

DOI:10.3963/j.issn.1671-4431.2011.11.029

# 基于 FFMPEG 的跨平台视频编解码研究

胡 聪<sup>1</sup>, 周 甜<sup>1</sup>, 唐璐丹<sup>2</sup>

(1. 桂林电子科技大学电子工程与自动化学院, 桂林 541004; 2. 桂林电子科技大学信息科技学院, 桂林 541004)

**摘 要:** 针对视频监控提出了视频编解码器的软件实现。监控客户端采用 S3C2440 作为嵌入式硬件平台, 并通过 CMOS 摄像头 OV9650 采集视频, 经过 FFMPEG 编码传输给监控服务器端。监控服务器端接收到视频流后通过 FFMPEG 实时解码播放, 实现远程监控。实验结果表明, 系统达到了监控客户端的视频采集、编码, 监控服务器端解码和播放。

**关键词:** FFMPEG; 编解码; 嵌入式; 视频; OV9650

中图分类号: TP 391

文献标识码: A

文章编号: 1671-4431(2011)11-0139-04

## Research on Cross-platform Video Codec Based on FFMPEG

HU Cong<sup>1</sup>, ZHOU Tian<sup>1</sup>, TANG Lu-dan<sup>2</sup>

(1. College of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China; 2. Institute of Information and Technology, Guilin University of Electronic Technology, Guilin 541004, China)

**Abstract:** A software implementation of video codec for video monitoring is presented. Monitor client used S3C2440 as embedded hardware platform, acquired video with OV9650 of CMOS camera, transmitted to the monitoring server with FFMPEG coding. Monitor server received video stream, decoded and playing real-time by FFMPEG, realized remote monitoring. The experimental results show that the system achieves the monitor client video acquisition, encoding, and monitor server decoding and playing.

**Key words:** FFMPEG; codec; embedded system; video; OV9650

随着经济的发展, 银行、交通、家居、仓储、国防等领域对安全防范、现场记录、报警系统的需求越来越大, 要求越来越高。视频监控系统主要具备远程现场的图像采集、处理、传输、实时显示及视频存储等功能。它以监控目标实时性强、适于远距离传输、能提供直接有效的现场证据等突出优点, 已在诸多领域成为人们不可或缺的工具<sup>[1]</sup>。考虑到视频监控系统必须需要对视频信号进行编解码, 讨论了一种基于 FFMPEG 的视频监控系统的编解码研究。

## 1 FFMPEG 简介

FFMPEG 是一个集录制、转换、音/视频编解码功能为一体的、完整的开源解决方案。FFMPEG 的开发基于 Linux 操作系统, 也可在大多数操作系统中编译和使用。FFMPEG 支持 MPEG、DivX、MPEG4、AC3、DV、FLV 等 40 多种编码, AVI、MPEG、OGG、ASF 等 90 多种解码<sup>[2]</sup>。

## 2 系统方案

系统采用 C/S 设计模式来实现客户端与服务器端的交互, 由服务器端向客户端发出监听请求, 客户端

收稿日期: 2011-09-17.

基金项目: 国家自然科学基金(61172053).

作者简介: 胡 聪(1981-), 男, 硕士, 讲师. E-mail: hucong@guet.edu.cn

验证并进行连接。连接成功后,客户端监测点启动摄像头进行视频数据采集,采集的视频数据是原始数据 rgb565(. rgb16) 格式,经格式转换后,再经过 FFMPEG 压缩成 H. 263 格式视频,经通信网络传输到服务器端计算机中;监控服务器端接收视频数据后,经 FFMPEG 解码后,通过 VFW(Video for Windows) 显示在服务器端界面上。视频发送端采用 SAMSUNG 公司的具有 ARM9 内核的 S3C2440 作为嵌入式微控制器,服务器端采用普通 PC 机。系统方案如图 1 所示。

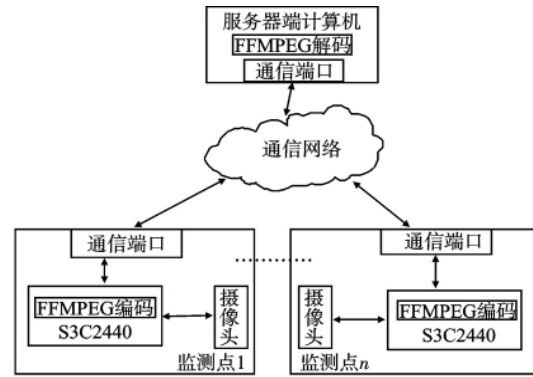


图1 系统方案

### 3 客户端

客户端的主要工作首先是搭建嵌入式 Linux 平台,然后是搭建交叉编译环境,做完这些准备工作后,就是开发摄像头驱动程序,并通过操作摄像头采集图像并经 FFMPEG 压缩,监听服务器端的请求后发送给服务器端。

#### 3.1 建立嵌入式 Linux 平台

建立嵌入式 Linux 平台主要有以下步骤:

1)使用 H-JTAG 快速烧写 BIOS 到开发板。2)格式化 Nand Flash。3)安装 bootloader。4)安装内核文件。5)安装文件系统<sup>[3]</sup>。

#### 3.2 搭建交叉编译环境

在 Linux 平台下,要编译内核、bootloader,还有其他一些应用程序,均需要交叉编译工具链。使用 arm-linux-gcc 4.3.2<sup>[3]</sup>。

#### 3.3 摄像头驱动配置

OV9650 是 OmniVision 公司生产的一系列 CMOS 摄像头中的一种,在接口上能够保持与 S3C2440 的一致性。输出图像最大为 130 万像素,输出图像格式包括 SXGA, VGA, QVGA, CIF, QCIF 等,并提供加窗功能以输出不同尺寸的图像。对于不同的输出图像格式,输出最高帧率可不同,最高可达 120 f/s。输出的 8 位数据格式包括 YUV/YCbCr(4:2:2)、GRB(4:2:2)、原始 RGB 数据 3 种<sup>[4]</sup>。

OV9650 图像传感器在 Linux 中作为字符设备来描述,其驱动程序提供给应用程序一个流控制接口。用户进程通过设备文件与硬件打交道,对设备文件的操作本质是一些系统调用。若要将系统调用和设备驱动程序关联起来,须用到 struct file\_operations 这个关键数据结构。因此,编写设备驱动的主要工作就是编写数据结构中定义的子函数,并填充 file\_operation 的各个域。

在开发阶段用 make modules 命令将驱动程序编译成模块 OV9650\_2440.ko,用 insmod 命令来加载驱动,可避免反复编译烧写内核<sup>[5]</sup>。驱动模块编译加载后,便可像操作普通文件一样对摄像头进行数据读取操作<sup>[6]</sup>。在开发完成阶段中,可以把驱动程序编译进内核,以免每次都手动加载驱动。

加载了驱动程序后,像操作普通文件一样操作摄像头。如:定义 int m\_filev4l2,通过 m\_filev4l2 = open("/dev/camera", O\_RDWR)打开摄像头,通过 read(fd, &in\_rgb565, D\_SIZE)读取摄像头的视频数据到数组 in\_rgb565 中,通过 close(m\_filev4l2)<sup>[7]</sup>。有了视频数据后,就可以通过 FFMPEG 进行编码。

#### 3.4 FFMPEG 编码

用 FFMPEG 编码时,首先对 FFMPEG 库初始化,注册所有的编解码器,配置编码器、码率、帧速率、编码像素格式和分辨率等,然后开始编码。通过设置结构体 AVOutputFormat 的成员 video\_codec 的值设置编码器,如:aofmt->video\_codec = CODEC\_ID\_H263 设置 H. 263 编码;通过设置结构体 AVCodecContext 的成员 pix\_fmt 的值设置待编码的数据像素格式,如 acc->pix\_fmt = PIX\_FMT\_YUV420P 设置 YUV420 像素格式;通过设置 acc->bit\_rate, acc->width, acc->height 等可以设置码率,宽度和高度等。编码的核心函数是 avcodec\_encode\_video。系统每采集一帧数据,就送给 avcodec\_encode\_video 函数进行编码成 H. 263 视频流。其编码流程如图 2 所示。

根据系统的要求,传输的视频流为 H. 263 格式。采集数据中,OV9560 输出的是 rgb565(. rgb16) 格式,首先需要转换成 rgb888(. rgb24) 格式,然后再转换成 yuv420(. i420) 格式,然后再由 FFMPEG 进行压缩编码成 H. 263 视频格式。其中 RGB 与 YUV 的转换公式如公式(1)<sup>[8]</sup>

$$\begin{aligned}
 Y &= 0.257R + 0.504G + 0.098B + 16 \\
 U &= -0.148R - 0.291G + 0.439B + 128 \\
 V &= 0.439R - 0.368G - 0.071B + 128
 \end{aligned}
 \quad (1)$$

### 4 服务器端

服务器端的主要工作是监控所有客户端,按照需要服务器端可以任意连接某个监控客户端。连接后接收数据,经过 FFMPEG 解码后显示在服务器端。

用 FFMPEG 解码时,首先对 FFMPEG 库初始化,注册所有的编解码器,配置解码器和解码像素格式等,然后开始解码。通过设置结构体 AVCodec 内容设置待解码数据流格式,如:AVCodec \* codec = avcodec\_find\_decoder(CODEC\_ID\_H263)设置待解码数据流格式为 H. 263;通过设置结构体 AVPicture 的值设置解码像素格式,如:avpicture\_fill((AVPicture \*) pFrameYUV, out\_buffer, PIX\_FMT\_YUV420P, D\_Width, D\_Height)设计解码像素格式为 YUV420。解码的核心函数是 avcodec\_decode\_video。其解码流程如图 3 所示。

#### 4.1 服务器端解码

由于解码的是 H. 263 视频流,因为不知道要解码的数据流长度,跟文件解码操作不一样<sup>[9]</sup>。在本设计中,专门为解码器开辟了一个线程负责解码,同时也开辟了一个 FIFO 队列进行数据的缓冲。解码时,接收的数据不停地进入 FIFO 队列,然后从头取固定的字节数进行解码。而且由于解码时解出的帧是完整的帧,这就不会出现解码后还会剩余不完整的帧字节,这些字节需要重新和后面的视频数据流拼接重新解码<sup>[10-11]</sup>。用 FFMPEG 对 H. 263 解码出来格式是 YUV(. i420) 格式,需要转换成 RGB(. rgb24) 格式显示。

#### 4.2 服务器端显示

显示视频采用的是 VFW (Video for Windows)。显示的核心函数是 DrawDibDraw 函数。DrawDibDraw 画图针对的是 DIB 图像,是一种 BMP (Bitmap) 图像。DrawDibDraw 函数有 13 个参数,最重要的参数是第 7 个和第 8 个参数,第 7 个参数是填入位图的头信息,第 8 个参数填入的是 RGB (. rgb24) 图像数据<sup>[12-13]</sup>。

由于用 DrawDibDraw 画图时,图像上下颠倒,在采用 DrawDibDraw 进行画图前,将 rgbdata 指向的缓存中的图像数据进行行的颠倒操作,首先得知道图像的宽度,假如为 Width,然后 Width \* 3 就是一行的字节数,接着第一行的数据和最后一行的数据交换,第二行的和倒数第二行的数据进行交换,依次类推。

### 5 质量效果控制

系统在中国移动 3G 通信环境下进行实验,通道的数据流需设置为 H. 263 编码格式数据流,分辨率需设为 QCIF (176 \* 144),帧率、关

键帧比例由实际情况可以自由设置,码率由实际情况拍摄的视频决定,由于传输通道带宽为 48 kb/s,所以码

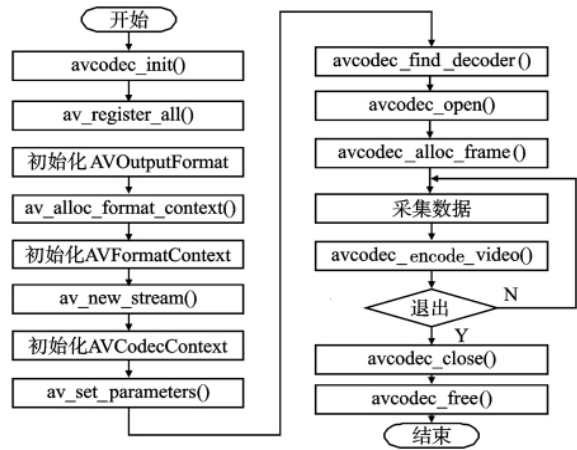


图2 客户端编码流程图

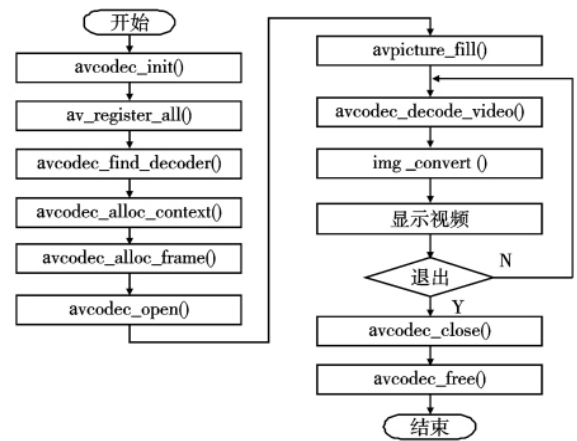


图3 服务器端解码流程图

表 1 码率

时间点	1	2	3	4	5	6	7	8	9
码率/(b · s <sup>-1</sup> )	5 914	5 728	5 968	3 155	3 227	3 502	3 557	2 919	3 000

率不能超过 48 kb/s。当设置关键帧比例为 1/6 时,帧率为 6 f/s 时,实际测得码率如表 1 所示。

如果想画面流畅些,可以增大帧率,如果想增强视频的对移动物体的显示效果,可以增大关键帧比例,但不能超过系统传输的带宽<sup>[14]</sup>。

## 6 压缩率

压缩编码采用 H. 263 压缩编码,采用关键帧比例为 1/6 时,实测压缩率表格如表 2 所示。

表 2 压缩率

时间点	1	2	3	4	5	6	7	8	9
压缩率/%	2.59	2.51	2.61	1.38	1.41	1.53	1.55	1.27	1.31

压缩率与关键帧比例、实际视频及运动等有关,关键帧比例越低、视频像素相似度越高、运动量越小,压缩率越高<sup>[15]</sup>。

## 7 结 语

随着视频压缩技术的日益成熟,嵌入式视频监控成为当今视频监控的主流。采用 ARM9 内核的 S3C2440 为嵌入式硬件平台,通过摄像头采集数据,在嵌入式 Linux 与 Windows 操作系统相结合的跨平台上,实现 FFMPEG 的编解码,对实际嵌入式视频监控系统的的设计开发,具有重要意义和实用价值<sup>[16]</sup>。

### 参考文献

- [1] 杨晓健. 基于 ARM9 的嵌入式视频采集系统设计[J]. 西安工程大学学报, 2010, 24(2): 208-212.
- [2] 蒋志峰. FFMPEG 的快速音视频开发方法[J]. 单片机与嵌入式系统应用, 2008(1): 69-71.
- [3] 师娟娟. 基于 ARM9 的嵌入式 Linux 移植[J]. 武汉理工大学学报, 2008, 30(2): 205-208.
- [4] 杨海山, 何东健. 基于 ARM 和 Linux 的视频采集系统研究与开发[J]. 微计算机信息, 2009, 25(11-2): 122-124.
- [5] 阙大顺, 杜 玮. Vivi 在 S3C2410 上的移植研究[J]. 武汉理工大学学报, 2007, 29(12): 47-50.
- [6] 熊平华, 杨建刚. IP 可视电话视频会议系统设计与实现[J]. 计算机工程与设计, 2004(4): 619-621.
- [7] 冯国进. 嵌入式 Linux 驱动程序设计从入门到精通[M]. 北京: 清华大学出版社, 2008: 102-146.
- [8] 冯永超, 罗 敏, 贺贵明. 一种快速 YUV\_ RGB 彩色空间变换方法[J]. 微型机与应用, 2002(7): 59-60.
- [9] 詹慧静. MPEG-4 编码特性剖析及应用研究[J]. 武汉理工大学学报, 2005, 27(6): 103-106.
- [10] 吕 雪. 基于网络多媒体的流媒体技术[J]. 武汉理工大学学报, 2005, 27(12): 121-125.
- [11] 黎燕霞, 李 扬, 刘奕宏, 等. 基于 S3C2440 的视频采集驱动设计[J]. 仪器仪表用户, 2009, 16(3): 98-99.
- [12] 查 婧, 刘 波, 曹剑中. 嵌入式视频采集与网络传输系统[J]. 电子器件, 2009, 32(3): 646-648.
- [13] 余兆明, 李晓飞, 陈春来. MPEG-4 标准及其应用[M]. 北京: 北京邮电大学出版社, 2002.
- [14] 曹少坤. 一种嵌入式网络摄像机的设计[J]. 微计算机应用, 2008, 29(10): 69-73.
- [15] 张学武, 杨学星, 江 冰. 基于 H. 263 的视频编码、解码的研究及软件实现[J]. 计算机工程与设计, 2005, 26(9): 2491-2493.
- [16] 郑旭东, 张培仁, 高修峰, 等. 嵌入式网络视频监控系统[J]. 仪表技术与传感器, 2006(08): 24-26.

(上接第 119 页)

- [15] Jennifer Read, Tricia Wevill, Tim Fletcher, et al. Variation Among Plant Species in Pollutant Removal from Stormwater in Biofiltration Systems[J]. Water Research, 2008, 42: 893-902.
- [16] Bratieres K, Fletcher T D, Deletic A, et al. Nutrient and Sediment Removal by Stormwater Biofilters: A Large-scale Design Optimisation Study[J]. Water Research, 2008, 42: 3930-3940.
- [17] Hatt B E, Fletcher T D, Deletic A. Pollutant Removal Performance of Field-scale Stormwater Biofiltration Systems[J]. Water Science and Technology, 2009, 59(8): 1567-1576.
- [18] Hatt B E, Fletcher T D, Deletic A. Hydrologic and Pollutant Removal Performance of Biofiltration Systems at the Field Scale[J]. Journal of Hydrology, 2009, 365(3-4): 310-321.