

基于 ARM11 的视频实时传输系统^①

李 岸 许雪梅 郭巧云 黄 帅 墨 芹 (中南大学 物理学院 湖南 长沙 410083)

摘 要: 本文介绍了一种 H.264 视频流实时传输系统的设计和实现方法。该系统包括服务器和客户端两大部分。服务器端以三星公司最新 ARM11 芯片 S3C6410 作为主控制器, 在嵌入式 linux 环境下, 利用 V4L2 接口技术、最新的 H.264 视频编码技术、RTP/RTCP 传输协议完成视频的实时采集、编码、传输。在客户端完成了视频实时显示与存储的应用程序开发。经测试, 该系统可达到 25fps@720×480 的画质效果, 传输视频清晰、稳定, 具有很强的实用性和广阔的应用前景。

关键词: ARM11; 视频监控; H.264; RTP

Video Streaming Real-Time Transmission System Based on ARM11

LI An, XU Xue-Mei, GUO Qiao-Yun, HUANG Shuai, MO Qin

(Institute of Physics, Central South University, Changsha 410083, China)

Abstract: This paper presents a method of design and realization of H.264 video streaming real-time transmission system, which contains two parts—server and client. Using the Samsung 's latest ARM11 chip S3C6410 as the main controller, Server adopts V4L2 interface technology, the latest H.264 video coding technology, and RTP / RTCP Real-time Transport Protocol to complete video capture, encoding, and transmission process in the embedded environment of linux. The application development of video real-time display and storage can be achieved in the client end. Experimental results show that the system can achieve 25fps @ 720 × 480 image quality, and the video is delivered clear and stable with strong practical and broad application prospects.

Keywords: ARM11; video surveillance; H.264; RTP

1 引言

随着信息技术与网络技术的发展, 人们对视频传输的效果如实时性、清晰度等, 传输设备的体积大小, 功耗节能等方面提出了更高的要求。因此开发出便携轻巧的高清视频传输设备迫在眉睫。自新一代视频编码标准 H.264^[1]颁布以来, 因其优异的压缩性能、良好的网络的亲和性、很强的抗误码能力, 获得了广泛的应用^[2]。但以往的应用主要集中在 PC 平台上, 依靠软件方式完成 H.264 的编解码功能, 从而在很多场所受到限制。

本文采用三星公司最新的 ARM11 芯片进行开发, 利用内部模块 MFC 完成 H.264 的编解码, 在嵌入式 Linux 环境下, 完成视频的采集、传输、实时显示、存储等功能。本嵌入式视频传输系统具有功耗低、画质效果好, 轻巧便携等优点, 将在家庭安防、智能交通监控、远程教育、森林防火监控、远程医疗等众多领域产生巨大影响。

2 嵌入式视频服务器的硬件构架

基于 ARM11 的视频采集与编码系统的硬件部分

① 基金项目:中南大学米塔尔创新创业项目(08MX02);中国博士后基金项目(20070420825)

收稿时间:2010-03-16;收到修改稿时间:2010-04-15

主要由主控制板模块和摄像头模块组成，其中主控制板选择以ARM1176JZF-S为内核的S3C6410微处理器作为控制器。S3C6410接口丰富，通过CameraIF接口连接摄像头，串口与PC机相连。系统整体框图如图1所示。

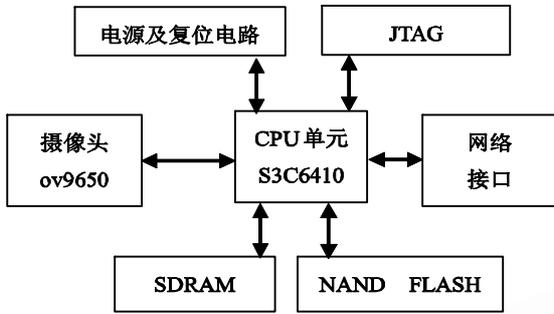


图1 系统整体框图

嵌入式系统硬件部分主要包括：微处理器及存储电路模块，电源、时钟和复位电路模块，外围接口电路模块等。本系统由S3C6410处理器控制接收摄像头的视频信号，利用S3C6410内部集成的多媒体编解码器(MFC)进行基于H.264的压缩编码[3]，实现视频数据的快速采集和高效压缩，以后将进行传输方面的研究。本系统中NAND FLASH用于存储各种固化程序，SDRAM用于系统运行时程序的存取，JTAG用于程序的下载，串口用于系统打印信息的输出及程序的调试。其中基于S3C6410微处理器的核心控制板如图2所示。



图2 嵌入式核心控制板

3 视频采集、编码和驱动实现

嵌入式视频服务器的实现，需在硬件平台上移植linux操作系统，建立好交叉开发环境，开发视频采集驱动ov9650.ko，完成视频采集、编码和传输等应

用程序。

3.1 嵌入式Linux下的视频采集

采用Linux下最新视频操作标准V4L2(Video For Linux Two)对Camera接口进行控制，并利用内存映射方式见图5，获取视频数据。由于是把设备里的内存映射到应用程序的地址空间，只是指向数据buffer的指针被交换，数据本身不用被拷贝，因此可以减少采集时间，提高视频数据的采集效率。视频采集在打开设备，初始化采集图像的大小、颜色、帧率等后，进行数据采集，具体如图3和图4所示：

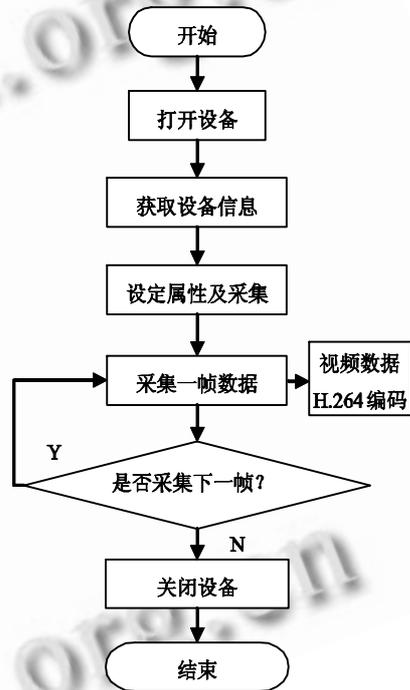


图3 视频采集流程

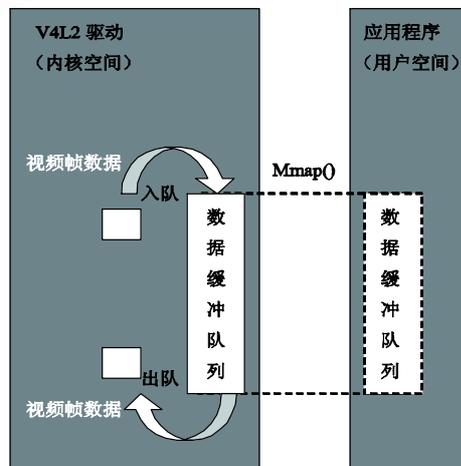


图4 内存映射方式获取视频数据示意图

3.2 H.264 视频编码

H.264 视频的编码由 S3C6410 内多媒体编解码模块 MFC 完成, MFC 由内嵌的位流处理器和视频编解码核心模块组成, 支持 MPEG4、H.263、H.264、VC1 等多种视频格式。在嵌入式服务器上, 加载了 MFC 驱动后, 可以像操作普通文件一样调用 MFC 函数对视频数据进行 H.264 编码, 系统实现时主要用到的函数有:

(1)mfc_fd=open(MFC_DEV_NAME,O_RDWR|O_NDELAY); 此函数的作用是打开 MFC 设备, 如果打开成功, 则返回 MFC 设备号

(2)ioctl(mfc_fd,IOCTL_MFC_GET_FRAM_BUF_ADDR,&get_buf_addr);

此函数的作用是/获得到 MFC 输入缓冲区地址, 获得的地址是 get_buf_addr 结构体内的指针所指地址。

in_buf = (char *)get_buf_addr.out_buf_addr;

(3)ioctl(mfc_fd,IOCTL_MFC_GET_LINE_BUF_ADDR, &get_buf_addr);

此函数的作用是获得到 MFC 输出缓冲区地址, 获得的地址为 get_buf_addr 结构体内的指针所指地址。

out_buf=(char*)get_buf_addr.out_buf_addr;

(4)read(cam_fp, in_buf, (out_width* out_height * 3/2));

此函数的作用是读取摄像头的 YUV 原始数据, 并将此数据作为 MFC 数据的输入, 如果读取成功, 则返回读取的字节数。

(5)ioctl(mfc_fd,IOCTL_MFC_H264_ENC_EXE,&enc_exe);

此函数的作用是控制 MFC 对输入缓冲区数据进行 H.264 编码, 并将压缩后数据放入输出缓冲区。

4 嵌入式编码器平台组网及视频传输的实现

嵌入式视频编码器可通过有线或无线的方式连入互联网完成视频服务器的功能, 实现视频的传输。

4.1 实时传输模型

本系统采用 ISMA2.0 传输方式^[4],通过 C/S 架构, 在应用层采用 RTSP^[5]控制协议, 传输 SDP 文件开启会话, 视频数据采用 RTP^[6]封装后承载在 UDP 上, 并通过 RTCP 协议监测传输质量。开销相对较小, 传输效率较高。



图 5 ISMA 方式的流传输基本过程

音/视数据	SDP
RTP	RTSP
UDP	TCP
IP	
以太网/无线网	

图 6 ISMA 方式的传输协议栈

4.2 分层结构及打包策略

H.264 编码后的视频数据分为视频编码层(VCL: Video Coding Layer)和网络提取层(NAL: Network Abstraction Layer)。VCL 层负责高效的数字视频压缩, 而 NAL 层负责以网络要求的方式对数据进行封装, 使得 VCL 层中数据能够有效地在各种协议的网络系统传输。同时对于 TCP/IP 四层网络协议, IP 层位于链路层之上, 在传输层如 UDP/TCP 数据做为 IP 包的负荷进行传输, RTP 数据则作为 UDP 数据包的负荷。

打包策略: 如果一个 NAL 单元大小小于网络的 MTU, 且与后一个 NAL 单元重组后大小超过 MTU, 则将它用单一打包方案进行封装, 即将此 NAL 单元作为 RTP 包的有效载荷部分, 加上 RTP 报文头部后封装成一个 RTP 包; 否则, 将后面的 NAL 单元重组, 直到重组后的 NAL 单元的大小最大且不超过 MTU, 对重组后的 NAL 单元封装成 RTP 包; 如果一个 NAL 单元大小大于网络的 MTU, 对此 NAL 单元进行分割, 然后再对分割后的 NAL 单元进行单一打包方式进行封装。

4.3 程序设计

下面给出实现该算法的实现:

```
while(FrameEndFlag!=TRUE) //
H.264 视频流未结束
{
next_start_code = find_next_start_code(readbuf); //
```

```

寻找下一个 NAL 单元头地址
if(next_start_code >0)
    { length = next_start_code - 4;
      if(length < fMaxSize) { //判断当前的 NAL 长度是否小于 IP 最大传输单元 MTU
        if(length + find_next_start_code(readbuf)-4 < fMaxSize) //判断是否组包 MTU
            { Length += find_next_start_code(readbuf)-4;
              //长度取两 NAL 单元长度之和
              STAP-A_packet_deliver(); //对组合后的 NAL 单元进行封包 }
            else simple_packet_deliver(); //将此 NAL 单元进行简单 RTP 封包;
          }
        else { //进行分包处理
          FU-A_packet_deliver(); //将此 NAL 单元进行分割封包
          Length = fMaxSize;
        }
      }
    else //未发现下一个 NAL 单元头,H.264 视频流结束;
      {FrameEndFlag = TRUE ;}
      sess.SendPacket((void*)sendbuf,length,99,true ,0); //发送封装好的 RTP 包
    }

```

5 系统测试及分析

在局域网内数台 PC 机上对系统进行测试,该系统采用前面介绍的硬件开发平台,其主频为 553MHz,系统上运行专门定制的 Linux 系统,该系统内核版本为 2.6.21,交叉编译器为 arm-linux-gcc4.1;客户端系统为 PC 机上的 XP 系统或 Linux 系统,播放器为 VLC。



原始图像



压缩后图像

图 7 视频对比截图

首先对图像编码质量进行测试,将摄像头 OV9650 采集得到的未被压缩的原始的 YUV 图像,与经过 MFC 模块 H.264 压缩编码后的图像(都转换为 YUV 格式)进行 PSNR 值计算比较。设置编码率为 2.4Mbit/s,测试帧率为 25f/s 的 VGA(720*480)视频 80s,对 1998 帧图像进行 Y-PSNR 值计算,得到 PSNR 的平均值为 35.56。对比视频截图如图 7 所示,绘出的 Y-PSNR 曲线如图 8 所示。由结果可以看出:从整个视频图像来说,PSNR 值变化稳定,视频图像压缩整体质量较高;但具体到每一帧来看,相对而言,当图像画面运动较为剧烈时,Y-PSNR 值较低;反之,画面较为稳定时,Y-PSNR 值较高。

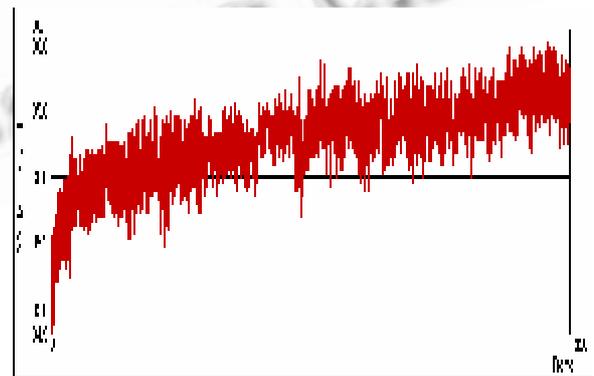


图 8 PSNR 曲线图

其次对网络传输质量进行了实验,在其它编码参数相同和保证平均 PSNR 大于 30(dB)的情况下,对不同分辨率的视频(25f/s)的码率和丢包率各进行 30min 的统计,并计算平均值得出实验数据如下表 1:

(下转第 42 页)

表 1 码率与丢包率测试表

分辨率	QCIF(176*144)	CIF(352*288)	D1(720*480)
平均码率 (kb/s)	43.21	254.37	2256.
平均丢 包率(%)	0.08	0.07	1.14

实验表明:随着分辨率的增大,平均码率相应增加,将占用更多的带宽,导致丢包率有所上升,同时在实验过程中发现,视频场景相对静止时,码率较小,场景快速变化时,码率明显上升,导致丢包率增加,本系统在传输 720*480 的视频时,平均丢包率较小,图像质量无明显下降。

6 总结

本文采用新型处理器 ARM11,设计并实现了一个 DVD 画质的视频实时传输系统。本系统在完成视频的采集、H.264 压缩编码和网络传输等视频服务功能,更具有便携、低功耗等优点。本文作者将对系统在差

错控制、差错恢复和安全传输方面做进一步改进。

参考文献

- 1 毕厚杰.新一代视频压缩编码标准——H.264/AVC.北京:人民邮电出版社,2005.
- 2 余兆明,查日勇.图像编码标准 H.264 技术.北京:人民邮电出版社,2006.
- 3 Rahul Vanam Eve A. Riskin H.264/MPEG-4 AVC Encoder Parameter Selection Algorithms for Complexity Distortion Tradeoff. IEEE Data Compression Conference, 2009.
- 4 卢官明,宗昉. IPTV 技术及应用.北京:人民邮电出版社,2007.
- 5 Baugher M, McGrew D, Naslund M, Carrara E, Norrman K. The Secure Real-time Transport Protocol (SRTP), RFC 3711, 2004.
- 6 Schulzrinne H, Casner S, Frederick R, Jacobson V. RTP: A Transport Protocol for Real-Time Applications, IETF RFC 3550, 2003.