

基于 TMS320DM6437 平台的视频 系统设计与实现*

·实用设计·

贾浩,崔慧娟,唐昆

(清华大学 电子工程系;微波与数字通信技术国家重点实验室,北京 100084)

【摘要】针对 TI 公司芯片 TMS320DM6437 硬件平台,在开源编码器 X264 基础上,基于 TI 公司提供的 DSP/BIOS 系统,实现了多线程,具有视频采集、编码、传输、解码和显示功能的视频系统。实验结果表明,通过 DSP/BIOS 的实时可伸缩性内核,合理设置各个任务线程之间的通信和调度机制,并利用实时分析和配置工具,合理配置代码和数据空间,可以实现整个系统的高速运行,并可以大大缩短系统的开发时间。

【关键词】TMS320DM6437; DSP/BIOS; 编码器; 多线程

【中图分类号】TN92; TP391.9

【文献标识码】A

Design and Implementation of Video System Based on TMS320DM6437 Platform

JIA Hao, CUI Hui-juan, TANG Kun

(State Key Laboratory on Microwave and Digital Communications; Department of Electronic Engineering, Tsinghua University, Beijing 100084, China)

【Abstract】In this paper, according to the characteristics of TI's TMS320DM6437 platform, a video system is implemented based on an open source H.264 encoder X264. TI's DSP/BIOS system is used to speed up the development process. This multithread video system implements video capture, video encode, transmission, video decode and video display. Experimental results show that, after making full use of DSP/BIOS system and setting up the thread schedule, the entire system can run at a high speed and the development time can be reduced significantly.

【Key words】TMS320DM6437; DSP/BIOS; X264 encoder; multithread

1 引言

TI公司的 TMS320DM6437(简称 DM6437)是一款高性能的通用数字媒体处理器^[1],属于 TI 公司的 C64x+ 系列,主频可达 400~700 MHz,峰值运算速率更是高达 3 200~5 600 MI/s(兆指令/秒)。相对于 TI 公司的 C64x 系列 DSP,在保持指令兼容性的同时,DM6437 拥有自己独特的扩展指令集,而且拥有更大的 cache 配置容量和更为灵活的 cache 配置方式^[2]。针对视频应用,DM6437 还提供了强大的视频处理子系统(Video Processing Sub-System,VPSS)^[3],其包括提供数字视频输入及预处理的视频处理前端(Video Processing Front End,VPFE)和提供数字视频输出及驱动显示设备的视频处理后端(Video Processing Back End,VPBE)。这一子系统提供的各种接口为视频的采集和显示处理提供了极大的便利。

DSP/BIOS 是 TI 公司为了满足用户各种实时性应用的需求提供的一种尺寸可裁剪的实时操作系统内核^[4],它为需要实时线程调度和同步、主机与 DSP 间通信或实时监测的应用而设计。DSP/BIOS 提供了抢占式多线程、硬

件抽象、实时分析和配置工具。通过使用 DSP/BIOS 提供的一系列丰富的内核服务,开发者能够快速创建满足实时要求的多任务应用程序。这些内核服务具有跨越 C6000, C5000 和 C28x DSP 平台的标准 API 接口,能被用户程序调用,易于移植。

H.264/AVC 是视频编码专家组(ITU-T VCEG)和运动图像专家组(ISO/IEC MPEG)共同开发的新一代视频处理标准^[5]。H.264 标准具有如下特点:1)支持多尺寸块变换和多参考帧运动补偿,1/4 像素精度运动搜索和基于上下文的算术编码和变长编码等高级特性,因此能够达到很好的压缩效率;2)把编码器框架分为 VCL 部分和 NAL 部分,VCL 单元负责高效压缩视频,而 NAL 单元使压缩码流具有良好的网络适应性;3)支持冗余条带编码、数据分割,任意条带顺序等抗误码选项工具使得视频传输相对稳健。相对于以往的视频编码标准,H.264 更加适合无线信道下高压缩率的应用场景。但它具有相当高的编码复杂度,实时编解码算法需要高性能专用集成电路(ASIC)或者 DSP 的支持。

* 国家自然科学基金项目(60572081)

2 系统结构

2.1 线程类型的选择

DSP/BIOS 支持多种不同优先级的线程,包括硬件中断(HWI)、软件中断(SWI)、任务(TSK)、后台线程(IDL),每种线程类型都有不同的执行和抢占特性^[4]。

根据系统需求,线程需要满足以下条件:

- 1) 多个优先级,使初始化线程能够先于其他线程运行,完成系统初始化;
- 2) 多种与其他线程共享数据的方式,根据需要共享数据的大小和类型,选择不同的方式在线程之间传递数据;
- 3) 多种执行状态,需要任务具有阻塞状态以等待所需数据。

根据这些要求,本文选定任务(TSK)^[4]实现系统的各个线程,通过多优先级的选择实现线程的依次执行,通过流、管道、邮箱实现多线程中各种不同性质数据的共享,通过信号灯和邮箱实现多线程的同步,TSK 的这些特性完全可以满足系统对线程的各种需求。

2.2 系统的结构设计

视频系统的结构如图 1 所示。

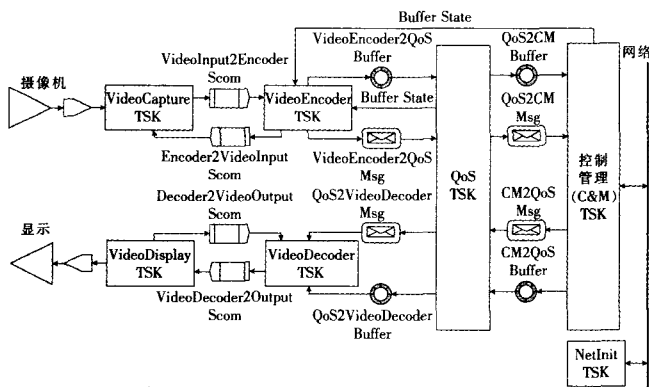


图 1 视频系统结构框图

根据系统需求,共设置了视频采集/显示(VideoCapture/Display)、视频编码/解码(VideoEncoder/Decoder)、网络初始化(NetInit)、服务质量(QoS)、控制管理(Control & Management)等任务线程,每个线程完成其特定的功能,并根据系统的运行顺序设置不同的优先级,线程之间通过邮箱(MBX)、同步通信模块(SCOM)、共享存储空间等实现线程之间的同步、通信和数据传递。视频数据按以下顺序在系统中传递:

- 1) VideoCapture TSK 从视频采集设备(如 CCD Camera)获取 YUV 4:2:2 的视频帧,对色度分量进行降采样后得到 YUV 4:2:0 数据;

2) YUV 4:2:0 数据被送至 H.264 编码器,经过编码后得到编码后的比特流;

3) 在 QoS TSK 缓冲区不满的情况下,H.264 比特流被送至 QoS TSK,进行组包;

4) QoS TSK 产生的固定长度的包送至 CM TSK,在选定信道编码模式下进行信道编码,编码后数据送至网络,发送到接收端;

5) 接收端收到足够的信道包后进行信道解码,得到固定长度的数据包;

6) 数据包被送至 QoS TSK,拆包后形成 H.264 的比特流;

7) 比特流送至 H.264 解码器后经过解码形成 YUV 4:2:0 的视频数据帧;

8) VideoPlay TSK 从解码器获得数据帧后对色度分量进行差值,得到 YUV 4:2:2 视频帧;

9) 视频帧在输出显示设备上显示。

以上步骤实现了发送端采集编码视频、接收端解码显示视频的系统功能,并可以提供独立于信源采集编码的信道编码方式以满足不同网络条件的不同需求。

2.3 线程之间的通信和数据传输

2.3.1 邮箱(MBX)

MBX 是 DSP/BIOS 提供了一种用于任务间传递消息的方式^[4]。DSP/BIOS 系统维护一个固定长度的共享邮箱来实现任务间的数据传递,可以保证信息流的输入不会超过系统处理这些信息的能力。

函数 MBX_handle, MBX_create(msgsize, msglength, attris)^[6]和 Void MBX_delete(mbx)^[6]分别用于动态地创建和删除邮箱,在创建邮箱时可以指定邮箱的长度和单个信息的大小。还可以在 DSP/BIOS 中静态的创建邮箱对象,并设置各种参数。函数 Bool MBX_post(mbx, msg, timeout)^[6]和 Bool MBX_pend(mbx, msg, timeout)函数^[6]则分别用于向邮箱中发布信息 and 从邮箱中读取信息。

但 MBX 存储的消息数量有限,而且每个消息的长度都较小(例如 8 byte),所以 MBX 在系统中往往被用于信令的传递。在本文中,MBX 用于在多个任务之间传递控制信令,信令的大小都不超过 8 byte。

在本文中,共设置 mbx_video_capture/player, mbx_videoencoder2QoS, mbx_QoS2videodecoder, mbx_QoS2CM 和 mbx_CM2QoS 等 MBX。其中,mbx_video_capture/player 主要用于处理采集、显示端的缓冲区状态;mbx_videoencoder2QoS 和 mbx_QoS2videodecoder 主要用于协调 QoS 任务与 VideoEncoder 任务和 VideoDecoder 任务执行顺

序。在采集编码端,当 VideoEncoder 任务产生 H.264 比特流后,通过邮箱 mbx_videoencoder2QoS 通知 QoS 任务对缓冲区中的数据进行处理。QoS 任务处理得到数据后,通过邮箱 mbx_QoS2CM 通知 CM 任务,进行信道编码并通过网络发送数据。解码显示端的 3 个 MBX 起到的作用与采集编码端 MBX 的作用基本一致。

2.3.2 同步通信模块(SCOM)

SCOM 是通过封装 DSP/BIOS 提供的信号灯(Semaphore)^[4]和队列(Queue)^[4]两种基本组件实现的,用于在多个线程之间通信的一种基本模块。一个任务通过调用 SCOM_putMsg() 函数将 SCOM 信息放置在一个 SCOM 队列中,发送给其他任务。目标任务通过 SCOM_getMsg() 函数从队列中获取消息。一般情况下,发送任务为需要传递的数据申请一定的空间,并将指向这一空间的指针封装在 SCOM 消息中,接收任务从 SCOM 消息中得到需要处理数据所在空间的地址,对数据进行处理。通过共享空间地址的模式,实现了数据在任务线程之间的高速传递,如图 2 所示。

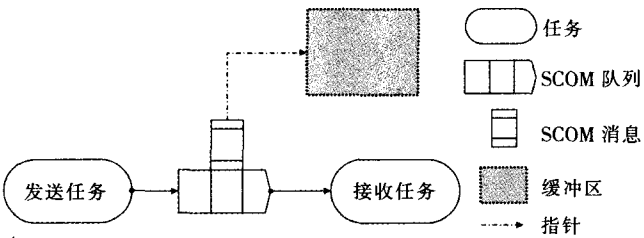


图 2 SCOM 模块实现线程之间数据传递

本文中,在 VideoCapture 和 VideoEncoder 之间、VideoDecoder 和 VideoDisplay 之间设置了 2 组 SCOM 信息。在采集编码端,VideoCapture 任务申请存储空间,将采集到的 YUV 数据放置在此存储空间内,然后将数据空间的指针通过 SCOM 信息发送至 VideoEncoder 任务,VideoEncoder 任务通过指针直接访问 YUV 视频数据,进行 H.264 编码。这种指针的传递避免了较大数量数据的复制,节省了任务线程之间数据传递的时间。

2.3.3 线程间的数据传递

通过 SCOM 模块,可以实现 2 个任务线程之间快速传递大量数据的目的。但是为了适应不同的网络条件,有必要在 VideoCapture/Display 和 QoS 之间、QoS 和 CM 之间添加缓冲,使得下层网络条件的变化不会影响到编码器的正常运行,使任务之间尽可能的独立。采用 SCOM 模块会增加任务线程之间的耦合度,所以在 VideoCapture/Display 和 QoS 之间、QoS 和 CM 之间传递大量数据时,需要考虑采用其他方式。

任务线程之间最快的通信方式就是全局变量,因此在使用全局变量的基础上需要选择一种能够实现数据缓冲的数据结构,本文使用循环队列来实现线程间的大量数据传递,过程如图 3 所示。

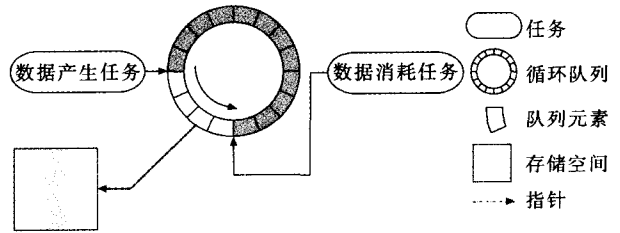


图 3 循环队列实现线程间大量数据传递

在整个系统启动之初,为需要传输、缓冲数据的任务线程创建循环队列,并记录循环队列中的未使用队列和已使用队列的起始位置,每个队列元素中包含一个指向已申请好的空间的指针,如图 3 所示。队列的具体使用描述如下:在产生数据的任务线程产生数据前,获得未使用队列的第一个队列元素中的空间地址指针,将产生的数据放置在这一空间后,将这一队列元素标记为已使用,更新循环队列的未使用队列的起始位置;使用数据的任务线程执行时,根据已使用队列的第一个队列元素中的地址指针获得需要的数据,对数据进行处理后,将队列元素标记为未使用,更新循环队列中的未使用队列的起始位置。

通过循环队列和指针传递,直接使用地址对数据进行访问,既避免了大量数据在任务线程之间的复制,又可以使数据的传递独立于 2 个任务线程,提高各个模块的可移植性。此外,还可以在在一定程度上缓解不同任务线程之间数据处理速度不同造成的匹配问题,对网络条件的变化也具有了更好的适用性。

2.4 线程的调度(图 4)

采集显示端的运算复杂度很高,H.264 编码器的编码速度严重限制了系统采集编码端的运行速度,因此在线程调度中应以 H.264 编码器为核心,合理调整其他线程与编码线程之间的速度关系,才能实现采集编码端的高效运行。

采集编码端的调度流程如图 4a 所示,描述如下:

- 1) 具有高优先级的 NetInit TSK^[7]首先运行,完成网络的初始化;
- 2) VideoCapture TSK 开始循环运行,采集视频数据完成后,通过 SCOM 消息将视频数据地址传送至 VideoEncoder TSK,VideoCapture TSK 进入阻塞状态;
- 3) VideoEncoder TSK 在接收到 VideoCapture TSK

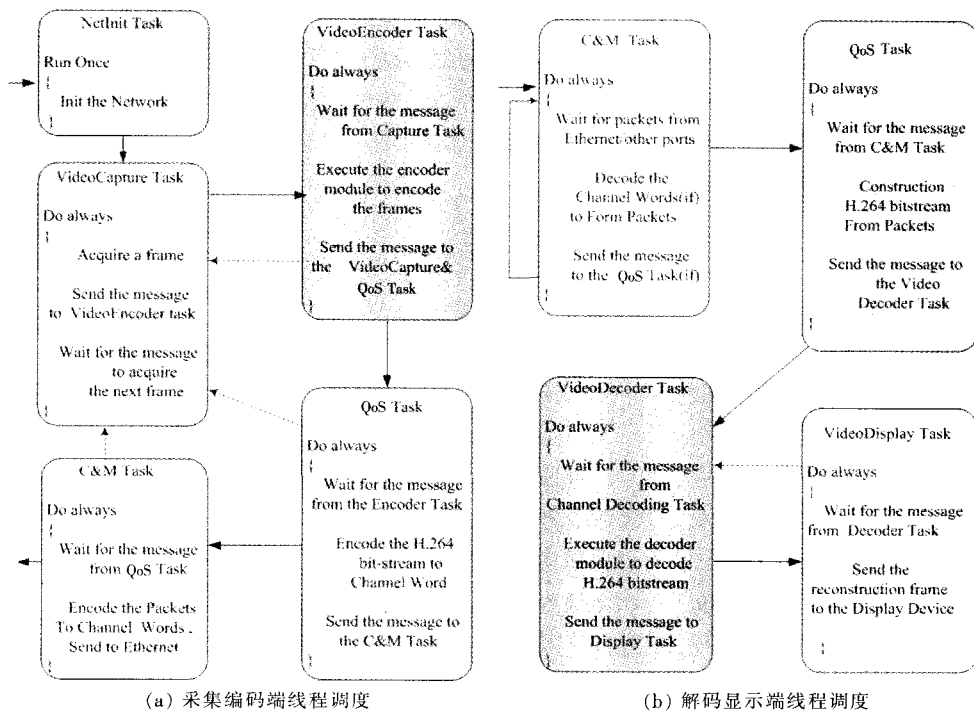


图 4 视频系统的线程调度

的 SCOM 消息后,从阻塞态进入执行态,根据 SCOM 信息提供的地址访问视频数据,并完成 H.264 编码,将编码数据放入 VideoEncoder TSK 和 QoS TSK 之间的缓冲区(如果该缓冲区未滿),并通过 MBX 消息向 QoS TSK 传递信令,同时通过 SCOM 消息通知 VideoCapture 线程编码完成;

4) QoS TSK 循环运行,在收到 VideoEncoder TSK 的 MBX 消息后开始从缓冲区中取得数据进行组包,组包后数据放入 QoS TSK 和 C&M TSK 之间的缓冲区(如果该缓冲区未滿),通过 MBX 消息通知 C&M TSK,并将 VideoEncoder TSK 和 QoS TSK 之间缓冲区状态返回至 VideoCapture TSK;

5) C&M TSK 循环运行,在收到 QoS TSK 的 MBX 消息后开始从缓冲区中取得数据进行信道编码,信道编码后数据通过网络接口进行发送,并将 QoS TSK 与 C&M TSK 之间状态报告 VideoCapture TSK;

6) 在 2 个缓冲区状态都未滿的情况下,VideoCapture TSK 采集数据,重复 2)~6)。

整个系统的速度不仅取决于每个模块的速度,更取决于模块之间能否合理地调度运行。因此,在设计线程调度时就应该合理设置线程之间的依赖关系,实现线程之间的协调工作。系统的解码显示端运算复杂度较低,而且运行逻辑相对简单,调度关系如图 4b 所示。

3 系统的改进和优化

经过以上的设计,整个系统可以在多种网络条件下

运行,在一端对数据进行采集和编码,在另一端实现视频的解码和显示,实现了系统在设计之初提出的功能要求。但是仅仅实现功能是远远不够的,系统在实现之初的运行速度是很慢的,无法显示视频的流畅采集显示,所以要通过各种方式实现系统性能的优化,以实现较好的系统性能。

为了衡量系统的运行速度,笔者对系统进行部分修改,将采集端的采集部分修改为读取标准序列文件,系统自环运行(系统同时完成编码端和解码端的所有功能),选择典型测试序列

foreman(352×288)的前 25 帧进行测试,码率为 200 kbit/s。

主要在以下几个方面进行系统优化:

1) 线程调度的调整。合理地匹配各个线程的运行速度对系统性能的提高是有很大的帮助的。例如,在系统设计之初,采集编码端的线程也是依次顺序运行的,这样的调度结构虽然简洁清晰,但是在某一线程需要等待数据时(比如采集线程获取数据时)其他线程也需要等待,这就拖慢了系统的运行速度。将调度结构按照第 2.4 节进行调整后,在某一线程等待数据的过程中,其他线程运行,这一方式提高了整个系统的运行速度。

2) 编译器优化。通过合理选择 CCS 编译器提供的编译选项提高系统的性能,性能如表 1 所示。MI·s⁻¹ 表示兆指令/秒。

表 1 编译器优化后的系统性能

比较项	系统	H.264 编码器	H.264 解码器	YUV 降采样和插值	其他
运算量/(MI·s ⁻¹)	1 884.09	983.10	464.15	145.77	291.07
所占比重/%	100	52.17	24.64	7.74	15.45

3) 空间结构的调整。例如,DM6437 提供了多种速度不同的存储空间^[2],合理地设置空间的使用方式可以提高系统的运行速度。在所有线程中,H.264 编/解码器占用了系统运算量的 75% 以上,因此在安排整个系统的空间时,着重考虑了编码器和解码器的空间需求。经过对编码器/解码器结构和数据大小的分析以及不断的时间分析,笔者选定如下的存储空间设置:L1P 存储空间全部配置

为cache,大小为32 kbyte;L1D存储空间48 kbyte配置成为内部数据存储空间,32 kbyte配置成为数据cache;L2存储空间32 kbyte配置成cache(程序cache或者数据cache),剩余96 kbyte设置为数据存储空间。在内部存储空间中主要放置编解码需要频繁使用的数据以及全局变量。经过以上的设置,编解码器相对于所有数据放置在片外的情况,速度提高了约20%,系统性能如表2所示。

表2 空间结构优化后的系统性能

比较项	系统	H.264 编码器	H.264 解码器	YUV降 采样和插值	其他
运算量/(M1·s ⁻¹)	1 422.25	683.80	334.95	143.27	260.23
所占比重/%	100	48.08	23.55	10.07	18.30

4) 汇编优化。在合理配置空间的基础上,还可以将编解码器中调用次数较多、运算量较大的函数用线性汇编实现。通过线性汇编的编写,在编解码器运行速度上还可以获得约10%的提高。

经过以上的各种优化方法,可以使系统运行在一个可以接受的速度上,实验结果如表3所示。

表3 系统主要模块运算量分析

比较项	系统	H.264 编码器	H.264 解码器	YUV降 采样和插值	其他
运算量/(M1·s ⁻¹)	1 204.28	617.60	314.09	140.80	131.79
所占比重/%	100	51.28	26.08	11.69	10.95

DM6437的主频按照600 MHz来计算,可得系统自环的运行速度为12.46 f/s(帧/秒),如果系统单独作为发送端或接收端运行,运行速度都可以基本达到25 f/s的实时性要求。

4 小结

笔者通过系统框架设计完成了系统的架构,通过线

程的设计和实现完成了系统线程的设置和调度,通过各种优化方式实现了系统运行速度的优化,最终基本实现了基于DM6437视频系统的实时运行。根据对系统实际运行数据的分析,也找到了系统运行速度的主要瓶颈,为进一步的优化工作打下了基础。接下来的工作中将进一步优化系统运行速度,完善系统功能,以实现更高分辨率、更高码率的实时通信。

参考文献:

- [1] Texas Instruments. TMS320 DSP/BIOS V5.41 user's guide[EB/OL]. [2010-01-20].<http://focus.ti.com/lit/ug/spru423h/spru423h.pdf>.
- [2] Texas Instrument. TMS320C64x DSP two level internal memory reference guide [EB/OL]. [2010-01-20].<http://focus.ti.com/lit/ug/spru610c/spru610c.pdf>
- [3] Texas Instruments. TMS320DM6437 digital media processor[EB/OL]. [2010-01-20].<http://focus.ti.com/lit/ds/sprs345d/sprs345d.pdf>.
- [4] Texas Instruments. TMS320 DSP/BIOS V5.41 user's guide[EB/OL]. [2010-01-20].<http://focus.ti.com/lit/ug/spru423h/spru423h.pdf>.
- [5] JVT of ISO/IEC and ITU-T. ITU-T recommendation H.264——ISO/IEC 14496-10 AVC.[EB/OL]. [2010-01-20].http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=52974.
- [6] Texas Instrument. TMS320C6000 DSP/BIOS 4.90 application programming Interface (API) reference guide[EB/OL]. [2010-01-20].<http://focus.ti.com/lit/ug/spru403g/spru403g.pdf>.
- [7] Texas Instruments. TMS320C6000 network developer's kit (NDK) programmer's reference guide [EB/OL]. [2010-01-20].<http://focus.ti.com/lit/ug/spru524g/spru524g.pdf>.

作者简介:

贾浩(1984-),硕士生,主研视频压缩编码与传输;

崔慧娟,女,教授,从事多媒体通信方向的研究;

唐昆,教授,博士生导师,从事多媒体通信方向的研究。

责任编辑:任健男

收稿日期:2010-04-22

(上接第29页)

参考文献:

- [1] SLEPIAN J D, WOLF J K. Noiseless coding of correlated information sources[J]. IEEE Trans. Information Theory, 1973, 19(7): 471-480.
- [2] WYNER A D, ZIV J. The rate-distortion function for source coding with side information at the decoder[J]. IEEE Trans. Information Theory, 1976, 22(1): 1-10.
- [3] GIROD B, AARON A M, RANE S, et al. Distributed video coding[J]. Proceedings of the IEEE, 2005, 93(1): 71-83.
- [4] 史萍,罗坤. 分布式视频编码中符号 Turbo 码交织器的设计[J]. 电视技术, 2008, 32(4): 35-37.
- [5] 史萍. 分布式视频编码在视频压缩和容错视频传输中的应用[J]. 中国传媒大学学报, 2007, 3(9): 39-46.

- [6] AARON A, ZHANG R, GIROD B. Wyner-Ziv coding of motion video[C]//Conference Record of the Thirty-Sixth Asilomar Conference on Signals and Systems, 2002. Pacific Grove, CA: IEEE Press, 2002: 240-244.
- [7] AARON A, RANE S, GIROD B. Wyner-Ziv video coding with hash-based motion compensation at the receiver[C]//Proc. 2004 IEEE International Conference on Image Processing. Singapore: IEEE Press, 2004: 3097-3100.

作者简介:

史萍(1964-),女,教授,主要研究领域为视频编码技术。

责任编辑:哈宏疆

收稿日期:2010-07-07