

视频解码计算复杂度的线性建模理论及在线预测方法

田 婷¹⁾ 余胜生¹⁾ 郭红星¹⁾ 苏曙光²⁾

¹⁾(华中科技大学计算机科学与技术学院 武汉 430074)

²⁾(华中科技大学软件学院 武汉 430074)

摘 要 视频解码是一类最典型的多媒体应用,其计算量大、耗能高。现代多媒体计算平台可利用视频解码计算复杂度固有的动态变化特征来自适应地调整所需计算资源,从而节省能耗,其前提是对视频解码计算复杂度进行准确估计。作者基于解码计算复杂度与帧长之间的线性关系,提出了一种利用状态变量法对解码计算复杂度进行理论建模和在线估计的方法。与传统的直接对帧长和计算复杂度之间的输入-输出依赖关系进行建模所不同,这里将视频解码系统表征为由视频内容特征的状态变化所驱动的系统。首先从语义层面对解码器各模块的解码复杂度进行分析,并导出各模块计算复杂度与语义参数间的依赖关系模型,总解码复杂度为各子模块的复杂度之和。经过简化得到解码计算复杂度与帧长之间的线性模型,其中模型系数为上述语义参数的函数,表征了视频内容特征的状态变化,被定义为状态变量。再结合压缩视频流中相邻帧语义参数之间的相关性,将系统状态方程定义为反映视频内容变化程度的分段线性函数。根据 I 帧和 P 帧状态轨迹特性及其在压缩码流中位置属性的不同,分别进行计算复杂度在线估计:对于 I 帧,采用统计分析方法获得其状态变量的均值并进行在线估计;而对 P 帧,则是在运行过程中利用状态方程对状态变量进行实时更新和计算复杂度估计。在基于 SimpleScalar 的软件仿真平台和基于 DSP 的嵌入式硬件平台上分别对 H. 264、MPEG-4 压缩码流的解码计算复杂度进行在线估计,实验结果表明:对解码计算复杂度的平均估计误差在 7% 以内,预测精度非常高,而且状态方程更新过程简单,在线运行复杂度低,特别适用于嵌入式移动设备。

关键词 视频解码;计算复杂度;线性模型;在线预测;状态变量分析法
中图法分类号 TP302 **DOI 号:** 10.3724/SP.J.1016.2012.02048

Linear Modeling and Online Predicting for Video Decoding Complexity

TIAN Ting¹⁾ YU Sheng-Sheng¹⁾ GUO Hong-Xing¹⁾ SU Shu-Guang²⁾

¹⁾(School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

²⁾(School of Software Engineering, Huazhong University of Science and Technology, Wuhan 430074)

Abstract Multimedia applications, such as video decoding, always take heavy computational complexity and cause huge energy consumption. To save energy, existing hardware platforms tend to adjust their resources according to the actual decoding computational demand. The effectiveness of such methods depends on the prediction accuracy of decoding complexity. Based on the linear relationship between frame length and the decoding complexity of each frame, this paper proposes a linear modeling and online predicting method which utilize State-Variable Analysis (SVA) for video decoding complexity prediction. The state equation of decoding system is established as follows. The semantic meanings of state variables which directly affect the linear relationship between frame length and decoding complexity are exploited through analysis on decoding procedure. Then combined with the semantic correlation between neighboring frames in

收稿日期:2012-06-30;最终修改稿收到日期:2012-08-13。本课题得到国家自然科学基金“嵌入式多媒体流计算的质量驱动机制与共生调优”(61173044)、国家自然科学基金“嵌入式多媒体流计算的自适应机制与跨层优化”(60873029)、中央高校基本科研业务费专项资金资助。田 婷,女,1985 年生,博士研究生,主要研究方向为嵌入式计算、流媒体技术。E-mail: tianting_hust@gmail.com。余胜生,男,1944 年生,教授、博士生导师,主要研究领域为计算机存储系统、多媒体技术。郭红星(通信作者),男,1971 年生,博士,副教授,主要研究方向为嵌入式计算、多媒体网络。E-mail: guohx@mail.hust.edu.cn。苏曙光,男,1975 年生,博士,讲师,主要研究方向为视频编码与通信、嵌入式系统与中间件。

compressed streams, the state equation is established as a piecewise function which can reflect the variation of video content. Considering the different characteristics of state trajectory between intra-frame and inter-frame, we provide online decoding complexity predicting methods for them respectively. The mean values of state variables for intra-frame are obtained through offline statistical analysis and then utilized directly during online process to predict decoding complexity. For inter-frame, the values of state variables are calculated using state equation in the online process and then the decoding complexity can be predicted. The proposed method is implemented on both software-based simulation platform and DSP-based hardware platform, the decoding complexity of compressed streams which use either H. 264 or MPEG-4 coding standards is predicted. The experimental results show the proposed method can give fairly accurate prediction for decoding complexity. The average prediction errors are less than 7%. Moreover, since the state equation can update the state variable at a very low cost, the runtime overhead of the method is very small. It is very useful for those resource-limited mobile devices.

Keywords video decoding; computational complexity; linear model; online prediction; state-variable analysis

1 引言

视频编解码一类的多媒体应用不仅计算复杂度高,而且计算过程需要进行密集数据存取,导致其功耗需求较一般应用程序要大得多.因此,如何构建高性能、低功耗的多媒体系统一直是设计者长期追求的目标,特别是对于目前方兴未艾的无线移动多媒体应用,由于其硬件平台资源严重受限,且采用电池供电,能量有限,更迫切需要采用节能技术来提高续航能力.这已经成为衡量众多移动设备性能的最重要指标之一.

通过动态调节技术能使系统在运行阶段根据工作负载的动态变化来调整工作模式,从而降低能耗^[1].现代计算机硬件的可伸缩性也为动态调节技术提供了支持.按照调节方式的不同可将动态调节技术分为两类:动态功耗管理(DPM)技术^[2]和动态电压频率调节(DVFS)技术^[3].DPM根据系统工作负载的变化情况有选择地将设备设置为低功耗模式(或直接关闭),从而降低系统能耗.DVFS则通过动态预测后续任务所需计算量,并根据预测结果调节处理器运行电压和频率,以达到任务响应时间和系统功耗之间的平衡.DPM技术适用于间歇性任务,如硬盘读写操作;对于实时性要求高、任务时间间隔短的多媒体流计算任务,频繁的设备状态切换不仅增大了系统开销,而且还导致设备处于不稳定状态,在这种情况下,DVFS技术更为适用.

对多媒体流的计算复杂度进行精确估计是提高DVFS效率的前提.最坏路径法(单位计算单元运行最大开销)和平均值法(单位计算单元运行平均开销)是对系统计算量进行估计的常用方法.由于多媒体系统计算复杂度受视频压缩标准、视频帧的编码类型、采用的编码工具集合、视频内容、幅面、压缩码率等因素的影响,变化幅度非常大,传统的计算复杂度估计方法不适用于时变性非常强的多媒体任务,迫切需要发展针对多媒体流计算任务的精确计算复杂度估计方法.

视频解码是目前应用最为广泛的一类多媒体应用.本文将以视频解码计算复杂度为研究对象,探讨对其进行分析和建模的方法;同时通过该分析和建模过程归纳出对多媒体系统进行分析的共性方法并指导后续工作.

2 相关研究

本节将在分析已有视频解码复杂度估计方法的基础上,简要说明本文的工作及其意义.

2.1 视频解码复杂度估计研究进展

典型视频解码复杂度估计方法可分为两类:基于统计分析的方法和基于预测的方法.基于统计分析的方法首先将解码过程分成几个相对独立的模块(如变长解码、运动补偿等),然后对每个模块进行统计分析并确定各模块所需计算量与各种参数之间的依赖关系,总解码计算量为各模块计算量之和.这类

方法的典型代表包括新加坡国立大学 Wang 等人^[4]提出的直接将每个模块的计算复杂度分别建模为残差系数个数、不同编码类型宏块的数目等编码参数的线性模型. 纽约大学 Wang 等人^[5]提出的对每个模块的基本操作单元进行分析, 并将模块复杂度建模为基本操作单元的个数与操作单元平均计算复杂度之积. 加州大学洛杉矶分校 Schaar 等人^[6]提出的利用混合高斯模型建立视频特征参数以及解码执行时间的联合概率密度分布函数, 并通过离线学习与在线更新相结合的方法对上述概率密度函数进行动态更新, 从而适应视频内容的变化. 南加州大学 Kuo 等人^[7]提出的将宏块运动补偿(帧内预测)的计算复杂度建模为 Cache 未命中次数、横向插值次数、纵向插值次数和宏块内运动矢量个数这 4 个参数的线性表达, 并通过对多个测试序列进行统计分析, 获得每个参数的权值因子. 上述方法能够对解码计算复杂度进行非常精确的分析, 估计结果准确. 但是, 为了获得解码复杂度模型中的各种参数, 必须在解码端插入大量的分析语句, 或者在传输码流中插入指示复杂度的信息. 这样不仅增加了数据传输量, 还需要对解码系统进行修改, 系统运行时的复杂度增加, 也不利于模块化系统设计.

基于预测的方法是根据视频帧之间的相关性, 利用已解码帧的计算复杂度来估计后续帧的计算复杂度. 这类方法的重点和难点是如何挖掘出相邻帧计算复杂度之间的相关性. 一旦建立起这种相关性就可以通过简单的预测来获取每一帧的计算量, 在线运行时开销小, 适用于处理能力较弱的多媒体计算平台. 典型算法包括伊利诺伊大学香槟分校 Nahrstedt 等人^[8]提出的利用相邻时间段内的视频帧解码计算复杂度的概率密度函数具有连续性和相似性的特点, 以组为单位对解码计算复杂度概率密度函数进行动态更新, 并将前面组的概率密度函数近似作为待解码视频组的概率密度函数. 之后利用解码系统的软实时性特点, 在满足解码率 ρ (即在截止期之前被解码的视频帧占总视频帧的比率) 的条件下, 在概率密度函数中寻找合适的计算复杂度 C , 使得 $P(c \leq C) \geq \rho$, 将 C 作为对待解码视频组中每一帧的计算复杂度的预测. 该算法虽然利用了解码计算复杂度概率密度函数的相关性, 但对视频帧计算复杂度的估计是以组为单位进行, 粒度较大, 没有充分挖掘出相邻视频帧之间的相关性, 因此估计结果并不精确.

我们提出了一种融合离线统计分析和在线预测

的解码计算复杂度估计方法^[9], 利用解码计算复杂度与视频帧长之间的线性关系对其进行建模. 在离线阶段通过统计方法对模型系数变化规律进行分析并对相邻帧模型系数之间的关系式进行求解; 在线运行时利用这一关系式动态更新模型系数并进行解码计算复杂度估计. 这一方法融合了统计分析和在线预测两种手段, 实验结果表明该算法能以非常小的系统开销对解码计算复杂度进行精确估计. 这部分工作很好地利用了相邻帧模型系数的相关性, 但对于相关性的内在机理分析不够, 建模和分析方法不足以推广至其它多媒体应用程序.

2.2 本文的研究工作及其意义

本文在已有基础上主要开展了如下工作:

(1) 根据多媒体系统输出与视频内容特征之间的强依赖性以及相邻视频帧之间潜在的相关性, 首次引入信号系统分析理论, 将多媒体解码系统定义为状态驱动的系统. 将系统输出响应(解码计算复杂度)的求解过程转化为描述系统状态变量变迁轨迹(即状态方程)的过程.

(2) 采用状态变量分析法对解码系统计算复杂度进行求解, 并利用状态变量简化法优化系统状态空间. 解码计算复杂度最终表征为以帧长作为输入, 系数 (K, B) 为状态变量的输出响应. 状态变量 K 用以表征压缩视频编码信息的变化(视频编码信息主要指预测模式、运动矢量精度及运动矢量范围); 状态变量 B 表征压缩视频幅面的变化, 当视频幅面相同时, B 用以表征被编码残差的变化.

(3) 以 (K, B) 为状态变量利用压缩视频帧被编码残差和视频编码信息之间固有的量变与质变关系(即相邻帧之间的变化首先通过残差表现; 当变化过大时, 相邻帧的编码信息将发生明显变化), 给出了一种建立系统状态方程的方法, 并将其最终定义为表征视频内容变化程度的分段线性函数.

(4) 提出了对解码计算复杂度在线估计的一整套完整解决方案, 实现了对不同编码方式、不同幅面、不同帧类型的视频帧解码计算复杂度在线估计, 并通过软仿真和硬仿真两种方式分别对 H. 264 和 MPEG-4 压缩码流进行测试. 实验结果表明本方法能很好地适应各种类型的压缩码流.

除上述对解码计算复杂度进行理论分析、建模和动态估计等工作外, 本文所提出的方法从另外两个方面展现了对其它多媒体系统建模的意义, 包括:

(1) 本文所提出的方法从本质上来讲是回归了解码系统的物理属性, 从解码机理出发探讨影响解

码系统计算复杂度的因素. 多媒体系统不同于其它计算系统的重要特点是其具有丰富的语义层信息, 虽然在计算机中多媒体是以信号的形式被表示、存储和传输的, 但语义层信息却广泛地被用来对多媒体信号处理过程进行优化. 遗憾的是, 多媒体信号经过越复杂的处理, 其语义层信息被利用的程度就越低, 以至于最后通常选择复杂数学模型对视频信号进行分析和处理, 语义层信息熵被人为增大. 本文所提出的解码计算复杂度估计方法是一种寻找编解码系统中语义层信息传递的方法, 即原始视频帧时域和空域的相关性是如何转化为压缩视频帧之间的相关性. 首先利用基于输入-输出的统计分析法获得可能体现压缩域视频帧相关性的参数, 然后利用状态变量法从理论层面进行推导, 确定待选参数作为状态变量的语义层含义. 最终落脚点是从语义层分析这些包含物理含义的状态变量随视频帧连续变化而动态变迁的过程, 并将其转化为数学形式进行求解, 而不是单纯的依靠数学工具. 这种方法能够挖掘不同形式视频信号中所潜在的语义层信息, 保持语义层信息熵的稳定性, 同时也为提高多媒体系统分析效率和减少分析复杂度提供了新思路.

(2) 本文所获得的状态方程虽然最终服务于解码计算复杂度估计, 但是由于它能够很好地表征压缩视频内容、残差以及视频幅面的变化, 故可以作为独立系统服务于其它需要对视频运动复杂度以及纹理复杂度等视频特性进行分析的多媒体应用中.

本文第 3 节通过实验对解码计算复杂度和压缩视频帧长呈正相关这一现象进行描述, 并采用统计的方法验证将帧长与解码计算复杂度之间的正相关性建模为线性关系的合理性; 第 4 节阐述利用状态变量分析法获得解码系统状态变量(K, B)物理含义的原理以及系统状态方程的建立过程, 并给出了利用状态方程分别对 I 帧和 P 帧解码计算复杂度进行动态估计的方法; 第 5 节是本文的实验部分, 将分别对 H. 264 和 MPEG-4 压缩码流进行在线解码复杂度动态预测, 本节还将详细展现在线预测过程中表征视频特征的状态变量的变迁过程; 最后, 第 6 节是结论.

3 计算复杂度与帧长的线性模型

通过对运行在典型嵌入式平台上的解码任务计算量与视频流帧长之间的关系进行实验分析, 发现

视频帧的解码计算复杂度与帧长正相关. 以基于 TI TMS360DM642 DSP 芯片的嵌入式解码平台为例, 采用遵循 MPEG-4 编码标准的 XVID 编码器将 CIF 格式的测试序列 Akiyo、Hall、Coastguard、Crew、Foreman、Mobile 在各种码率下 (200 kbps、400 kbps、500 kbps、600 kbps、800 kbps) 进行压缩, 将生成的压缩码流输入到上述嵌入式解码器中进行解码, 并记录每帧压缩视频的帧长及所需计算复杂度. 编码参数为 XVID 编码器的默认参数配置, 将最大 I 帧间隔设置为 32. 采用的解码器是遵循 MPEG-4 编码标准的 DIVX 解码器^①, 利用 TI 公司提供的线性汇编语言对解码器进行优化, 没有任何算法上的修改, 所得结果不失一般性.

图 1 给出了码率为 500 kbps 时, Foreman 和 Mobile 序列每一帧的帧长 (以字节为单位) 和解码计算量 (以解码一帧所需要的时钟周期数 cycle 作为度量) 对比图. 其它码率下, 各测试序列的表现形式与此类似.

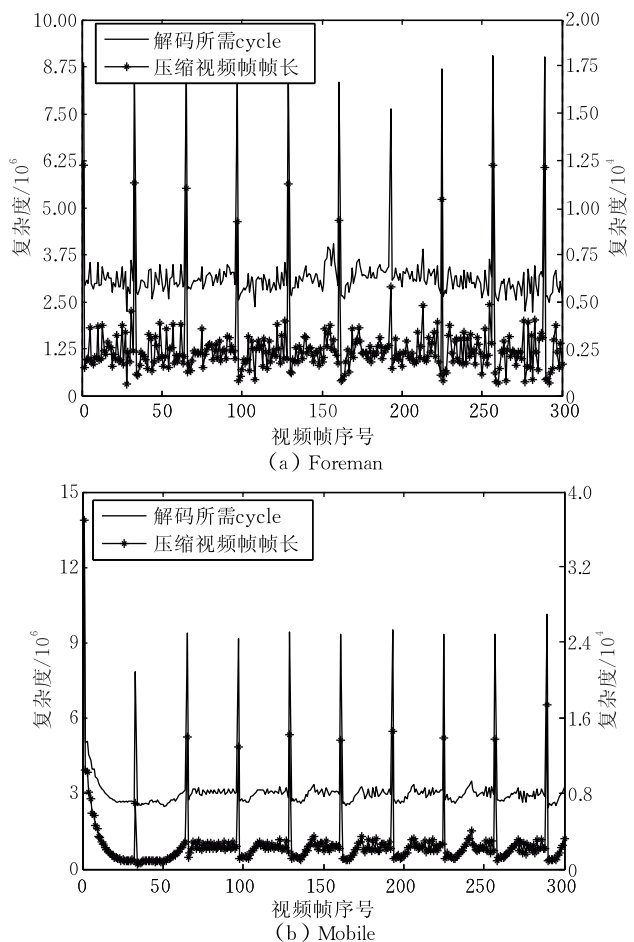


图 1 500 kbps 测试序列每帧帧长与解码计算复杂度对比图

① http://en.pudn.com/downloads270/sourcecode/multimedia/etai1232774_en.htm

从图 1 中可以看出,解码计算复杂度与帧长的走势一致,当帧长发生剧烈变化时,计算复杂度也发生明显变化.因此,可将解码计算复杂度(记为 C)与帧长(记为 L)之间的关系建模为

$$C = f(L) \quad (1)$$

其中关系函数 f 应体现帧长与计算复杂度之间的正相关性.考虑到算法实现时的低复杂度要求,尝试将 f 定义为线性函数,即

$$C(L) = K \times L + B \quad (2)$$

式(2)成立的充要条件是随机变量 C 和 L 线性相关,即两者相关系数 ρ 等于 1. 以 CIF 格式的测试序列 Akiyo、Hall、Coastguard、Crew、Foreman、Mobile 在各种码率下(200 kbps、400 kbps、500 kbps、600 kbps、800 kbps)采用 XVID 编码器编码并解码后所得到的计算复杂度和帧长为样本,计算样本相关系数,结果如图 2 所示.

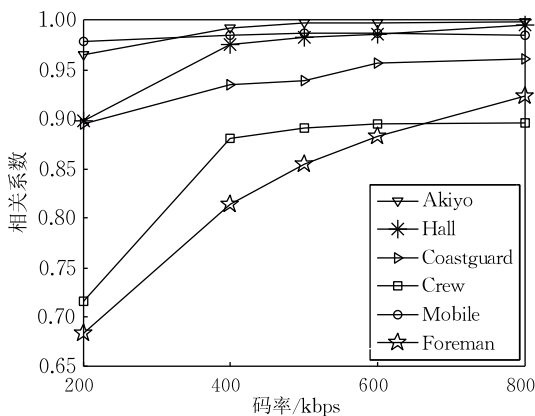


图 2 各序列在不同码率时解码计算复杂度与帧长相关系数

在大多数情况下解码计算复杂度与帧长之间的相关系数大于 0.8,表明两者确实存在很强的线性相关性;视频运动复杂度低(如 Akiyo、Hall 等序列),两者相关系数大于 0.95,可以认为解码计算复杂度与帧长是线性相关的.因此将 f 定义为线性函数是合理的.

式(2)中函数系数(K, B)可通过文献[9]所给出的线性拟合方法离线分析获得,在线运行过程中便可利用该拟合系数对解码计算复杂度进行估计.但是由于系数(K, B)是随输入视频内容的变化而动态变化的(图(2)中计算复杂度和帧长相关系数不等于 1 可以说明这一点),静态系数无法反映视频内容特征的变化.因此有必要挖掘系数(K, B)随视频内容特征动态变化的规律,从而指导我们对解码计算复杂度进行在线动态估计.

4 融合统计分析和状态变量分析法的解码复杂度在线动态预测

为了挖掘系数变化规律,需要从多媒体系统内部机理进行分析.在信号系统分析法中,状态变量分析法正是从系统内部机理进行分析,构建系统各节点的状态变量与系统输入输出之间联系的方法.采用状态变量分析法需要首先确定系统状态变量,然后建立两个基本方程,即

输出方程: $\mathbf{Y}(n) = \mathbf{C}\mathbf{X}(n) + \mathbf{D}\mathbf{e}(n)$,建立输出向量 \mathbf{Y} 和状态向量 \mathbf{X} 以及输入向量 \mathbf{e} 之间的联系.

状态方程: $\mathbf{X}(n+1) = \mathbf{A}\mathbf{X}(n) + \mathbf{B}\mathbf{e}(n)$,建立状态向量 \mathbf{X} 在输入 \mathbf{e} 的作用下的变化.

考虑到视频具有天然的语义层信息,视频编解码器运行的过程就是挖掘和解释视频运动特点、纹理特点的过程,因此系统的输出是由视频特征所驱动的;而且由于视频帧具有时域上的连续性,其特征也具有时域上的相关性.因此可借鉴状态变量分析法的思想,将帧长与计算量线性模型中的模型系数(K, B)重新定义为表征视频内容特征的状态变量,并通过分析状态变量的具体物理含义来建立系统状态方程.

4.1 视频解码系统状态方程的建立

4.1.1 解码系统状态变量物理含义分析

根据第 3 节的分析,系统输出方程可以定义为 $C(n) = K(n) \times L(n) + B(n)$, n 为视频帧序号.每一帧的帧长在压缩后已经确定,可以认为 $L(n)$ 为常量,解码系统状态变量为(K, B).本小节将从视频编解码器工作流程切入,对编解码器计算复杂度各组成部分进行分析,并通过状态变量化简,推导出以(K, B)为状态变量的系统输出.

图 3 给出了一个典型的视频编解码器的工作流程.压缩码流输入到编解码器,熵解码模块(ED)对输入的码字进行解析,得到压缩域残差系数.重排序模块(Reorder)对这些残差系数进行反 zig-zag 扫描,获得按行列顺序排列的压缩域残差系数.反量化模块和反变换模块(IQ&IT)则将上述重排序后的压缩域系数进行反量化和反变换操作获得像素域的残差信息.最后 MC/IP 模块利用块预测模式和运动矢量,取得当前块的预测块,并将其与像素域残差信息相加,得到最终重构后的块.因此,视频编解码计算复杂度 C 可表征为熵解码(C_{ED})、重排序(C_{ReO})、反量化和反变换($C_{IQ\&IT}$)以及 MC/IP 模块($C_{MC/IP}$)的计

算复杂度之和,即

$$C = C_{ED} + C_{ReO} + C_{IQ\&IT} + C_{MC/IP} \quad (3)$$

换言之,4个模块的计算量作为系统节点处的输出,可被定义为解码系统状态变量(如图3中虚线

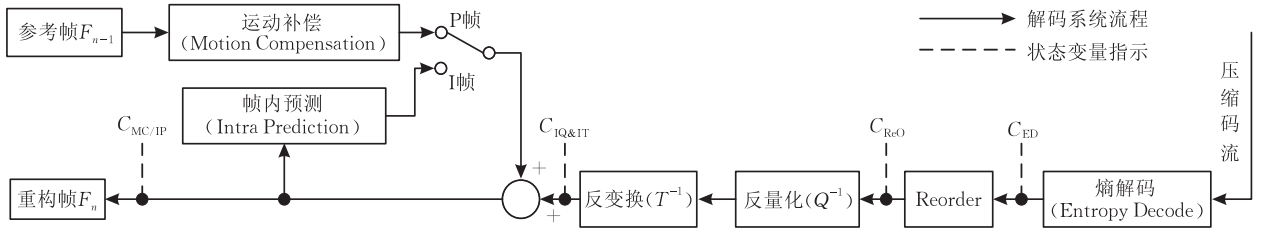


图3 视频解码器工作流程图

熵解码模块按照码流输入顺序对码字逐个进行处理,其解码复杂度与视频帧帧长呈正比关系^[5]. 设 L 为码流长度, C_0 为解码每个码元所需要的平均计算量,熵解码计算复杂度可以定义为

$$C_{ED} = C_0 \times L \quad (4)$$

Reorder 模块对固定大小的块(4×4 或 8×8) 进行反 zig-zag 扫描,由于反 zig-zag 扫描的步骤不变,因此消耗在每个块的计算复杂度基本不变. Reorder 模块的计算量由数据块个数决定. 在视频编码中,对于采用统一编码标准的视频帧,其数据块大小固定,块数目由视频帧的幅面决定. 因此 Reorder 模块的计算量与视频帧帧长基本无关,主要取决于视频帧幅面. 设视频所包含的分块数为 N , 每个固定大小块所需要的计算复杂度为 C_1 , 则

$$C_{ReO} = C_1 \times N \quad (5)$$

IQ&IT 模块对每个固定大小的压缩域系数块进行相同的加法和乘法操作,与 Reorder 模块类似,其计算量与视频帧帧长基本无关,主要由视频帧幅面决定. 但是目前所采用的优化 IDCT 算法会利用许多 DCT 系数为零这个特点对计算过程进行简化,在这种情况下反变换的计算量与非零系数的个数和出现位置有关. 设视频所包含分块数为 N , 处理每个块所需平均计算复杂度为 C_2 , 则

$$C_{IQ\&IT} = C_2 \times N \quad (6)$$

MC/IP 模块的计算复杂度包含两部分: 获取参考块所需计算量(C_{fetch})以及将参考块与残差相加所需计算量(C_{calc}). 取参考块的计算复杂度取决于块类型、块预测模式以及运动矢量的精度,当预测模式复杂、采用分数精度运动矢量时计算复杂度会急剧增加. 由于块类型、块预测模式及运动矢量都是宏块头的组成部分,复杂度高的预测模式以及分数精度运动矢量将导致宏块头消耗较多码字,因此可以认

处指示):

$$\mathbf{X} = (C_{ED}, C_{ReO}, C_{IQ\&IT}, C_{MC/IP}).$$

为了获得系统状态方程,对每个状态变量进行独立分析,获得其各自表达式.

为 C_{fetch} 与宏块头长度正相关,从视频帧的角度来说就是与帧头长度正相关. 将残差与参考块相加所需计算复杂度对单个块来说是一样的,总计算量取决于视频幅面. 设视频帧头部信息长度为 L_h 且帧头中每个码元预取数据所需平均计算复杂度为 C_3 , 视频所包含的分块数为 N , 每个块相加所需平均计算复杂度为 C_4 , MC/IP 计算复杂度可定义为

$$C_{MC/IP} = C_{fetch} + C_{calc} = C_3 \times L_h + C_4 \times N \quad (7)$$

综合式(3)至式(7),解码计算复杂度可以表征为 $C = C_0 \times L + C_1 \times N + C_2 \times N + C_3 \times L_h + C_4 \times N = C_0 \times L + C_3 \times L_h + (C_1 + C_2 + C_4) \times N$.

将上式与 $C(L) = K \times L + B$ 对比,可令

$$\begin{cases} K = C_0 + C_3 \times \frac{L_h}{L} \\ B = (C_1 + C_2 + C_4) \times N \end{cases} \quad (8)$$

考察状态变量 K 的组成部分, C_0 为熵解码模块解码每个码元所需要的平均计算量,实验表明该系数的变化幅度不大^[5]. K 值的大小主要取决于 $C_3 \times (L_h/L)$ 的大小,其中 C_3 是 MC/IP 模块中每个码元预取数据所需平均计算复杂度,当视频采用复杂的预测模式、运动矢量为分数精度、运动矢量范围很大时 C_3 的值较大. L_h/L 则表征的是帧头占帧长的比例,该值由率失真优化模型决定,总是试图在头信息和残差信息所占长度之间取得最好的平衡. 因此其大小与码率分配策略以及视频内容密切相关,当码率分配均匀时,如果视频内容变化较小,相邻帧的帧头在帧长所占比例应基本保持不变,否则此部分的值将由于视频的预测模式、运动矢量等编码信息的变化而发生较大幅度的变化. 因此 K 值的大小由视频编码信息所决定的.

考察状态变量 B 的组成部分, C_1 、 C_2 、 C_4 分别为 Reorder、IQ&IT 以及 MC/IP 模块中对固定大小块

所做操作的平均计算量,原则上讲是不变的,因此 B 值主要取决于 N 的大小,即视频幅面.但是当压缩块残差系数全部为零时,解码器将不对该块进行相应的操作;当为零的系数很多时,IQ/IT 模块也将采取一定的快速算法,单个块的计算复杂度也将发生变化,因此在这种情况下,残差的变化决定了 B 值的变化.但是必须指出的是,在给定正常码率的条件下,不可能出现所有块系数为零的情况,决定 B 值的主因仍然是视频幅面.

为了验证上述对状态变量(K, B)物理含义的分析,将 CIF 和 4CIF 格式标准测试序列在不同码率下进行压缩并解码,1 帧间隔设为 250,对获得的帧长和计算量进行拟合,得到不同码率时各测试序列的状态变量拟合值,实验结果如表 1 和表 2 所示.

表 1 不同码率下 CIF 格式测试序列状态变量拟合值

视频序列	不同码率(cycles/byte)下的拟合值				
	K_200	K_400	K_500	K_600	K_800
Akiyo	355.15	285.19	275.32	269.86	264.82
Hall	453.87	356.77	325.44	322.78	270.13
Coastguard	367.54	215.57	210.64	209.42	208.32
Crew	762.48	710.70	624.43	559.20	497.98
Foreman	589.93	363.96	327.16	315.28	288.38
Mobile	264.06	263.23	266.16	265.79	265.13

视频序列	不同码率(10^6 cycles)下的拟合值				
	B_200	B_400	B_500	B_600	B_800
Akiyo	1.7594	1.8181	1.8369	1.8458	1.8555
Hall	1.9003	2.0765	2.1648	2.1739	2.3725
Coastguard	2.2687	2.2916	2.3194	2.3408	2.4035
Crew	2.6730	2.0473	2.0626	2.0958	2.0764
Foreman	2.1015	2.2660	2.3136	2.3340	2.4009
Mobile	2.4263	2.4055	2.4079	2.4184	2.4289

表 2 不同码率下 4CIF 格式测试序列状态变量拟合值

视频序列	不同码率(cycles/byte)下的拟合值				
	K_800	K_1500	K_2000	K_2500	K_3000
City	358.30	313.53	295.63	278.26	274.56
Crew	829.68	808.13	677.74	606.98	533.26
Harbor	351.88	373.05	341.71	345.92	305.79
Ice	427.67	332.20	292.83	283.46	268.62

视频序列	不同码率(10^6 cycles)下的拟合值				
	B_800	B_1500	B_2000	B_2500	B_3000
City	9.1524	9.3952	9.5363	9.6922	9.7274
Crew	11.3620	7.8848	8.0159	8.0077	8.3511
Harbor	9.4620	9.4887	9.7558	9.6878	10.1810
Ice	8.3821	8.4888	8.7077	8.7200	8.8715

观察上述两表,可以发现:

(1)随着视频序列运动复杂度增加, K 值在不断增大,Crew 序列的运动复杂度最大,其 K 值明显大于其它序列.

(2)4CIF 格式视频序列 B 的拟合值大约为 CIF 格式视频序列拟合值的 4 倍,与两种视频幅面的比值相等.

(3)当视频序列幅面相同时,随着被编码残差数量的增加 B 值将增大.如 Mobile 序列,其纹理非常复杂,导致残差量较大;Foreman 序列,其视频运动复杂度较高,残差量较大;对于 Crew 序列,由于其运动复杂度很高,在编码时用以表示宏块编码模式、运动矢量等编码信息的码字较多,最终被编码的残差数据较少,实验表明,Crew 序列被编码残差量明显小于其它序列,在 500 kbps 时,其残差所占长度仅为帧长的 53%,同等测试条件下 Mobile 序列这一比例为 76%.

(4)随着码率增加,被编码残差数量增加, B 值呈增大的趋势.

上述实验结果表明 K 确实主要受视频预测模式、运动矢量的范围和精度的影响,而 B 基本与上述视频编码信息无关.在幅面相同条件下,被编码残差数对 B 的影响较大;当视频幅面不同时, B 的变化非常大,此时 K 的变化并不明显.因此, K 和 B 的物理含义可以定义为:

状态变量 K 可以表征被编码视频编码信息变化,其中视频编码信息主要指预测模式、运动矢量精度及运动矢量范围.

状态变量 B 可以表征被编码视频幅面的变化,当视频幅面相同时, B 可以表征被编码残差的变化.

4.1.2 解码系统状态方程的建立

状态方程用来表征系统状态变量在输入激励下所产生的变化,在视频解码系统中就是建立相邻帧状态变量(K, B)之间的联系.通过对 K 和 B 物理含义的分析,可以将相邻帧状态变量的变化映射为编码信息以及被编码残差的变化.

根据视频压缩原理,从相邻帧变化角度来说,编码信息和被编码残差之间是质与量的关系,即相邻帧之间的变化首先通过被编码残差的数量发生改变而体现;当视频内容发生剧烈变化时,被编码残差数量发生很大幅度变化,此时编码器率失真优化模块和码率控制模块将通过改变编码信息来提高压缩视频的率失真性能.因此 K 和 B 之间的关系也可以认为是质与量的关系,以相邻帧视频内容的变化程度作为衡量质变与量变的准则,则有

(1)相邻帧视频内容变化小,视频内容的差别主要由被编码残差的数量的变化来体现.相邻帧 K 值保持不变, B 值发生变化.

(2)相邻帧视频内容变化大,视频内容的差别主要由视频编码信息的变化来体现.相邻帧 K 值变化, B 值保持不变.

采用数学形式对上述过程进行描述,则可以给出一种以视频内容变化程度为条件的分段函数,对状态方程进行求解. 设前一帧(第 $i-1$ 帧)的状态变量为 (K_{i-1}, B_{i-1}) , 通过该状态变量值预测得到当前帧(第 i 帧)的解码计算量 C_i ; 解码后得到当前帧的真实计算量为 $C_{R(i)}$, 第 i 帧的状态变量为 (K_i, B_i) , 则当前帧的解码复杂度估计误差为 $C_{R(i)} - C_i$, 且

$$\begin{cases} C_i = K_{i-1} \times L_i + B_{i-1} \\ C_{R(i)} = K_i \times L_i + B_i \end{cases}$$

当视频内容变化较小时,相邻帧状态变量应基本相同,则当前帧的解码复杂度估计误差较小;而当视频内容变化大时,相邻帧状态变量差别较大,当前帧的解码复杂度估计误差较大. 因此可用当前帧的解码复杂度估计误差来表征视频内容变化程度,进而将其作为选择状态方程的约束条件.

利用计算复杂度估计误差、视频内容变化以及状态变量变化三者之间的映射关系,解码系统状态方程可由式(9)、(10)给出.

状态方程 1: 当前帧解码复杂度估计误差在一定范围内,即 $\Delta_1 \leq C_{R(i)} - C_i \leq \Delta_2$, 则

$$\begin{cases} K_i = K_{i-1} \\ B_i = C_{R(i)} - C_i + B_{i-1} \end{cases} \quad (9)$$

状态方程 2: 当前帧解码复杂度估计误差超过一定范围,即 $C_{R(i)} - C_i \leq \Delta_1$ 或 $C_{R(i)} - C_i \geq \Delta_2$, 则

$$\begin{cases} K_i = \frac{C_{R(i)} - C_i}{L_i} + K_{i-1} \\ B_i = B_{i-1} \end{cases} \quad (10)$$

4.2 解码复杂度在线动态预测

利用状态方程对解码计算复杂度进行在线估计. 由于状态方程描述系统前后帧状态变量的变化关系,而在压缩序列中除全 I 帧序列外, I 帧一般独立出现,其后续帧多为 P 帧和 B 帧,前后帧属性不同,相关性无法通过状态方程体现. 因此本文针对 I 帧和 P 帧状态变量轨迹特性和在编码过程中的位置属性不同,分别提出了两种预测方法. 针对 I 帧主要是采用统计的方法获得 I 帧状态变量的统计特性,并利用其对解码复杂度进行预测;对于 P 帧则是融合统计分析和状态方程进行动态预测. B 帧从编码原理上来说同样是采用帧间预测编码,其估计过程与 P 帧类似,在此不单独讨论.

为了分析不同编码方式对视频状态变量的统计特性以及相邻帧状态变量的变化趋势的影响,还将对 MPEG-4 和 H. 264 编码标准进行对比讨论.

4.2.1 基于统计分析的 I 帧计算复杂度预测

表 3 和表 4 分别给出了 CIF 格式全 I 帧 MPEG-4 压缩码流和 H. 264 压缩码流在不同码率时拟合得到的 (K, B) 值. 可以发现对于 MPEG-4 压缩码流,当视频内容、压缩码率发生变化时,状态变量 (K, B) 值基本保持不变. 而对于 H. 264 压缩码流,当视频内容、压缩码率发生变化时, K 值发生明显变化, B 值有一定区别,但变化幅度并不明显.

表 3 不同码率下 CIF 格式全 I 帧 MPEG-4 压缩码流状态变量拟合值

视频序列	不同码率(cycles/byte)下的拟合值				
	K_800	K_1000	K_1500	K_2000	K_2500
Akiyo	218.59	219.32	207.46	203.20	201.98
Hall	214.38	212.58	208.95	206.26	214.91
Coastguard	218.69	215.57	210.64	208.32	206.90
Crew	238.58	233.24	222.33	215.20	212.91
Foreman	210.36	211.21	216.98	221.38	218.79
Mobile	206.68	206.63	206.65	206.28	203.78

视频序列	不同码率(10^6 cycles)下的拟合值				
	B_800	B_1000	B_1500	B_2000	B_2500
Akiyo	6.3157	6.3523	6.4301	6.5007	6.5210
Hall	6.4069	6.4261	6.4642	6.4931	6.3916
Coastguard	6.4146	6.4408	6.4961	6.5310	6.5552
Crew	6.3531	6.3799	6.4448	6.5010	6.5356
Foreman	6.4098	6.4114	6.3677	6.3004	6.2971
Mobile	6.4826	6.4834	6.4829	6.4991	6.5545

表 4 不同码率下 CIF 格式全 I 帧 H. 264 压缩码流状态变量拟合值

视频序列	不同码率(cycles/byte)下的拟合值				
	K_500	K_800	K_1000	K_1500	K_2000
Akiyo	318.17	274.37	248.67	261.17	103.58
Hall	374.25	379.10	333.37	275.49	146.82
Coastguard	238.33	228.36	221.96	203.55	191.01
Crew	227.73	244.44	265.60	259.40	176.53
Foreman	312.93	326.30	310.34	286.84	241.33
Mobile	185.62	178.90	182.28	177.06	156.58

视频序列	不同码率(10^7 cycles)下的拟合值				
	B_500	B_800	B_1000	B_1500	B_2000
Akiyo	1.4223	1.4358	1.4358	1.4332	1.5888
Hall	1.3848	1.3811	1.3997	1.4554	1.6002
Coastguard	1.4130	1.4200	1.4213	1.4440	1.4720
Crew	1.4896	1.4830	1.4704	1.4743	1.5579
Foreman	1.3812	1.3812	1.3987	1.4238	1.4670
Mobile	1.5523	1.5525	1.5374	1.5296	1.5790

由于状态变量 K 表征被编码视频运动信息的变化, I 帧只采用帧内预测. 在 MPEG-4 视频编码标准中只有两种方向的帧内预测, 且计算复杂度基本相同, 状态变量 K 值将保持基本不变. 对于 H. 264 编码标准, 由于其采用 9 种 4×4 预测模式和 4 种 16×16 预测模式, 且每种模式计算复杂度区别较大, 同时每种预测模式所需码长也不同, 因此随着视频纹理的变化 K 值将发生明显变化.

对于状态变量 B , 当视频幅面相同时, B 的值与

被编码残差数量变化相关, MPEG-4 压缩码流中各种帧内预测模式消耗的码字相同, 在同样码率的条件下被编码残差数量相近, B 值保持不变. 在不同码率条件下, 随着码率的增加, B 值会增加, 但相对于 B 值的大小来说增幅较小. 而对于 H. 264 压缩序列, 虽然各种预测模式消耗码字不等, 但由于所占码字较少, 残差量变化并不明显, 因此 B 值有一定变化, 但其主导因素还是视频幅面.

通过以上分析可以发现, 虽然 MPEG-4 压缩码流和 H. 264 压缩码流的状态变量统计特性随编码方式的不同而发生改变, 但物理含义始终保持不变. 对于在码流中单独出现的 I 帧而言, 由于它无法利用连续帧之间的相关性进行状态变量的更新, 这种物理含义的一致性就显得尤为重要. 通过分析状态变量所表征的压缩码流特征, 了解影响状态变量变化的主要因素, 为利用 I 帧状态变量的统计特征值提供了理论支持. 更确切的说, 就是将状态变量 (K, B) 的物理含义与表 3、表 4 中状态变量的统计特性相结合, 可得到:

规律 1. 当视频幅面一定时, 随着视频内容、压缩码率的变化, 状态变量 (K, B) 值基本保持不变.

对于不同幅面的视频, 表 5 给出了 4CIF 格式全 I 帧 MPEG-4 压缩码流在不同码率时拟合得到的 (K, B) 值, 将它与表 3 对比, 可以发现:

规律 2. 当视频幅面变化时, 状态变量 K 的值基本不变, 而状态变量 B 的值发生变化; 且对不同幅面的视频, 其状态变量 B 的比值约等于幅面的比值.

表 5 不同码率下 4CIF 格式全 I 帧 MPEG-4 压缩码流状态变量拟合值

视频序列	不同码率(cycles/byte)下的拟合值				
	K_2000	K_4000	K_5000	K_6000	K_8000
City	219.54	212.27	209.55	208.65	205.58
Crew	240.75	222.80	218.47	220.02	212.14
Harbor	212.38	205.63	202.32	200.15	196.87
Ice	206.07	216.72	216.76	211.58	209.74
视频序列	不同码率(10^7 cycles)下的拟合值				
	B_2000	B_4000	B_5000	B_6000	B_8000
City	2.5289	2.5548	2.5659	2.5731	2.5911
Crew	2.5082	2.5445	2.5544	2.5501	2.5802
Harbor	2.5706	2.5956	2.6094	2.6209	2.6405
Ice	2.5275	2.5079	2.5115	2.5263	2.5317

需要指出的是, 虽然 H. 264 压缩码流 K 值的变化并不满足上述变化规律, 但是由于 I 帧作为关键帧, 其在码流中出现的频率很低, 预测误差对系统整体性能的影响并不大. 而且通过后续实验证明, 在确定了 B 值的基础上, I 帧的平均预测误差也不超

过 4%. 上述规律可以应用至 H. 264 压缩码流.

利用规律 1 和规律 2, 对 I 帧解码计算复杂度估计可以分为离线分析和在线预测两个部分. 离线分析得到针对某一幅面的压缩码流的状态变量统计值, 在线运行时根据接收视频与离线分析所使用的视频的幅面的比值对状态变量统计值进行调整, 调整后的值将用来对 I 帧解码复杂度进行预测. 因此有算法 1.

算法 1. I 帧计算复杂度预测.

输入: 压缩视频流中的 I 帧及其帧长

输出: 输入的 I 帧解码计算复杂度估计值

1. 离线分析得到某种幅面视频状态变量统计值:

1.1. 选择与解码器所采用的压缩标准一致的编码器, 采用此编码器将某种幅面大小(记为 S_1)的标准测试码流以全 I 帧形式在不同码率下进行编码;

1.2. 将压缩码流进行解码, 并获取解码每一帧的计算复杂度及压缩帧帧长, 记为 (C_n, L_n) ;

1.3. 以帧长为自变量, 解码计算复杂度为因变量对上述生成的 (C_n, L_n) 利用最小二乘法进行线性拟合, 获得 I 帧状态变量统计值, 记为 (K_1, B_1) .

2. 在线预测 I 帧的计算复杂度:

2.1. 初始化状态变量值, 令 $(K_{1(0)}, B_{1(0)}) = (K_1, B_1)$;

2.2. 计算接收视频幅面是否发生改变, 如改变, 则计算接收视频的幅面 S_2 与步 1 中所使用的视频幅面 S_1 之比, 即 $r = S_2/S_1$, 根据 r 对状态变量值进行比例缩放, 即 $(K_1, B_1) = (K_{1(0)}, r \times B_{1(0)})$. 否则保持 (K_1, B_1) 不变;

2.3. 读取当前帧帧长 L , 利用公式 $C = K_1 \times L + B_1$ 估算所需解码计算复杂度.

4.2.2 融合统计分析和状态方程的 P 帧计算复杂度动态预测

虽然 MPEG-4 与 H. 264 标准在预测模式、运动矢量精度、运动矢量范围等编码工具上有所区别, 但是它们都利用相邻帧之间的相关性进行状态方程更新, 因此两类码流中对于 P 帧的解码复杂度预测方法类似. 图 4 给出了对 P 帧解码计算复杂度进行预测的框图.

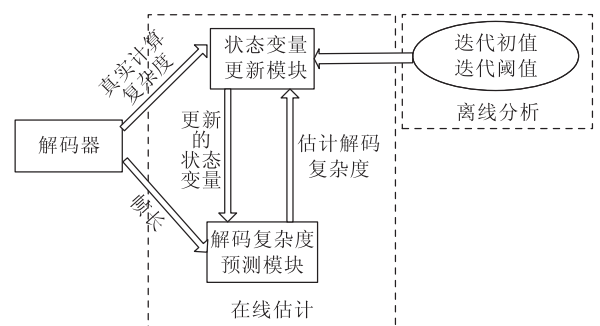


图 4 解码计算复杂度估计框图

在解码视频帧之前, 状态变量更新模块将更新后的状态变量输入至解码计算复杂度在线预测模块. 预测模块结合待解码帧的帧长, 对解码计算复杂度进行预测. 操作系统根据预测得到的计算复杂度进行 DVFS 调节之后, 解码器对当前帧进行解码并记录其真实计算复杂度, 将其输入至状态变量更新模块. 更新模块计算当前帧真实计算复杂度和预测值之间的差值, 并将其与迭代控制阈值进行比较, 选择 4.1.2 节中所给出的状态方程(9)或者(10)对状态变量进行动态更新, 更新后的状态变量将指导下一个帧的解码计算复杂度预测. P 帧计算复杂度动态预测算法如算法 2 所示.

算法 2. P 帧计算复杂度动态预测.

输入: 压缩视频流中的 P 帧及其帧长

输出: 输入的 P 帧解码计算复杂度估计值

1. 离线分析得到系统状态变量迭代初值和迭代控制阈值:

1.1. 确定迭代控制阈值(Δ_1, Δ_2), 本算法是通过确定状态变量 B 的范围(B_{\max}, B_{\min})来确定(Δ_1, Δ_2);

1.2. 确定迭代初值(K_P, B_P).

2. 在线动态预测输入 P 帧的计算复杂度:

2.1. 初始化系统状态变量值及迭代控制阈值, 令($K_{P(o)}, B_{P(o)}$) = (K_P, B_P), $B_{\max(o)} = B_{\max}$, $B_{\min(o)} = B_{\min}$;

2.2. 计算接收视频幅面是否发生改变, 如改变, 则计算接收视频的幅面 S_2 与步 1 中所使用的视频幅面 S_1 之比, 即 $r = S_2/S_1$. 根据 r 对状态变量值进行比例缩放, 即(K_P, B_P) = ($K_{P(o)}, r \times B_{P(o)}$), ($B_{\min} = r \times B_{\min(o)}, B_{\max} = r \times B_{\max(o)}$). 否则保持系统状态变量值及迭代控制阈值不变;

2.3. 读取当前帧帧长 L , 直接利用公式 $C = K_P \times L + B_P$ 估算其所需解码计算复杂度;

2.4. 对当前帧进行解码, 解码后得到真实计算量 C_R 以及当前帧解码复杂度估计误差 $\Delta = C_R - C$;

2.5. 计算当前帧的解码复杂度估计误差的阈值 [Δ_1, Δ_2], $\Delta_1 = B_{\min} - B_P$, $\Delta_2 = B_{\max} - B_P$, 判断估计误差所处阈值范围, 如果 $\Delta_1 \leq \Delta \leq \Delta_2$, 则按式(9)更新 B_P 为 $B_P = B_P + \Delta$; 否则按式(10)更新 K_P 为 $K_P = \Delta/L + K_P$, 将更新后的 (K_P, B_P) 用于下一个 P 帧解码计算复杂度预测.

由于篇幅的限制, 迭代控制阈值(Δ_1, Δ_2) 以及迭代初值(K_P, B_P) 的计算方法没有详述, 具体方法可参看作者前期工作^[9].

5 实验结果与分析

采用基于统计的解码计算复杂度预测算法 (Statistical Analysis based Prediction, SP) 以及融合统计分析和状态方程的解码复杂度预测算法

(Statistics merged State Variable Analysis, SSVA) 分别对 I 帧和 P 帧的解码复杂度进行估计. 由于本算法在实际运行阶段利用了相邻帧状态变量之间的相关性进行解码计算复杂度预估, 属于基于预测算法, 本节将给出与伊利诺伊大学香槟分校 Nahrstedt 等人^[8]提出的 CDP (Cumulative Distribution based Prediction) 算法的对比.

为了验证算法的有效性, 实验分别选取了采用 MPEG-4 和 H. 264 两种视频编码标准的解码器, 在不同平台上进行测试. 其中 MPEG-4 解码器运行在基于 TI TMS320DM642 DSP 芯片的嵌入式开发板上, 采用 DIVX 解码器. H. 264 解码器运行在 SimpleScalar 仿真器上^[10], 仿真器所仿真的硬件环境为基于 X86 架构的通用 PC, 解码器源码选取开源代码 P264. 如文献^[8]所述, CDP 算法包含 3 个控制参数: 视频帧解码率 ρ , 每组视频帧的数目 ω 和计算复杂度分组间隔 g , 在本实验中, 将其分别设置为 0.96, 50 帧和 10000 个时钟周期.

5.1 基于统计分析的 I 帧计算复杂度预测结果

5.1.1 MPEG-4 压缩码流 I 帧预测结果

利用 4.2.1 节中所提出的算法 1, 对 I 帧解码复杂度进行估计. 首先选取 Akiyo、Coastguard、Foreman 和 Mobile 等 CIF 格式的视频序列作为测试序列, 采用 MPEG-4 编码标准将其在不同码率下 (800 kbps、1000 kbps、1500 kbps、2000 kbps 和 2500 kbps) 以全 I 帧形式进行压缩并解码, 记录每一帧视频的帧长和解码计算复杂度, 记为 (C_n, L_n). 以帧长为自变量, 解码计算复杂度为因变量, 利用最小二乘法对得到的 (C_n, L_n) 进行线性拟合, 得到状态变量 (K, B) 值为 (215.51, 6.4108×10^6).

利用拟合得到的状态变量值预测 CIF 格式 MPEG-4 压缩码流的解码计算复杂度. 将测试视频 (Akiyo、Hall、Coastguard、Crew、Foreman 和 Mobile) 在不同码率下 (800 kbps、1000 kbps、1500 kbps、2000 kbps 和 2500 kbps) 按全 I 帧进行压缩, 并将其输入到解码器进行解码复杂度动态预测. 图 5 给出了预测误差的概率密度分布.

如图 5 所示, 采用 SP 所得到的估计误差的概率密度分布呈正态分布, 且平均估计误差为 0.47%, 近 89% 视频帧预测误差范围在 1% 以内, 最大预测误差为 2.94%, SP 方法精确度高于 CDP 方法. 而且采用 SP 方法的在线运行开销非常小, 计算复杂度可以通过一次乘法以及一次加法预测得到. 在本实验环境下, SP 在线运行开销仅为 54 cycles/帧.

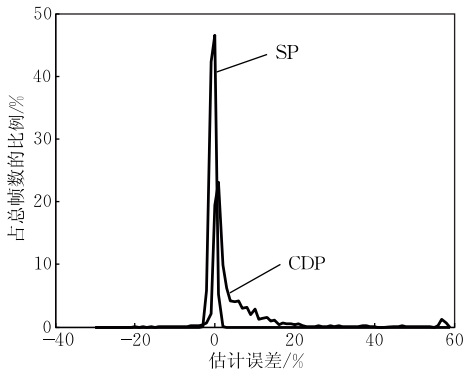


图 5 CIF 格式视频序列预测结果

更进一步的,利用 CIF 格式视频序列所得到的状态变量值动态预测 4CIF 格式视频序列解码计算复杂度.将测试序列(Ice、Harbor、Crew 和 City)在不同码率下(2000 kbps、4000 kbps、5000 kbps、6000 kbps 和 8000 kbps)按全 I 帧形式压缩,并将其输入到解码器中进行解码复杂度动态预测.由于在线输入视频幅面(4CIF)与离线分析视频所采用的幅面(CIF)不同,按照 4.2.1 节中所提出的规律 2,根据视频幅面的比值将状态变量(K, B)值修正为(215.51, 2.5643×10^7).为了评价规律 2 的有效性,本实验还给出了利用 4CIF 幅面视频进行离线拟合分析所得到的状态变量值:(218.25, 2.5421×10^7).

图 6 给出了采用不同预测方法所得到的预测误差的概率密度分布.其中将采用修正后的 CIF 格式视频序列拟合得到的状态变量值进行预测的方法记为 SPSimple,采用 4CIF 幅面视频直接离线拟合得到的状态变量值进行预测的方法记为 SP.

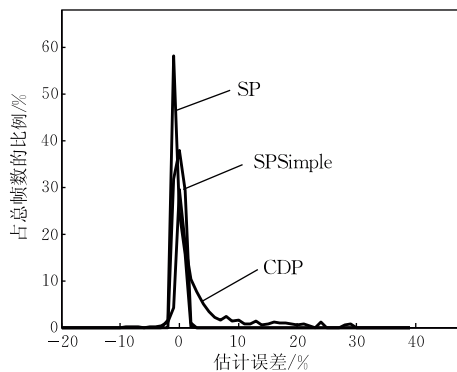


图 6 4CIF 格式视频序列预测结果

实验结果表明,SP 方法预测结果仍然很准确,这再次验证了 4.2.1 节中所提到的规律 1,即对同一幅面视频序列,其状态空间基本不变这一规律是正确的.虽然 SPSimple 的预测结果没有 SP 精确,但是也有不错的预测结果.近 69.79% 视频帧的预

测误差在 1% 以内,近 99% 的视频帧预测结果在 2% 以内,最大预测误差为 2.32%.这一结果表明规律 2 是正确的,另一方面也暗示了本算法的鲁棒性,虽然 SP 和 SPSimple 状态变量值稍有不同,但两者都可以达到比较好的预测效果.

5.1.2 H.264 压缩码流 I 帧预测结果

选取 CIF 格式的视频序列(Akiyo、Coastguard、Crew、Mobile)作为测试序列,采用 H.264 编码标准将其在不同码率下(500 kbps、800 kbps、1000 kbps、1500 kbps、2000 kbps)以全 I 帧形式压缩并解码,利用最小二乘法进行线性拟合得到状态变量(K, B)值为(222.6, 1.4749×10^7).

利用拟合得到的状态变量值对 CIF 格式 H.264 压缩码流的解码计算复杂度进行预测.将测试视频(Akiyo、Hall、Coastguard、Crew、Foreman 和 Mobile)在不同码率下(500 kbps、800 kbps、1000 kbps、1500 kbps、2000 kbps)按全 I 帧进行压缩,并将其输入到解码器进行解码复杂度动态预测.表 6 给出了不同序列在不同码率时的预测误差绝对值的均值.

表 6 采用 H.264 编码标准的全 I 帧测试序列在不同码率时预测误差绝对值均值

视频序列	预测误差绝对值的均值/%				
	500	800	1000	1500	2000
Akiyo	1.88	1.16	0.70	0.75	0.31
Hall	3.45	2.05	1.26	1.24	2.86
Coastguard	3.92	3.48	3.52	2.84	2.08
Crew	1.27	1.30	1.28	1.67	2.21
Foreman	3.66	3.22	2.46	1.21	0.92
Mobile	3.96	3.69	2.60	1.30	1.32

实验结果表明,虽然 H.264 编码标准中的帧内预测模式较 MPEG-4 编码标准复杂,状态变量空间有小幅度变化,但在确定了状态变量(K, B)数量级的基础上,仍能保持较高精度,所有测试序列估计误差均值都保持在 4% 以内.

5.2 融合统计分析和状态方程的 P 帧解码计算复杂度预测结果

5.2.1 MPEG-4 压缩码流 P 帧在线预测结果

利用 4.2.2 节所述的算法对 P 帧解码复杂度进行估计.选取 CIF 格式测试序列 Akiyo、Hall、Coastguard、Foreman、Mobile、Crew 作为测试序列,并按照文献[9]中所述方法获取迭代控制阈值及迭代初值: $\Delta_1 = 1.7 \times 10^6 - B_{(i-1)}$, $\Delta_2 = 2.4 \times 10^6 - B_{(i-1)}$; $(K_p, B_p) = (325.58, 2.3171 \times 10^6)$.

利用上述初值对 MPEG-4 压缩码流进行解码计算复杂度在线估计.表 7 给出了 SSVA 和 CDP 的平均绝对估计误差.

表 7 不同码率时 P 帧预测误差绝对值的均值

视频序列	预测方法	预测误差绝对值的均值/%				
		800	1000	1500	2000	2500
Akiyo	SSVA	6.28	3.95	2.33	3.76	2.96
	CDP	21.11	26.04	43.87	104.20	115.95
Hall	SSVA	2.89	2.05	1.52	1.50	2.78
	CDP	10.32	12.86	17.19	10.98	36.04
Coastguard	SSVA	2.46	2.65	1.85	3.36	2.58
	CDP	13.95	12.31	12.66	15.96	18.16
Crew	SSVA	4.80	5.00	5.32	5.47	5.49
	CDP	50.15	61.86	63.96	60.20	57.54
Foreman	SSVA	3.63	3.84	3.86	4.12	3.60
	CDP	15.70	15.21	18.57	18.28	22.85
Mobile	SSVA	2.33	2.45	2.36	1.95	2.38
	CDP	20.24	20.81	20.24	18.11	15.08

实验结果表明 SSVA 适用于不同内容、不同码率的压缩视频,特别是对于运动复杂度中等、纹理复杂度中等的视频序列(如 Hall、Coastguard 和 Mobile 序列),平均预测误差一般都在 3% 以内.对于运动非常剧烈的视频(如 Crew 序列),由于相邻帧之间的相关性较弱,导致相邻帧状态变量相似性减小,预测结果较差,但其平均估计误差仍然小于 6%.

图 7 给出了各序列预测误差概率密度分布,从图中可以看出 SSVA 算法估计误差呈正态分布,且估计误差均值和方差都接近 0,证明 SSVA 算法预

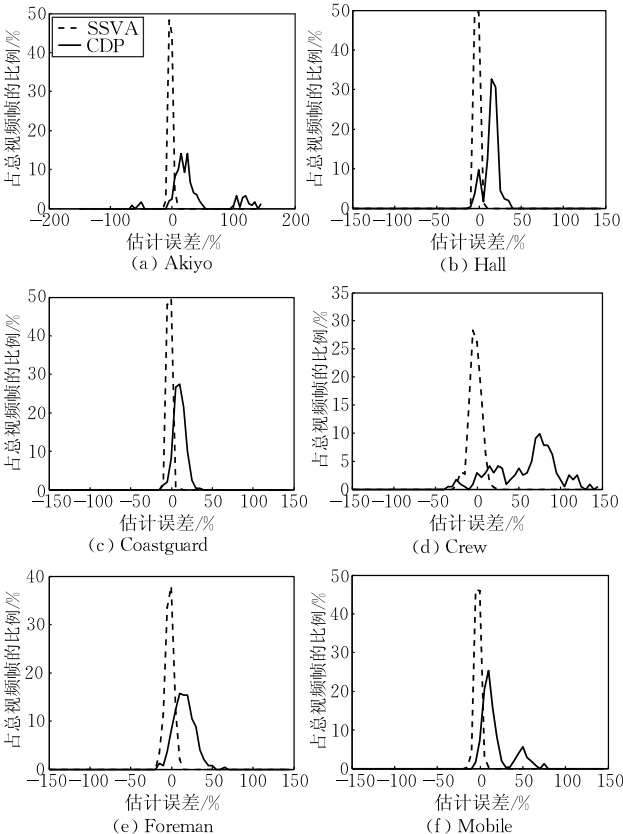


图 7 预测误差概率密度分布

测非常平滑.对 6 个序列综合计算,估计误差在 5% 以内的比例为 84.6%,误差在 10% 以内的比例为 96.6%,SSVA 精确度较高.

更进一步的,利用 CIF 格式视频序列所得到的状态变量的值动态预测 4CIF 格式视频序列解码计算复杂度.将测试序列(Ice、Harbor、Crew 和 City)在不同码率下(800 kbps、1000 kbps、1500 kbps、2000 kbps 和 2500 kbps)压缩,并将其输入到解码器中进行解码复杂度动态预测.由于在线输入视频幅面(4CIF)与离线分析视频所采用的幅面(CIF)不同,根据视频幅面的比值将状态变量迭代初值修正为 $(325.58, 9.2684 \times 10^7)$.为了评估该迭代初值的有效性,本实验还给出了利用 4CIF 幅面视频进行离线拟合分析所得到的迭代初值 $(344.24, 8.6397 \times 10^7)$ 进行在线估计所得到的估计结果.

实验结果如表 8 所示,采用 CIF 序列初值得到的估计初值记为 EST,D1 序列直接拟合得到初值记为 FIT.从表 8 中可以看出,两者效果类似,精确度都比较高.这说明迭代初值对本文所提出的算法影响并不大,只要确定状态变量(K,B)的数量量级,在线预测算法会迅速地根据估计误差调整状态变量 K 或者 B,达到快速收敛的效果.

表 8 MPEG-4 用 CIF 序列估计 D1 序列拟合初值(EST)以及用 D1 序列拟合选取初值(FIT)的预测绝对误差均值

视频序列	预测方法	预测绝对误差的均值				
		800	1000	1500	2000	2500
City	FIT	1.36	1.25	1.34	1.60	2.18
	EST	1.42	1.26	1.48	1.38	2.07
Crew	FIT	4.30	5.10	4.52	4.87	4.74
	EST	4.74	4.74	4.66	5.16	5.25
Harbor	FIT	2.56	2.23	1.94	1.93	1.77
	EST	2.57	2.26	2.18	2.77	2.20
Ice	FIT	3.11	3.14	2.38	2.31	2.11
	EST	2.98	2.96	2.34	2.34	2.24

对算法在线运行复杂度进行分析,一旦通过离线拟合确定了迭代控制阈值和状态变量迭代初值,在线运行的过程中只需要通过一次加法和一次乘法便可获得计算复杂度预测值,而状态变量的修正过程也只需两次加法和一次乘法.在本实验所用的 DSP 环境下,算法的在线运行开销为 103 cycles/帧.

5.2.2 H.264 压缩码流 P 帧在线预测结果

采用同 MPEG-4 压缩码流计算复杂度估计类似的方法,选取 CIF 格式测试序列 Akiyo、Hall、Coastguard、Foreman、Mobile、Crew 作为测试序列,并按照文献[9]中所述方法获取迭代控制阈值及迭代初值 $\Delta_1 = 9.5 \times 10^6 - B_{(t-1)}$, $\Delta_2 = 1.26 \times 10^7 -$

$B_{(i-1)}$; $(K_P, B_P) = (1109.99, 1.061 \times 10^7)$. 将 CIF 格式测试序列 (Ice, Harbor, Crew 和 City) 在不同码率下 (200 kbps, 400 kbps, 500 kbps, 600 kbps 和 800 kbps) 进行压缩, 对用 H. 264 编码标准进行编码的测试序列进行在线解码复杂度动态预测, 实验结果如表 9 所示.

表 9 采用 H. 264 编码标准的不同视频测试序列在不同码率下的预测误差绝对值均值

视频序列	预测误差绝对值的均值/%				
	200	400	500	600	800
Akiyo	0.53	0.76	1.14	1.89	0.80
Hall	0.76	0.63	0.63	0.61	0.89
Coastguard	0.55	0.43	0.55	0.77	0.96
Crew	1.60	1.65	1.65	1.68	2.05
Foreman	0.68	0.78	0.90	1.16	1.59
Mobile	0.51	0.39	0.74	0.64	0.87

从表 9 可以看出, 本文提出的 P 帧解码复杂度动态预测算法对 H. 264 编码标准也非常有效, 在线估计误差不超过 2%. 虽然 MPEG-4 和 H. 264 采用了不同的编码方式, 但是本算法是从解码系统机理出发, 利用相邻帧之间的相关性来揭示系统状态变量随视频内容变化的规律; 是从物理层面引出的一种解析方法, 满足解码器运行规律, 因此预测效果非常好. 对于采用其它编码方式进行编码的压缩码流 (如基于感兴趣区域编码的 ROI^[11]), 只要其采用基于 MC-DCT 混合编码框架, 相邻帧之间状态空间就具有极强的相关性, 就可以利用本文所提出的状态空间的变化规律进行求解, 算法普适性强.

5.2.3 解码系统状态变量轨迹分析

为了突出本文所提出的状态变量的物理含义, 本节将给出采用状态方程对状态变量进行求解后所得到的每一帧视频对应的状态变量值. 图 8 给出了在码率为 500 kbps 时, 采用 MPEG-4 编码标准进行压缩的各测试序列的状态变量在状态空间的轨迹.

从图 8 可以看出, 当视频运动简单, 没有突变时, 表征视频预测模式、运动矢量复杂度的状态变量 K 值的变化幅度较小, 如 Akiyo、Hall、Mobile 序列, K 值变化在 30 以内; 当视频出现突变时, 如 Coastguard 序列在 80 帧附近发生一次镜头切换, K 值发生大幅度跳跃; Foreman 序列从 180 帧到 230 帧发生了数次镜头切换, K 值也发生了几次跳变; 对于 Crew 序列, 由于全序列的运动非常复杂, K 值的切换也是非常明显的.

当视频运动趋于稳定后, K 值也基本稳定. 之后, 解码器将在某一 K 值平面上通过更新 B 值 (即残差的变化) 寻求最优位置, 表现在图形中就是 B 值在某一个 K 值平面中震荡. 这也进一步体现了编码信息与被编码残差之间质与量的关系.

值得注意的是, 在图 8 中 Mobile、Akiyo 序列的 K 值平面在开始处发生过大幅度切换, 这主要是因为迭代初值所表征的视频内容的特征与当前序列的特征不一致所导致, 解码器必须通过更新 K 值平面的方法迅速逼近最优位置. 一旦逼近最优状态平面, 将通过小幅调整 B 的方法精细搜索最优位置.

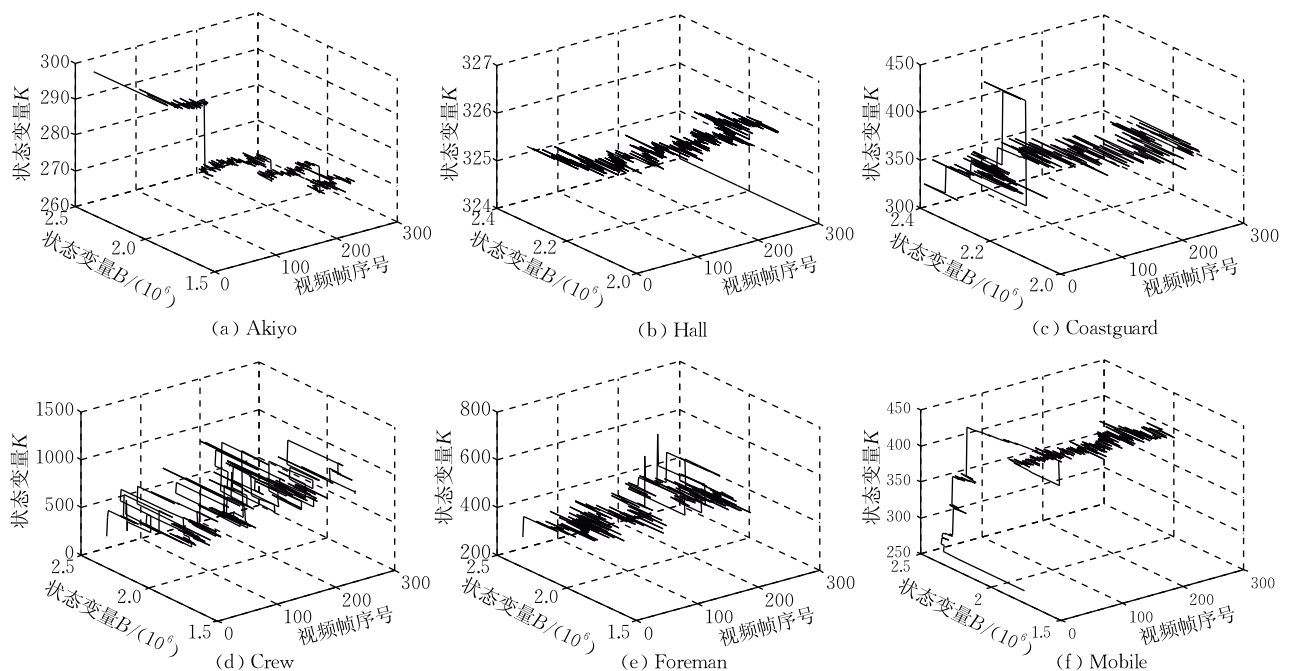


图 8 CIF 格式测试序列状态空间变化图

6 结 论

本文给出了一种基于状态变量法的视频流解码计算复杂度估计方法. 首先通过统计分析, 对解码计算复杂度与帧长之间的关系进行建模, 得到解码计算复杂度与帧长之间的线性关系, 并分析得出模型系数作为状态变量所包含的物理含义, 从而将在线模型系数求解问题转化为对系统状态变量进行更新的问题.

通过统计分析, 发现对于 I 帧, 其状态变量值仅与视频幅面相关, 可通过离线分析方法获得状态变量统计均值, 在线运行过程中直接利用该值对解码计算复杂度进行估计. 对于 P 帧, 则利用相邻视频帧编码信息及残差的变化特点, 结合状态变量所表征的物理含义, 将相邻视频帧的相关性转化为状态变量随视频内容变化而变化的规律, 并以预测误差作为反馈动态调整相邻 P 帧状态变量值, 最终快速高效地预测解码计算复杂度.

本方法采用简单的线性模型对解码复杂度进行估计, 系统输入(帧长)可由压缩码流直接获得, 解码器设计流程简单、码流兼容性好; 能对系统状态变量进行动态更新, 及时反映视频内容变化, 估计结果精确度高; 进行在线更新时仅需利用上一次更新结果做简单运算, 系统运行时开销小, 特别适合资源受限的嵌入式系统.

最后, 本文所提出的状态变量法利用视频的语义层信息, 通过挖掘状态的语义层含义直接构建系统状态方程, 摒弃了复杂的数学建模. 所得到的状态方程直接反映了多媒体解码系统固有的物理特性, 因此有非常好的效果. 这一建模方法为对其它多媒体系统进行理论分析和建模提供了新思路.



TIAN Ting, born in 1985, Ph. D. candidate. Her main research interests include cross-layer design for embedded multimedia system, multimedia streaming.

参 考 文 献

- [1] Benini L, Micheli G De. System-level power optimization: Techniques and tools. *ACM Transactions on Design Automation of Electronic Systems*, 2000, 5(2): 115-192
- [2] Li K, Kumpf R, Horton P, Anderson T. A quantitative analysis of disk drive power management in portable computers//*Proceeding of USENIX Winter Technical Conference*. San Francisco, USA, 1994: 279-291
- [3] Ishihara T, Yasuura H. Voltage scheduling problem for dynamically variable voltage processors//*Proceeding of the 1998 International Symposium on Low Power Electronics and Design*. Monterey, USA, 1998: 197-202
- [4] Huang Y C, Tran A. V, Wang Y. A workload prediction model for decoding mpeg video and its application to workload-scalable transcoding//*Proceeding of the 15th International Conference on Multimedia*. Augsburg, Germany, 2007: 952-961
- [5] Ma Z, Hu H, Wang Y. On complexity modeling of H. 264/AVC video decoding and its application for energy efficient decoding. *IEEE Transactions on Multimedia*, 2011, 13(6): 1240-1255
- [6] Kontorinis N, Andreopoulos Y, van der Schaar M. Statistical framework for video decoding complexity modeling and prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2009, 19(7): 1000-1013
- [7] Lee S W, Kuo C-C J. Complexity modeling of spatial and temporal compensations in H. 264/AVC decoding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2010, 20(5): 706-720
- [8] Yuan W, Nahrstedt K. Energy-efficient soft real-time CPU Scheduling for Mobile Multimedia Systems//*Proceedings of the 19th ACM Symposium on Operating Systems Principles*. New York, USA, 2003: 149-163
- [9] Tian T, Yu S S, Guo H X. Linear model-based adaptive prediction for video decoding complexity//*Proceeding of 2011 IEEE International Conference on Multimedia & Expo*. Barcelona, Spain, 2011: 1-6
- [10] Austin T, Larson E, Ernst D. SimpleScalar: An infrastructure for computer system modeling. *IEEE Computer*, 2002, 35(2): 59-67
- [11] Lu S P, Zhang S H. Saliency-based fidelity adaptation pre-processing for video coding. *Journal of Computer Science and Technology*, 2011, 26(1): 195-202

YU Sheng-Sheng, born in 1944, professor, Ph. D. supervisor. His main research interests include computer storage systems, multimedia technologies.

GUO Hong-Xing, born in 1971, Ph. D., associate professor. His main research interests include embedded computing, multimedia networking.

SU Shu-Guang, born in 1975, Ph. D., lecturer. His main research interests include video coding and multimedia communications, embedded system and middleware.

Background

This work is supported by the National Natural Science Foundation of China (Grant Nos. 61173044, 60873029) and the Fundamental Research Funds for the Central Universities. The main purpose is providing high-performance and energy-efficient multimedia technologies over a variety of hardware and software platforms.

Energy-efficient optimization has been becoming an increasingly important aspect in the design of those highly complicated multimedia systems, especially for battery-limited mobile devices. Scalability of hardware components provides us a new dimension to achieve this goal. Since the computational complexity and storage demand of the multimedia system vary with the video content, hardware platforms can adjust their resources according to the demand of the multimedia applications during runtime process. The effectiveness of such methods depends on the prediction accuracy of computational complexity and storage demand.

Video decoding is one of the most popular multimedia applications on mobile devices. Two kinds of methods for video decoding complexity prediction have been evolved during the past years. The first is analysis-based modeling, which works in a fine granularity manner and the prediction results are accurate. However, fine analysis causes redundancy in either video decoding process or compressed bitstre-

ams, which leads to low efficiency. The second is prediction-based method, which utilizes the correlation between the neighboring frames. Although the prediction overload can be low in terms of the coarse grained feed-back control strategy, desirable prediction results are hard to achieve since it is difficult to exploit the correlation between decoding complexities of the neighboring frames.

In this paper, we utilize the linear relationship between frame length and decoding complexity and consider the decoding system as a state-driven system, where the state means the characteristics of video content. Unlike those complicated mathematical modeling methods, our method utilizes the intrinsic semantic characteristic of video content and exploits the state variation patterns from the semantic layer. Then the state function is derived naturally with the state variation pattern and applied to update the state trajectory and perform online decoding complexity predicting for both MPEG-4 and H.264 compressed stream. The experimental results indicate our method can achieve good prediction results for bitstreams over a wide range of video contents and coding standards. It's very simple and effective.

Our past work was published at IEEE ICME 2011 and nominated as the best paper candidate. We will continue our work on design of energy-efficient multimedia systems.