ITU-T VCEG'01. Santa Barbara, USA: [s. n.], 2001.

本方法已经在本实验室开发的2个数字广播嵌入式平台(有线和有线/互联网双模)中采用,效果良好。这2种策略也可以推广运用到其他海量数据处理终端中,算法具有普适性。

参考文献

- [1] ETSI EN 301 192 V1.3.1 Digital Video Broadcasting(DVB). DVB Specification for Data Broadcasting[S]. 2003.
- [2] ETSI EN 300 468 V1.5.1 Digital Video Broadcasting(DVB). Specification for Service Information in DVB Systems[S]. 2003.
- [3] ISO/IEC 13818-1 Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Systems[S]. 1997.
- [4] 孟静. 操作系统教程 - 原理和实例分析[M]. 北京: 高等教育出版社, 2001.