

一种基于运动检测的运动搜索模式

刘振宇, 周 莉, 陈 杰

(中国科学院 微电子研究所, 北京 100029)

摘 要: 帧间编码的运动搜索在视频编码的运算量中占据了很大比重, 适当的运动搜索模式算法将有效地减少视频编码计算复杂度. 文中提出了一种基于运动检测的搜索模式算法, 相比传统的 UMhexagonS 搜索模式, 在保持了相当的 PSNR 和码率水平下减少了平均 60% 左右的搜索运算量. 该搜索模式易于硬件实现, 并且对各种视频编码制式都有较好的兼容性.

关键词: 运动检测; 运动搜索; 搜索模式

中图分类号: TP391.41

文献标识码: A

文章编号: 1000-7180(2011)02-0103-07

A Novel Motion-Search-Pattern Based on Motion-Detection

LIU Zhen-yu, ZHOU Li, CHEN Jie

(Institute of Microelectronics of Chinese Academy of Sciences, Beijing 100029, China)

Abstract: The motion-estimation of inter-frame-encoding occupied a large proportion in calculated amount of video encoding, proper search-pattern of motion-estimation could efficiently decrease the complexity of video encoding. This paper proposed a search-pattern based on motion-detection. Compared to traditional UMhexagonS search-pattern, this new search-pattern could decreased average 60% of caculated amount while keeping same PSNR and rate. This new search-pattern is ease to be implemented using ASIC and has a good compatability to various of video standard.

Key words: motion-detection; motion-estimation; search-pattern

1 引言

面向监控的视频编码是计算机图形学重要的应用之一, 由于长时间的连续工作, 对功耗和存储空间有较为严格的要求. 从微电子设计角度来讲, 降低功耗主要从以下两方面着手: 缩小芯片面积和降低运行频率.

为了实现高压缩率, JVT (Joint Video Team) 于 2001 开始制定 H. 264 视频编码标准, 相对传统视频, H. 264 在 PSNR 下可最高节省 50% 的码率^[1], 其 baseline 由于其编码和解码强度非常适合是监控应用. 由于 H. 264 标准提出了基于可变块的多模式帧间编码, 每一种模式都要进行运动搜索, 使运动搜索在帧间编码中最大占到了 60%~80% 的运算

量^[2], 如何有效减少运动搜索就成为一个重要的研究课题. 此前已经研究了多种简化的搜索模式算法, 以达到在类似编码 PSNR 下尽量减少计算量的效果. 为此, 研究者们已经提出了大量快速运动搜索 (Fast Motion Estimation, FME) 模式, 如: 三步法 (TS)、新三步法 (NTS)、预测式搜索区域算法 (EPZS)、非对称六边形搜索算法 (UMHexagonS). 其中, UMHexagonS 由于其在 H. 264 下综合性能最好, 被 JVT 参考代码版本 JM 作为其官方推荐搜索算法, 随后又有大量关于 UMhexagonS 算法简化的研究^[1,3-5].

在实际的工程应用中, FME 算法在对大运动向量 (如 mobile, football) 和大尺幅视频向量 (720P, 1080P) 进行编码时, 由于其运动矢量较大, 无论是

收稿日期: 2010-02-11; 修回日期: 2010-04-12

基金项目: 国家“八六三”计划项目 (2009AA011700)

搜索速度和搜索效果都有一定的局限性. 为了解决大运动向量和尺幅视频向量的视频编码问题, 文中通过运动检测对原视频向量进行分析, 分析其运动向量大小和合运动在视频中所在位置, 然后分对各 MB 进行分析.

基于上面分析, 文中提出了一种新的基于视频编码器的运动检测算法, 并提出了一种基于运动检测算法的搜索模式算法. 实验表明, 文中提出的搜索模式算法相比 JM 代码, 在保持 PSNR 持平甚至略有提升的前提下, 可节省最大接近 90% 的搜索次数. 并且在快速运动向量和尺幅测试向量下表现更为突出.

2 运动、边缘检测和运动搜索算法

运动检测和边缘检测是视频监控应用中重要的组成部分. 运动搜索是指在视频编码过程中寻找最佳运动矢量的运算过程. 虽然其应用背景不同, 但其之间存在着很强的内在联系.

2.1 运动检测算法

运动检测是指从序列图像中将变化区域从背景中分割出来, 它是图像处理和分析的关键技术^[6-7]. 在实际监控应用中的环境是相当复杂的, 天气、灯光的变化将严重地影响原始画面的亮度和色度. 最新的运动检测研究主要集中在两个层次上: 运动分割 (Motion Segmentation) 和 对象分类 (Object Classification). 前者旨在探测视频序列中是否发生了变化并区分其为噪音或运动物体; 后者旨在区分运动物体的具体分类 (行人, 汽车或其他运动物体). 运动分割的主要可分为背景差法 (Background subtraction), 统计学法 (Statistical Method)、相邻帧差法 (Temporal differencing) 和 光 流 法 (Optical flow)^[6].

2.1.1 背景差法, 统计学法和相邻帧差法

背景差法需要先设定一个背景模型, 在视频序列处理过程中通过灰度差或直方图算法判定灰度等特征的变化. 背景差法在背景已知的情况下, 可以提取完整的目标 (前景图像) 图像. 但它受光线、天气等外界条件影响较大^[7]. 统计学法的主要方法是将单个像素或者一组像素的特征提取出来, 并在序列处理过程中更新各 (组) 像素的统计特征 (如期望值、方差和像素间的协方差等). 利用这些特征, 建立前景物体和背景图像模型. 当前图像帧中的每个像素都可以通过和背景模型进行比对将其归类为背景或前景物体^[7]. 相邻帧差法的原理是将图像序列中相邻

的 2-3 帧相减, 随后判断所得到的图像灰度差的绝对值是否大于阈值以确定监控场景有无运动物体. 优点是运算简单, 检测速度快, 对光线变化不敏感, 适用于实时性较高的监控环境^[7].

2.1.2 光流法

光流是空间运动物体在观测面上的像素点运动产生的瞬时速度场, 包含了物体表面结构地和动态行为的重要信息, 光流法经典的公式最初由 Horn 和 Schunck 提出^[6].

光流法的主要计算过程是: 设 t 时刻当前图像上的点 (x, y) 处灰度值为 $I(x, y, t)$, 在时刻 $t + \Delta t$ 时, 这一点运动到 $(x + \Delta x, y + \Delta y, t + \Delta t)$, 对应的灰度值为 $I(x + \Delta x, y + \Delta y, t + \Delta t)$, 假设它与 $I(x, y, t)$ 相等, 即

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) \quad (1)$$

将左边在 (x, y, t) 点用泰勒公式展开, 则可以得到基本光流约束方程:

$$I_x u + I_y v + I_t = 0 \quad (2)$$

写成梯度形式为:

$$(\nabla I)^T U + I_t = 0 \quad (3)$$

其中, $U = (u, v)^T$ 为所求的光流场. 为了解出光流场, 还需要其他约束方程. 光流法的优点是能够检测独立运动, 不需要预先知道场景信息, 并且可用于传感器非静止的情况, 但是光流场可能由于多光源、噪音、透明、遮挡和噪音的情况导致光流场分布并使十分可靠和精确.

2.2 运动搜索算法

上述运动检测和边缘检测算法是应用在视频监控过程中的, 传统算法是独立在视频编码过程之外的, 而运动搜索算法是视频编码过程中的重要组成部分, 其目的是找到兼顾视频质量 (以峰值信噪比 PSNR 衡量) 和编码码流大小的运动矢量. 一般而言, 搜索软件会首先设定一个搜索窗 (Search Window), 并在搜索窗指定的范围之内搜索适宜的运动向量.

为了兼顾视频质量和码流大小, 编码算法一般都会使用一种“码率-失真模式” (Rate-Distortion Model, R-D 模型).

2.2.1 全搜索算法 (Full Motion Estimation)

全搜索算法是指对搜索窗内所有候选点都使用 R-D 模型进行比较, 以 JVT (Joint Video Team, 联合专家视频组) 推荐的 H. 264 开源编码软件 JM 为例, 其单位为宏块 (16×16 的像素块) 其主要流程为:

定义坐标为 (x, y) 的宏块(Macro Block, MB)为距离当前帧最左上角坐标为 (x, y) 的 16×16 像素矩阵,这里只讨论亮度(Y)矩阵和全搜索算法,每一点亮度记为 $Y(x, y)$,宏块的像素范围自 (x, y) 到 $(x+15, y+15)$. 设定一个 $M \times N$ 的搜索窗,对于全部坐标位于参考帧 $(x-M/2, y-N/2)$, $(x+M/2, y+N/2)$ 之间的宏块(参考帧选择因不同编码标准和参考索引而异),设其坐标为 (x', y') ,则其运动矢量为 $(x'-x, y'-y)$,其编码长度依不同标准而异(如较早期编码制式的编码内容为其运动矢量, H. 264 和 AVS 的编码内容为其运动矢量和预测运动向量之差),设定为 m 比特并作为 R-D 模型中的 R. 这里以误差绝对值和(sum of absolute difference, SAD)计算其预测误差,定义:

$$SAD = \sum_{i=0}^{15} \sum_{j=0}^{15} |Y((x'+i), (y'+j)) - Y((x+i), (y+j))|$$

并定义 SAD 为 R-D 模型中的 D. 设定参数 λ 用于权衡编码长度和预测误差,计算

$$Total_SAD_{INTER} = SAD + \lambda \times m$$

并以 $Total_SAD$ 值决定最后选用的预测运动向量. 同时,部分编码标准还将进行帧内预测(Intra Prediction),同时计算帧内预测的 $Total_SAD_{INTRA}$,若 $Total_SAD_{INTRA}$ 小于 $Total_SAD_{INTER}$,则选择帧内编码,这种情况一般发生运动比较剧烈的物体和运动物体进出前景障碍物. 如果参考帧出现杂在当前帧之前的物理时间,称之为前向搜索(其结果为 P 帧),若出现在当前和之后的物理时间,则称之为双向搜索(其结果为 B 帧).

2.2.2 快速搜索算法(Fast Motion Estimation)

全搜索算法计算量非常大,以 D1 尺幅(720×480)为例,全帧共计 1350 个宏块($(720 \times 480)/(16 \times 16)$),若搜索窗为 32×32 每一帧全搜索算法需搜索约 $1350 \times 32 \times 32 = 1382400$ 次像素级搜索,在 H. 264 中共计 7 种编码模式,如果全部进行搜索,共计 16×16 一次 16×8 和 8×16 各两次, 8×8 四次, 8×4 和 4×8 各八次, 4×4 十六次宏块级搜索,共计 $41 \times 1382400 = 56678400$ 次搜索,这种亿次级别的运算的运算压力在高清向量(720P/1080P)编码过程中尤其突出.

针对 H. 264 的计算压力问题,研究人员对快速搜索算法进行了大量研究^[8-9],文献^[10]提出了针对 H. 264 的快速运动搜索算法 UMHexagonS,并被 JVT 的官方开源代码 JM 自 2003 年以来被推荐为

快速运动搜索算法. 图 1 给出了 UMHexagonS 算法的搜索模板. 文献^[1,3-5]对 UMHexagonS 算法做出了进一步简化研究,其中文献^[5]算法被 JM 代码推荐为 Simplified UMHexagonS 算法.

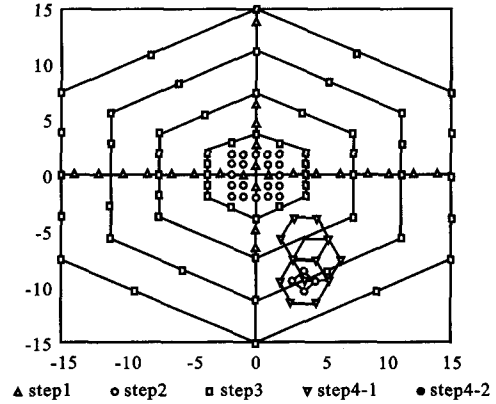


图1 UMHexagonS 算法的搜索模板

为了减少计算次数,快速搜索算法大都加入了提前终止条件判断,当 $Total_SAD$ 小于设定的提前终止条件时即终止搜索.

通过对当前主流 FME 算法进行分析,由于其对于任何测试向量而言,算法搜索步距是固定的,则对于小尺幅向量(如 CIF)和运动较慢的向量,就容易就达到提前中止条件,而对于大尺幅向量(如 720P, 1080P)和运动较快向大向量(如 mobile, football),则需要较多搜索次数才能满足终止条件,对于实时监控编码而言,很难控制总计算量.

3 一种新的运动检测算法

3.1 运动检测算法推导

通过比较运动检测算法和运动搜索算法,可以推导出一种适合于硬件实现的运动检测算法.

背景差法更接近于运动搜索中运动向量为 $(0, 0)$ 的使用长期参考帧的前向帧间搜索,所述的相邻帧差法类似于参考帧为运动向量为 $(0, 0)$ 的双向帧间搜索,而光流法的灰度光流实质更接近于视频压缩码流中的运动向量. 在一定程度上,运动预测的运动向量结果可以准确地反映光流实质.

同时,通过分析 SAD 的特性,可以分析出处于运动物体边缘的宏块拥有更大的 SAD,这个特性有助于将强运动检测的正确性.

3.2 一种新的运动检测算法

在序列编码过程之前设定一个 SAD 阈值 SAD_T , 一个 MV 阈值 MV_T 和一个被激活宏块计数阈值 $ActivatedMB_T$. 以上阈值在序列开始和每一帧开始的时候可以根据过往统计数据更新.

在进行运动搜索并获得运动矢量 MV 后,若当前 MB 最后被选择为帧内预测且选择模式的 SAD 大于 SAD_T ,则当前 MB 标记为 $Activated$,反之则标记为 $UnActivated$;若当前 MB 最后被选择为帧间预测且最终运动矢量绝对值之和 $(|MV_x|+|MV_y|)$ 大于 MV_T ,则标记当前 MB 为 $Activated$,反之则标记为 $UnActivated$. 如果需要对由于背景光线变化造成的编码器误产生运动向量进行滤波,可以同时判断当前帧间编码宏块的 $Total_SAD_{INTER}$ 是否都大于 SAD_T . 当一整帧图像完成全部帧内预测和帧间预测后,若被标记为 $Activated$ 的宏块总数大于 $ActivatedMB_T$,则标记当前帧为 $Activated$,也即检测到当前帧中存在运动物体,编码器可以根据当前帧是否为 $Activated$ 而决定是否对其进行编码. 如图 2 所示.

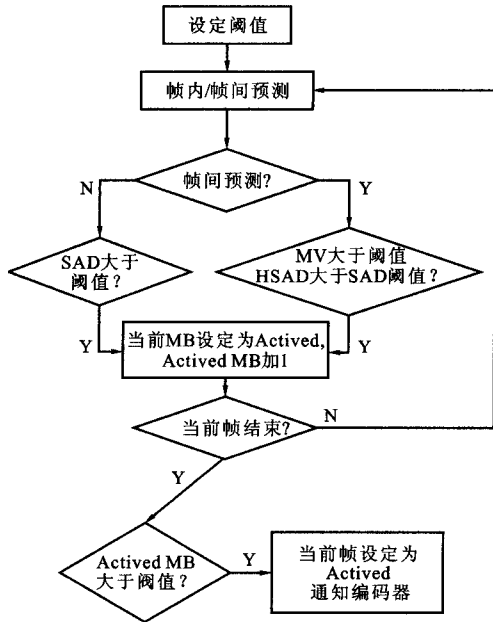


图2 文中提出的运动检测流程

3.3 MV 阈值的自适应更新

在实际应用中,运动物体的运动矢量会由于运动物体本身特征和距离传感器的远近而不同,同时对于同一应用场景,不同时间的运动物体运动矢量大小也会有所不同. 如图 3 所示.

文中提出了自适应阈值更新算法如下:在一帧的编码中,算法会对以下计数:当前含有运动矢量的宏块个数 $MB_COUNT_{mv_exist}$ 和被认为运动矢量有效的宏块个数 $MB_COUNT_{mv_valid}$. 根据场景要求,设定两个阈值:有效阈值下限 ($MV_Valid_Rate_Threshold_{low}$, $MVRT_L$) 和有效阈值上限 ($MV_Valid_Rate_Threshold_{high}$, $MVRT_H$),作为保留运动矢量和忽略运动矢量的一个容忍期望,当一个帧



图3 在实际向量中检测到近景有运动(左)和远景有运动(右)

间预测帧(P 帧)编码结束之后,若: $MB_COUNT_{mv_valid} / MB_COUNT_{mv_exist} < MVRT_L$,也即被运动矢量大于期望,则减少 MV 的判断阈值;若 $MB_COUNT_{mv_valid} / MB_COUNT_{mv_exist} > MVRT_H$,也即被运动矢量小于期望,则增加 MV 的判断阈值.

从表 1 可以看出, $MVRH_H$ 主观最佳结果基本上与活动物体所占画面比例成正比,这也是期望的运动矢量分布范围. 对于运动物体范围不是很固定的向量,可以将 $MVRH_H$ 和 $MVRH_L$ 调高到最大范围,并随后进行自适应修正.

表 1 MV 阈值的自适应更新主观最佳参数结果

实验序列	Container	Paris	foremen
序列特征	远镜头,运动较慢,运动物体占比例中等	镜头稍近,运动物体占画面比例大	近镜头,运动物体占画面比例较大
MVRH_H	0.15	0.47	0.35
MVRH_L	0.04	0.12	0.09
运动物体占画面比例	约 20%	约 45%	约 35%
图像序列预览			

图 4 给出了由表 1 给出参数下的 MV 阈值变化曲线. 可以看出,尽管使用相同的 MV 阈值初始值(16),但是在不同的视频序列下,经过一定帧数的训练, MV 阈值将收敛于适合当前运动目标的、稳定的值上. 经过观察,收敛后的 MV 阈值很好的反映了测试序列的运动速度实质. 收敛需要的帧数称之为训练帧数. SAD 阈值的自适应更新过程与 MV 阈值的自适应更新类似.

4 一种基于运动检测的搜索模式算法

上述运动检测算法只需要在传统的视频编码硬件基础上以很少的硬件消耗实现运动检测功能. 从运动分析的角度来讲,运动检测的结果给出了两方

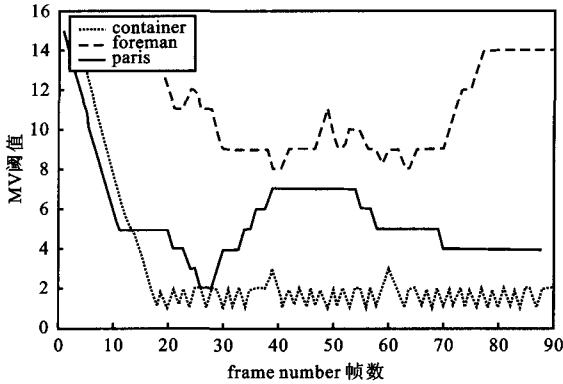


图4 MV_threshold 收敛曲线

面的信息: Activated_MB 给出了在视频一帧中运动所在的区域,而自适应修正的 MVRH_H 和 MVRH_L 给出了运动幅度大小的期望,自适应修正的 SAD_T 则给出了较为准确的搜索提前中止信息.

基于以上信息,文中提出了基于运动检测的搜索模式新算法.如图5所示.

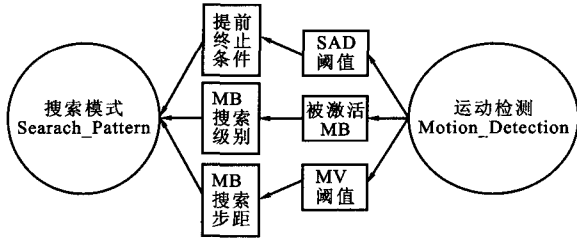


图5 搜索模式和运动检测的关系

4.1 自适应可变步距算法与分级搜索算法

在3.3节中提出的MV阈值自适应修正使MV阈值很好的反应了视频向量的运动实质并且具有很好的收敛性,以MV阈值确定搜索步距可以有效地在保持编码效果的前提下减少总体计算量.

在3.2节提出的运动检测过程中已经得出了每一帧的 Activated_MB 分布,也即一帧中哪些 MB 被认为是包含运动的 MB,也即一帧中的运动区域和物体边缘区域.一般而言,对于运动区域的运动期望会大于其他区域.实验证明,相邻帧的 Activated_MB 分布是有一定相关性的.文中基于此提出了分级搜索策略,若前一帧同位置 MB 已被认定为 Activated_MB 的 MB 被设为最高搜索级(3级),若前一帧同位置 MB 不是 Activated_MB 但同一帧中其周围已编码模块被设定为 Activated_MB(按编码顺序,是左上,上,右上,左四个 MB 中至少有一个被认定为 Activated_MB),则当前 MB 被认定为次级搜索级(2级),边缘 MB 被认定为0级,其余 MB 被认定为1级.图6给出了分级搜索的实例.由图6所示,中间为 paris 向量某一帧的运动检测结果,其中高亮并

被白边线标出的为被确认为 Activated_MB 的宏块;左边给出的是量化信息,标"1"的为 ActivatedMB;右边给出的是搜索级别,分别为0~3级. SAD 提前中止条件设定为自适应更新的 SAD 阈值 SAD_T .

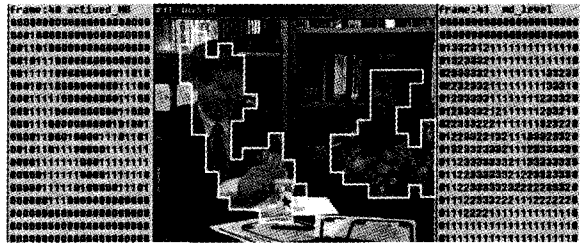


图6 搜索模式和运动检测的关系实例

4.2 分级搜索模式

第4.1节中给出了搜索模式的级别,级别越高,代表对其运动矢量的期望越高,应该给予其更多的搜索点以获得更精确地结果.文中提出的搜索模式分为两步:可变步距搜索和像素单位搜索.

4.2.1 可变步距搜索与像素单位搜索

图7给出了0~3级累加的搜索点,与其他快速搜索模式使用固定的像素间距作为搜索单位不同,本文提出的算法以可变步距为搜索单位.设可变步距搜索单位(即图中圆圈之间的距离)为 sp_radius ,计算:

$$sp_radius = (MVRH_H < 4) ? (1) : (MVRH_H / 4);$$

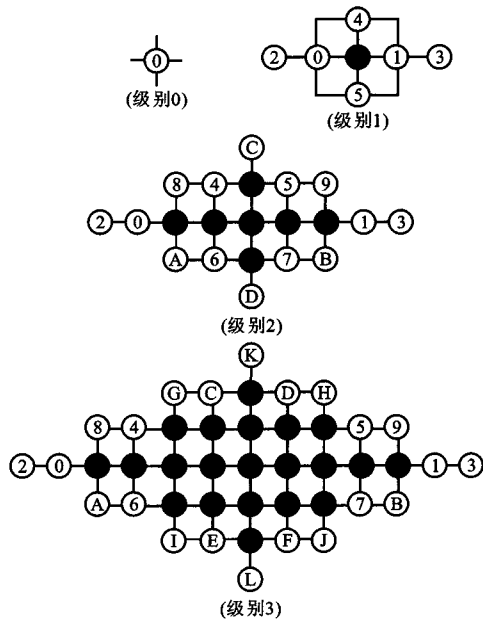


图7 分级搜索算法示意

文中提出的搜索模式是累加的,首先进行低级别搜索,没有满足中止条件就一直搜索到 MB 所在级别为止.图7中的黑色圆圈代表低级别中已经搜

索过的搜索点. 搜索点之间的距离为前述 sp_radius .

例当前 MB 被设定为 2 级, 则首先依次按级别 0 和级别 1 搜索, 若搜索的 $Total_SAD_{INTER}$ 结果小于提前中止条件, 则记录结果, 结束搜索. 若没有满足, 则按第 2 阶段搜索模式搜索, 若第二阶段还没有满足, 则结束搜索并记录到当前为止的最好结果. 同理, 若当前 MB 被设定为级别 3, 则若阶段 2 没有满足提前中止条件则进入级别 3 的增加点, 反之若为级别 1 的 MB 就搜索到级别 1 为止.

分级别的搜索模式同时也为控制整体搜索次数提供了条件.

在完成可变步距搜索之后, 针对步距之间的像素, 按小菱形法进行像素级搜索.

4.2.2 自适应提前中止条件

文献[5]通过对提前中止条件进行计算, 有效地减少了总计算次数, 但是其计算非常复杂. 由于本文提出的算法在运动检测阶段对 SAD 阈值进行了自适应处理, 且 SAD 阈值已经很好的反映了运动实质, 文中提出了一种基于 SAD 阈值和 MB 分级的自适应提前中止条件:

首先设定四个比例系数: a_0, a_1, a_2, a_3 , 分别对应搜索级别的 0~3 级. 则 4 个级别的提前中止条件为当前级别系数与 SAD_T 之积. 如 2 级 MB 提前中止条件即为: $a_2 * SAD_T$. 若当前 MB 搜索结果小于提前中止条件即停止搜索. 后续试验中, $a_0 \sim a_3$ 依次取 0.4, 0.6, 0.7, 0.8. 实验结果表明提前中止条件起到了很好的效果.

4.3 帧级整体搜索次数控制

传统 FME 一个缺陷就是很难控制整体搜索次数, 这对实时硬件编码设计相当不利. 文中算法通过对级别的控制, 有效地控制了对帧级整体搜索次数的控制.

整体搜索次数控制方法如下: 首先根据硬件处理速度(处理器速度、存储器带宽等算法优化结构等)决定一个每秒处理次数上限. 编码系统在编码过程中将在编码过程中实时统计每一帧的总搜索次数, 若当前帧编码使用的总搜索次数高于上限, 则上一帧是将每一 MB 的搜索级别都降一级处理, 而若总搜索次数小于上限, 则可以适当上调搜索级别(如将边缘 MB 上调至搜索级别 1 以提高编码 PSNR).

5 算法实验效果

表 2 给出了文中算法和 JM 代码中提供的 UMHexagonS 算法的实际效果对比(软件版本 JM8.6). 表 3 给出了文中算法和国内近年主要 UMHexagonS 简化算法的效果对比. UMHexagonS 在表中简称 UMHS. 表中“-”代表减少, “+”代表增加, bitrate 单位为 kbps, 表中训练次数指的是每 MB 的平均次数.

从表 2 可以看出, 相对于基于对 UMHexagonS 算法简化的^[1,3-4], 文中的自适应可变步距算法不但在搜索次数上有大幅优化, 并且 PSNR 也有提高. 相对于使用了复杂的提前中止条件的^[5], 不但提前中止条件简单, 是以硬件实现, 且搜索次数控制和 PSNR 也都好于文献[5].

表 2 文中搜索模式和 UMHexagonS 算法性能比较

尺幅	CIF	CIF	DI	DI	720p	1080p
向量	Stefan	mobile	cheers	mobile	parkrun	pedestrian
QP/测试帧数	28/10	28/30	28/50	28/30	28/30	28/50
训练帧数	0	4	0	14	0	0
文中算法平均整数搜索次数	189.2	224.9	231.18	237.46	247.29	190.13
UMHS 平均整数搜索次数	482.6	589.9	450.29	543.88	602.5	294.89
整数次数优化	-60.8%	-61.86%	-48.65%	-56.33%	-58.95%	-35.52%
文中算法 PSNR	35.7	33.862	35.08	33.977	32.9755	40.053
UMHS PSNR	35.6	33.861	35.07	33.976	32.9752	40.050
PSNR 优化	+0.01	+0.001	+0.01	+0.001	+0.0003	+0.003
文中算法 bitrate	146.6	241.55	979.82	724.67	2496.96	786.6
UMHS bitrate	144.0	240.19	967.91	722.23	2496.96	824.5
Bitrate 变化	+1.8%	+0.5%	+1.2%	+0.33%	-0.0%	+4.81%

从表 3 可以看出, 文中算法小幅增加码率的代

价下, 不但大幅度的减少了搜索次数, 并且无论向量

性质和尺幅,将每 MB 搜索次数稳定在 180~250 次的区间内,非常易于硬件实现设计。

表 3 文中算法与相关文献对比

文献序号	测试序列	QP/frame	DPSNR_Y /dB	DPSNR_U /dB	DPSNR_V /dB	DBITRate /%	ME_reduce
文献[3]	Mobile_cif	28 / 30	-0.01	0	-0.01	-0.056%	-11.63%
文中	Mobile_cif	28 / 30	+0.001	0	0	+0.56%	-61.86%
文献[1]	Mobile_cif	28 / 30	-0	-0	-0.01	+0.26%	-53.245%
文中	Mobile_cif	28 / 30	+0.001	-0	-0	+0.56%	-61.86%
文献[4]	Stefan_cif	28 / 30	-0	N/A	N/A	0	-43.83%
文中	Stefan_cif	28 / 30	+0.017	-0	+0.02	+1.85%	-60.08%
文献[5]	Paris_cif	28/30	-0.011	-0.012	-0.027	+0.130%	-47.46%
文中	Paris_cif	28/30	+0.01	+0	+0.2	+0.74%	-55.94%

6 硬件消耗预计

6.1 运动检测的硬件消耗估计

第 3.2 节中提出的运动检测和边缘检测基本算法不需要额外的存储器,需要帧内 SAD,MV 和 ActivatedMB 个数 3 个比较器,其他逻辑电路也比较简单.边缘检测可以共享这些资源.第 3.3 节提出的 MV 阈值自适应更新需要额外的 2 个比较器.加入 SAD 控制算法的运动检测需要增加额外 1 个比较器.SAD 阈值自适应更新需要增加一宏块行的 SAD 寄存器,2 个加法器和 3 个比较器.综上所述,加入了 MV、SAD 阈值自适应更新的算法共计需要额外 2 个加法器,1 个乘法器,6 个比较器和一宏块行的 SAD 寄存器(若一行 720 像素,则为 45×8bits).

下面以 D1 尺寸为例与传统的运动检测算法做一估计比较.如表 4 所示.

表 4 文中运动检测算法和传统算法硬件消耗比较

文献	文中	文中(*)	光流法 [3,5]	帧差法 [3]	背景差法 [3]
			720	720	720
存储器	0	45 ×8bits	×480 ×8bits ×2	×480 ×8bits ×3	×480 ×8bits ×2
加法器	1	2	6	0	0
乘法器	0	1	2(误差的平方计算)	0	0
比较器	3	6	3	1(单像素比较)	1(单像素比较)
其他逻辑电路	很简单	较简单	比较复杂	较简单	较简单

注:(*)是加入了卡尔曼滤波和 MV、SAD 阈值的自适应更新算法

从表 4 可以看出,由于传统的运动检测算法都需要配置独立的帧存储器,用以计算帧间差别,所以除了视频编码器配置的存储器之外,还需要配置比较大的存储器.而本文提出的方法将运动检测(Motion Detection)和编码器中的运动预测(Motion Estimation)整合为一体,不再需要在视频编码器之外再配置存储器,极大地节省了存储器的使用.边缘检测的硬件估计与运动检测类似.

6.2 运动检测的硬件消耗估计

文中提出的基于运动检测的搜索模式算法非常适宜于硬件实现,参考帧每一 MB 级数需要使用 2bits,同时需缓存当前帧两 MB 行的级数信息.搜索模式共计 43 点,每点 5bits.若以 D1 尺幅为例,存储器只需增加 3095Bits (仅 387Bytes),控制电路算法简单,增加逻辑电路非常少.

7 结束语

文中提出了一种新颖的运动检测和边缘检测算法.将两种算法和视频编码器的运动预测有机地结合,极大的节省了应用于两种算法存储器和电路规模.在此基础之上提出了一种视频搜索模式算法,相对传统的快速搜索算法,文中算法在保持了同等 PSNR 和码流水平的前提下,大幅减少了搜索次数,并且其结构非常易于硬件实现.

在实际使用中,由于文中提出的算法要求视频编码部分,尤其是运动预测部分,始终处于工作状态,导致电路功耗明显增大.下一阶段,将在减少能耗方面做出更大努力.同时,基于运动检测的码率控制也是下一阶段目标之一.

(下转第 114 页)

Δ 的方法,同样也可适用于其他均衡算法的优化.

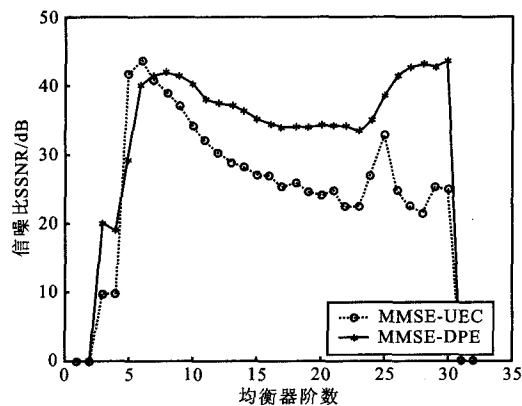


图5 不同均衡器阶数下的缩短信噪比

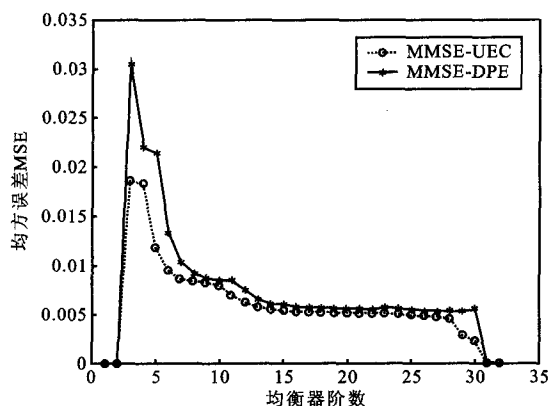


图6 不同均衡器阶数下的最小均方误差

参考文献:

- [1] 陈晓. 非对称数字用户线语音分离器设计[J]. 微电子学与计算机, 2006, 23(12): 165-167.
- [2] 相里瑜. ADSL 用户端 MODEM 中的均衡和回波消除技术研究[D]. 天津: 南开大学, 2003.
- [3] 胡峻岭, 刘文江, 王勇. 数字锁相实现 ADSL 系统中的时钟同步[J]. 微电子学与计算机, 2003, 20(12): 19-21.
- [4] Güner Arslan, Brian L Evans, Sayfe Kiaei. Equalization for discrete multitone transceivers to maximize bit rate[J]. IEEE Transactions on Signal Processing, 2001, 49(12): 3123-3135.
- [5] 陈王骞. 基于 DMT 调制研究[D]. 成都: 电子科技大学通信与信息工程学院, 2008.
- [6] 丁家昕, 何鹏, 周正. DMT 系统中一种易于实现的时域均衡器[J]. 通信学报, 2004(1): 110-116.
- [7] Guner Arslan, Biao Lu, Lloyd D Clark, et al. Iterative refinement methods for time domain equalizer design [C]// Hindawi Publishing Corporation EURASIP Journal on Applied Signal Processing. USA: Austin, 2006: 1-12.

作者简介:

卢绪成 男, (1985-), 硕士研究生. 研究方向为信号与通信系统、图象处理与可视化技术.

魏江 男, (1968-), 副教授. 研究方向为图像工程与可视化技术、信息获取与处理.

(上接第 109 页)

参考文献:

- [1] 徐培华. 简化 UMHexagonS 算法的探讨与改进[J]. 微处理机, 2008(6): 96-98.
- [2] 李海燕. H. 264 运动估计算法在 Imagine 流处理器上的加速实现[J]. 计算机科学与工程, 2006, 28(9): 47-49.
- [3] 杨育红. 快速运动估计 UMHexagonS 算法的探讨与改进[J]. 计算机工程与应用, 2006(11): 52-54.
- [4] 郑艺玲. UMHexagonS 算法的优化研究[J]. 广西工学院学报, 2009(3): 10-14.
- [5] Xiaoquan Yi. Improved and simplified fast motion estimation for JM [EB/OL]. [2010-01-15]. <http://wftp3.itu.int/av-arch/jvt-site>.
- [6] Horn B K P, Rhunck B G. Determining optical flow [J]. Artificial Intelligence, 1981, 17(1/3): 185-203.
- [7] Hu W M, Tan T N, Wang L, et al. A survey on visual surveillance of object motion and behaviors [J]. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 2004, 34(3): 334-352.
- [8] 汪烈军. 一种改进的快速块匹配运动估计算法[J]. 微电子学与计算机, 2007, 24(6): 106-109.
- [9] 孙力帆. 基于块时空相关性的 H. 264 快速帧间模式选择算法[J]. 微电子学与计算机, 2009, 26(3): 117-121.
- [10] Zhibo Chen. Fast integer pel, fractional pel motion estimation for JVT [EB/OL]. [2010-01-19]. <http://wftp3.itu.int/av-arch/jvt-site/>.

作者简介:

刘振宇 男, (1982-), 博士, 助理研究员. 研究方向为射频集成电路设计与多媒体编解码芯片设计.

周莉 女, 硕士, 副研究员. 研究方向为多媒体编解码芯片设计.

陈杰 男, 博士, 研究员, 博士生导师. 研究方向为导航芯片设计与多媒体编解码芯片设计.