

·信号与信息处理·

基于嵌入式系统的图像采集与压缩

姜 晖¹, 梁俊明², 任高星¹

(1. 合肥电子工程学院, 安徽 合肥 230037; 2. 61906 部队, 河北 廊坊 065001)

摘 要: 首先简要介绍了嵌入式监控系统开发平台, 接着阐述了在该平台上如何实现图像的采集与压缩, 重点讲述了 Video4Linux 编程实现和 JPEG 压缩算法, 最后讲解了如何在 Linux 系统下实现 JPEG 压缩, 实验结果令人满意。

关键词: 嵌入式 Linux; 图像采集; V4L; JPEG 压缩

中图分类号: TP311.1; TN911.73

文献标识码: A

文章编号: 1673-1255(2009)01-0070-03

Image Gathering and Compressing Based on Embedded System

JIANG Hui¹, LIANG Jun-ming², REN Gao-xing¹

(1. Hefei Electronic Engineering Institute, Hefei 230037, China; 2. 61906 Unit Army of PLA, Langfang 065001, China)

Abstract: The platform of the embedded monitoring system is introduced at first. Then how to realize image gathering and compression on the platform is described. The implementation of Video4 Linux program and the JPEG compression algorithm are emphatically expounded. Finally, JPEG compression is completed in Linux system, and the results are satisfied.

Key words: embedded Linux; image gather; Video4Linux; JPEG compression

目前, 图像采集系统正从模拟采集向数字采集发展, 其中嵌入式图像采集因其优越的性能而备受青睐。采用以 OV511 为控制芯片的 COMS 图像传感器对法律允许范围内的设施、家居和办公场所进行实时监控。由于采集到的图像需要远程传输, 而原始图像数据量大, 因此对图像进行了压缩, 使得在保证图像质量的情况下, 尽可能的提高压缩比, 以实现远程监控的功能。

1 构建开发平台

硬件平台采用 Samsung 公司的处理器 S3C2410^[1]。通过在 USB 接口上外接一个摄像头, 将采集到的视频图像数据放入输入缓冲区中, 通过应用程序对图像数据进行处理; 软件系统采用 Linux 操作系统, 它是源码开放的嵌入式操作系统, 可

移植性强, 支持多任务, 具有完备的 TCP/IP 协议栈并支持多种文件系统。

根据系统应用需求定制 Linux 嵌入式操作系统, 使其支持 USB 总线和 COMS 图像传感器, 确保该 OV511 接口芯片在 Linux 下正常驱动, 并得到主机系统 USB 控制器以及 Linux 内核中 Video4Linux 的支持; 在宿主机上安装适合于目标处理器的交叉编译器、调试器。

2 图像采集

图像采集分 2 步实现: 一是 COMS 图像传感器在内核中的底层驱动; 二是编写上层应用程序以获取视频流数据。

2.1 图像传感器驱动

(1) 在 arm linux 的 kernel 目录下 make menuconfig;

收稿日期: 2008-12-09

作者简介: 姜晖(1964-), 男, 副教授, 主要研究方向为嵌入式技术与图像处理系统; 梁俊明(1983-), 男, 内蒙古乌前旗人, 助理工程师, 主要研究方向为嵌入式系统; 任高星(1982-), 男, 陕西合阳人, 硕士研究生, 主要研究方向为嵌入式系统与图像处理。

(2) 选择 Multimedia device - > 下的 Video for linux. 加载 video4linux 模块, 为视频采集设备提供编程接口;

(3) 在 usb support - > 目录下选择 support for usb 和 usb camera ov511 support. 这使得在内核中加入了采用 OV511 接口芯片的 COMS 图像传感器的驱动支持.

(4) 保存配置退出.

(5) make dep; make zImage 就生成了带有 ov511 驱动的内核.

经编译后的 Linux 支持有 USB 接口的 OV511 图像传感器, 系统启动后能够自动识别 USB 设备 (在设备目录下为 /dev/video0). 可直接利用系统调用获取设备信息和数据, 以方便应用程序的开发; COMS 图像传感器驱动程序中提供基本的 I/O 操作接口函数 open、read、write、close, 以实现中断处理、内存映射以及对 I/O 通道的接口函数 ioctl 进行控制等功能.

2.2 Video4Linux 编程

Video4Linux 是 Linux 中针对视频设备应用程序编程所提供的一系列接口函数, 通过使用 video4linux 的 API 函数从视频设备中把图像数据读取出来. 具体过程如下:

(1) 打开视频

```
int v4l_open(char * dev, v4l_device * vd)
{
    if (! dev)
        dev = "/dev/video0";
    if ((vd->fd = open(dev, O_RDWR)) < 0)
    {
        perror("v4l_open:");
        return -1;
    }
    if (v4l_get_capability(vd))
        return -1;
    if (v4l_get_picture(vd))
        return -1;
    return 0;
}
```

(2) 读 video_capability 中信息

```
int v4l_get_capability(v4l_device * vd)
{
    if (ioctl(vd->fd, VIDIOCGCAP, &(vd->capability)) < 0)
    {
        perror("v4l_get_capability:");
        return -1;
    }
}
```

```
}
return 0;
}
```

该函数成功返回后将这些信息从内核空间拷贝到用户程序空间的 vd->capability 各成员分量中, 使用 printf 函数就可得到各成员分量信息. 例如 printf("maxheight = %d", vd->capability.maxheight) 获得最大垂直分辨率的大小.

(3) 读 video_picture 中信息

```
int v4l_get_picture(v4l_device * vd)
{
    if (ioctl(vd->fd, VIDIOCGPICT, &(vd->picture)) < 0) {
        perror("v4l_get_picture:");
        return -1;
    }
    return 0;
}
```

在用户空间可以改变 video_picture 中分量的值, 先给分量赋新值, 再调用 VIDIOCSPICTioctl 函数, 例如

```
vd->picture.palette = pal; //调色板
vd->picture.depth = dep; //像素深度
if (ioctl(vd->fd, VIDIOCSPICT, &(vd->picture)) < 0)
{
    perror("v4l_set_palette");
    return (NULL);
}
```

2.3 图像数据采集

视频图像截取方法有 2 种^[2]: (1) read() 直接读取, (2) mmap() 内存映射. 设计中采用了 mmap() 内存映射方式获取图像, mmap() 系统调用使得进程之间通过映射同一个普通文件实现共享内存. 普通文件被映射到进程地址空间后, 进程可以向访问普通内存一样对文件进行访问, 不必再调用 read(), write() 等操作.

使用函数 ioctl(vd->fd, VIDIOCGMBUF, &(vd->mbuf)) 获得图像传感器存储缓冲区的帧信息, 用 vd->map = mmap(0, mbuf.size, PROT_READ|PROT_WRITE, MAP_SHARED, vd->fd, 0) 把图像传感器对应的设备文件映射到内存区, 该映射内容区可读可写且不同进程间可共享. 该函数成功时返回图像数据的指针, 失败时返回值为 -1.

实现单帧采集: 设置 vd->mbuf.frame = 0, 即采集其中的第一帧, 使用 ioctl(vd->fd, VIDIOCMCAPTURE, &vd->mbuf) 函数, 若调用成功则以非

阻塞方式激活设备真正开始一帧图像的截取. 通过 `ioctl(vd->fd, VIDIOCSYNC, &frame)` 函数判断该帧图像是否截取完毕, 如成功返回表示截取完毕, 之后可把图像数据保存成文件的形式.

3 图像压缩

采集到的原始图像数据量大, 不利于在网络上传输, 这就要求图像采集平台对原始数据进行压缩处理, 这里采用 JPEG 压缩方式. JPEG 静止图像压缩标准是一种广泛认可的标准, 一般操作系统和应用都支持. JPEG 标准是在变换编码的基础上, 综合应用了 DCT 和哈夫曼编码两种手段达到了很好的图像压缩效果.

3.1 压缩算法

在 JPEG 算法中通常是先将图像分成一个个 8×8 的图像子块, 对每一个图像子块进行离散余弦变换, 所使用的二维离散余弦变换公式如下^[3]

$$F(u, v) = \frac{1}{4} k(u) k(v) \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

其逆变换如下

$$f(x, y) = \frac{1}{4} k(u) k(v) \sum_{x=0}^7 \sum_{y=0}^7 F(u, v) \cos \frac{(2x+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

式中, 当 $u = v = 0$ 时, $k(u) = k(v) = 1/\sqrt{2}$; 当 u, v 为其他值时, $k(u) = k(v) = 1$. $f(x, y)$ 经 DCT 变换后, $F(0, 0)$ 是直流系数, 其他为交流系数.

在离散余弦变换中, 设法使变换后的图像子块的能量下降速度加快, 则可以减少块效应并能使细节部分清晰. 正是基于这一点, 通过收敛性良好的 Chebychev 多项式来拟合离散余弦变换, 使得变换后的图像子块的能量下降速度加快, 从而提高图像数据的压缩比, 同时还能够改善图像的块效应^[4].

3.2 安装 jpeg 库

ARMLinux 中没有 jpeg 函数库, 经下载 `jpegsrc.v6b.tar.gz`^[5] 后, 在 `/usr/local/arm/3.2/` 解压安装:

```
# tar -zxvf jpegsrc.v6b.tar.gz
# cd jpeg-6b
# export CC=arm-linux-gcc
# ./configure --prefix=/usr/local/arm/3.2/arm-
```

```
linux/ --enable-shared --enable-static
# make
安装前需要在 arm-linux 下建个目录
# mkdir -p /usr/local/arm/3.2/arm-linux/man/man1
# make install
```

这样, ARMLinux 就支持 jpeg 函数库, 此处注意用编译器编译连接时带有 `-ljpeg` 选项才能生效.

4 实验结果

图 1 是同一背景 2 张大小均为 320×240 的照片. 其中图 1a 是未压缩图像, 数据量为 255 K. 图 1b 是在 JPEG 压缩质量为 80 时存储的图片, 其数据量为 9.52 K 字节. 结果表明图像的数据量大大减少, 压缩效果良好.

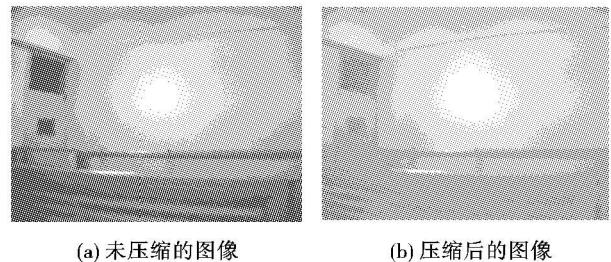


图 1 图像压缩前后的比较

5 结束语

基于嵌入式 Linux 系统介绍了图像的采集与压缩. 对于 32 位的嵌入式系统, 在应用于图像采集, 特别是远程的图像采集时, 因处理器足以完成复杂的运算, 可以使用 JPEG 压缩算法, 以获取更高的图像压缩比, 从而提高图像数据在网络上的传输速度.

参考文献

- [1] Samsung Electronics Co. Ltd. S3C2410A user's manual. revision 1.0 [EB/OL]. <http://www.samsung.com>. Yon-gin-City, Gyeonggi-Do, Korea [2004-03].
- [2] 李侃, 廖启征. 基于 S3C2410 平台与嵌入式 Linux 的图像采集应用[J]. 微计算机信息, 2006, 22(2-3): 125-127.
- [3] 杨济安. 图像通信与图像处理[M]. 重庆: 重庆大学出版社, 2003: 73.
- [4] 苏日建, 宋胜利. 嵌入式图像采集系统的 JPEG 算法改进[J]. 重庆工学院学报, 2006, 20(11): 75-76.
- [5] <ftp://ftp.uu.net/graphics/jpeg/jpegsrc.v6b.tar.gz>