文章编号:1671-4598(2014)05-1523-04　　中图分类号:TP302.1; TP301.6　　文献标识码:A

基于 Linux 的嵌入式实时视频跟踪系统

史　涛，裴海龙
（华南理工大学 自动化科学与工程学院，广州　510640）

摘要：将 ARM 与 DSP 相结合，设计并实现了一套嵌入式实时视频传输跟踪系统；该系统以集成 ARM 和 DSP 双内核的 OMAP3730 作为核心处理器，首先通过 ARM 上搭载的 Linux 操作系统实现基于 V4L2 的图像采集功能，然后调用 DSP 完成 H.264 视频编码，压缩后的图像由基于实时流传输协议的流媒体服务器传输至远程计算机端解码并显示，最后利用 camshift 跟踪算法实现对运动目标物体的实时跟踪。该系统是小型无人机自主导航与控制的视觉导航以及以无人机为平台的运动目标检测的基础，有很好的应用前景。

关键词：Linux；嵌入式；图像采集；H.264；Camshift 算法

Real－time Video Transmission System Based on Linux

Shi Tao, Pei Hailong
(College of Automation Science and Engineering, South China University of Technology, Guangzhou　510641, China)

Abstract: The research and implementation of a video transmission system based on ARM and DSP are introduced. The system use chip OMAP3730 which is a dual－core processor as its CPU. The program of capturing pictures run on Linux for ARM is based on V4L2. Then, the pictures are encoded to the H.264 format by DSP. The encoded images are transferred to the remote computer by the stream media server based on RTSP, then use the Camshift tracking algorithm for the real－time tracking of moving target object. The system can be used as the basement of visual navigation and moving target detection, where small UAV is applied as the implementation platform, which has broad application prospects.

Key words: Linux; embedded system; image capturing; H.264; Camshift algorithm

0　引言

实时视频传输技术在人们生活和工作中的应用越来越多，在对成本、空间、功耗和体积要求较为严格的工程应用中，嵌入式平台因小巧、软硬件可裁剪具有不可替代的优点，这使得以嵌入式为平台的多媒体系统成为当今工程实现及学术研究的热点之一。但视频信息具有信息量大、对计算资源需求高的特点[1-2]，导致嵌入式实时视频往往图像分辨率低，帧率小，应用效果较差。因此必须对采集到的原始视频信息进行压缩，同时又要考虑嵌入式设备计算资源的约束。本文为解决上述问题，采用了 ARM 和 DSP 双内核完成视频采集和基于 H.264 的视频压缩，大大加快了图像压缩速度，同时减轻了 ARM 的计算负担。系统基于实时流传输协议（即 RTSP，Real Time Streaming Protocol）搭建流媒体服务器，将压缩后的视频有效传输至远程计算机上解码显示，最后利用 camshift 跟踪算法实现对运动物体的跟踪。本系统目前已经用于实验室无人机监控平台，取得了很好的实验效果。

1　系统硬件平台

1.1　硬件组成

根据系统目标要求，本系统的硬件平台主要由嵌入式核心板、底板、USB 彩色摄像头和远程计算机组成。嵌入式核心板选用以色列 Compulab 公司生产的 CM－T3730。该核心板体积小，功耗低，处理器集成了 720 MHz 的 Cortex－A8 ARM 内核和 520 MHz C64x＋ DSP 内核，同时核心板还有 NAND Flash、WIFI 模块等，图形计算能力强，支持运行 Linux、WinCE 操作系统，并且具有硬件可扩展结构。系统中采用实验室自行开发的 SB－T35 作为该核心板的底板，在无需外接电路的情况下，已经内置了 USB 主从控制接口可以直接连接 USB 设备。经测试核心板在上述底板上运行稳定可靠，很好的满足了系统开发和运行的需求。由于 Linux 内核中包含 UVC（USB video class）模块及 USB 设备驱动，可使用 V4L2（Video for Linux Version 2）对摄像头进行操作，因此本系统摄像头设备选用支持 UVC 的即插即用 USB 摄像头即可。这里我们选用罗技的 USB 网络摄像头 Pro9000，其最高支持分辨率为 720P 视频采集，且支持自动对焦，具有较好的成像效果。系统的远程计算机装有 Ubuntu 操作系统，用于调试开发工作及实时视频的解码显示并对运动目标进行跟踪处理。

本视频系统的主要工作（采集、通信、传输）都在核心板 CM-T3730 上完成，处理后的视频流通过 WIFI 网络即可传送给远程计算机。

2　系统软件平台

在视频实时传输过程中，嵌入式系统首先打开摄像头设备并配置摄像头参数，然后创建采集压缩函数供 RTSP 流媒体服务器循环调用。在采集压缩函数中，采集得到的原始视频帧放到采集传输进程和调用 DSP 压缩的进程共享的内存空间中，经 DSP 压缩后再次放回共享内存区，然后被 RTSP 流媒体服务器放入发送缓冲区。RTSP 流媒体服务器会为该视频流创建网络访问地址，供远程计算机访问。在远程计算机端，使用具备 H.264 解码能力的播放器访问视频流网络地址即可实时观看远程摄像头采集到的信息并进行目标物体的跟踪。系统整体

收稿日期:2013-12-25；　**修回日期**:2014-01-26。
基金项目:教育部科技创新工程重大项目培育资金项目(708069)。
作者简介:史　涛(1989-)，女，湖南长沙人，硕士研究生，主要从事嵌入式系统、图像处理方向的研究。

流程如图1所示。

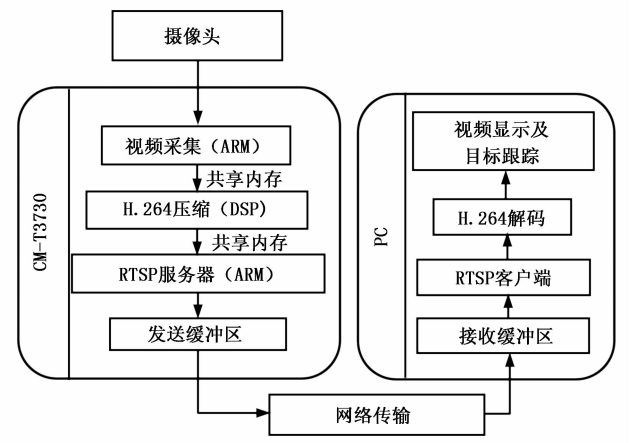

图1 系统软件流程图

2.1 网络文件系统

在嵌入式Linux的开发过程中，开发者需要在Linux服务器上进行所有的软件开发，交叉编译后，通用FTP方式将可执行文件下载到嵌入式系统运行，但这种方式不但效率低下，且无法实现在线的调试。因此，可以通过建立NFS，把Linux服务器上的特定分区共享到待调试的嵌入式目标系统上，就可以直接在嵌入式目标系统上操作Linux服务器，同时可以在线对程序进行调试和修改，大大地方便了软件的开发。

系统加载Linux内核之后，就会挂载文件系统到根目录上。使用网络文件系统，可以方便的实现主机和目标板之间文件和数据的交换。从客户端看来，使用NFS的远端文件就像是使用本地文件一样，避免了反复的进行烧写镜像文件。对嵌入式Linux系统来说，NFS服务可以让不同的主机通过网络将远端的NFS服务器共享出来的文件安装到自己的系统中。

为了构建嵌入式多媒体终端的网络文件系统，需要将开发板资料中提供的根文件系统复制到主机服务器Linux下的/home/files文件目录下，配置和开启主机服务器上的NFS服务，并修改嵌入式多媒体终端的Linux内核启动参数。

网络文件系统服务器端的配置方法如下，修改/etc/exports配置文件，添加如下内容：/home/filesys * （rw，no_root_squash，no_all_squash，sync）。

服务器网络文件服务开启的方法如下：

```
Host #/usr/sbin/exportfs -a
Host #/sbin/service nfs restart
```

为了使Linux内核在启动完成后能成功挂载网络文件系统，还需要修改内核的启动参数。

在上面介绍的Linux内核烧写中设置的环境变量即包含了网络文件系统的启动参数。其中root=/dev/nfs即设置根文件系统设备节点为网络文件系统设备节点。nfsroot=（nfshost）:(rootpath)，其中setenv nfshost 192.168.1.3，setenv rootpath /home/filesys，组合后即为nfsroot=192.168.L3：/home/filesys，该路径是文件系统挂载的位置，192.168.1.3是Linux服务器的IP地址，在内核启动挂载前，应该保证系统能够访问到服务器，且最好在同一个局域网内。

在内核启动结束后，系统将根据内核启动参数挂载到网络文件系统上。基于以上工作，就实现了基于CM－T3730处理器ARM核上的嵌入式Linux系统构建。

2.2 基于V4L2的图像采集

为使本系统外接摄像头可方便更换以满足不同环境的需求，同时使代码易于移植，系统采用基于V4L2（video for linux version 2）的图像采集程序。V4L2是Linux内核中专门用于视频设备的驱动，它为程序员提供了访问视频设备的通用接口[3-4]，包括视频设备的打开、设备属性的获取与设置、图像的获取等。当系统需要更换摄像头时，只需要编译相应设备的驱动模块添加到Linux内核即可而不需要修改系统程序，这样大大方便了系统的开发与扩展。基于V4L2的图像采集程序流程[5]如图2所示，具体步骤如下文所述。

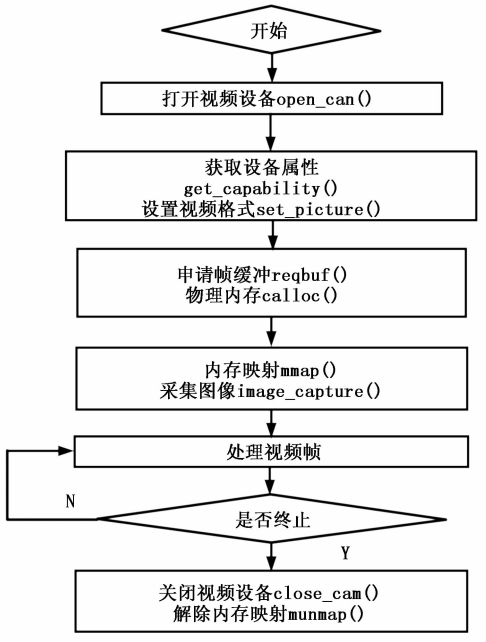

图2 图像采集流程图

（1）打开设备文件。int fd＝open（“/dev/video0”，O_RDWR）；Linux操作系统中，由V4L2驱动的视频设备节点文件通常是/dev/videoX。

（2）取得设备属性，ioctl（fd，VIDIOC_QUERYCAP，&cap），变量cap为struct v4l2_capability类型；设置视频格式，ioctl（fd，VIDIOC_S_FMT，&fmt），变量fmt为struct v4l2_format类型。

（3）向驱动申请帧缓冲，ioctl（fd，VIDIOC_REQBUFS，&reqbuf），变量reqbuf为struct v4l2_requestbuffers类型；申请物理内存（内存映射），调用calloc函数向用户空间申请相同数目的帧缓冲，并使用mmap内存映射方式将帧缓冲映射到用户空间。

（4）将申请的帧缓冲放入用于存放采集到视频帧的队列，ioctl（fd，VIDIOC_QBUF，&buf）。然后调用ioctl（fd，VIDIOC_STREAMON，&type），type是enum v4l2_buf_type枚举类型。

（5）出队列以取得采集到的视频帧，ioctl（fd，VIDIOC_DQBUF，&buf）。如果需要循环，则将buf重新放入队列并反复获取帧缓冲中的数据。

2.3 基于DSP的H.264压缩

H.264视频压缩技术是现在世界上最新的图像编解码标准，

它以低码率、高清晰的特点能够持续提供高质量的视频[6]。H.264 特别注重视频帧的压缩效率。H.264 标准的主要特性包括压缩效率，传输效率并且关注视频压缩的一些普遍应用。在本系统选择编码器时，调用 FFMPEG 软件中的 MPEG4 和 H.264 进行了压缩率的对比。在相同的帧间和帧内预测选择参数情况下，H.264 压缩率是普通 MPEG4 的 4 倍左右，同时视频图像更加稳定。因此系统选用 H.264 作为视频压缩算法。

尽管与其他编码器相比，H.264 的压缩效率非常突出，但带来的问题是其算法复杂度也大大提高[7]，对系统的计算资源提出了更高的要求。本文在选用 DSP 实现 H.264 图像压缩前，首先在 ARM 平台运行的 Linux 操作系统上通过调用交叉编译的 FFMPEG 和 X264 库实现 H.264 图像编码。经实验观察，在压缩分辨率仅为 160 * 120 大小的视频流时，平均每帧仍需耗时 90～105 ms，视频实时性很差。而且压缩任务占用了大量 ARM 运算资源影响了系统其它程序的运行，因此寻求实现快速的 H.264 压缩方法是实时视频处理的关键因素之一。DSP 具有强大的数字信号处理能力，在一个 DSP 上可以同时进行多路音视频信号的压缩处理。而且 DSP 产品成熟，开发周期短，可以实现快速技术更新和产品换代。所以本系统使用 TI 公司 OMAP3730 处理器中的 DSP 核实现视频的 H.264 压缩，这样大大节约了视频压缩时间，并减轻了 ARM 核的运算压力。

2.4　流媒体服务器搭建

系统使用开源的 LIVE555 项目实现 RTSP 流媒体服务器的搭建。利用 LIVE555 搭建 RTSP 流媒体服务器主要过程是：

（1）初始化本地环境，建立 RTP 和 RTCP 套接字，将 RTP 套接字与本机绑定，建立 sink 接收器并与该 RTP 套接字关联。

（2）建立于该 RTP 套接字、本机环境、sink 接收器配套的流控 RTCP 实例。

（3）建立 RTSP 服务器（包含与 RTP/RTCP 关联的会话）。输出该 RTSP 服务器的网络地址（本机地址加流名称）。至此，向外发数据（RTP/RTCP 远程通信）所需的准备已经完成。

（4）建立图像数据传递：根据 WebCamDeviceSource 建立 VideoSource。WebCamDeviceSource 初始化时，就使 Webcam 负责摄像头图像采集、与 DSP 压缩程序通信并将压缩图像存入 fTO 指向的缓冲区。VideoSource 初始化时就与 WebCam 关联了，因此也关联了 fTO 指向的缓冲区，这样图像经采集压缩后传入 VideoSource。

（5）sink 调用 startPlaying（），开始播放流，sink 就循环的接受经采集压缩后的数据。

（6）事件循环，当 RTSP 客户端输入访问地址 rtsp：//202.38.214.22：8554/testStream 连接服务器时，服务器就开始对客户端对话进行处理。

2.5　camshift 跟踪算法

CamShift 算法的全称是“Continuously Adaptive Mean－SHIFT”，即：连续自适应的 MeanShift 算法。其基本思想是对视频序列的所有图像帧都作 MeanShift 运算，并将上一帧的结果（即搜索窗口的中心位置和窗口大小）作为下一帧 MeanShift 算法的搜索窗口的初始值，如此迭代下去。简单点说，meanShift 是针对单张图片寻找最优迭代结果，而 camShift 则是针对视频序列来处理，并对该序列中的每一帧图片都调用 meanShift 来寻找最优迭代结果。正是由于 camShift 针对一个视频序列进行处理，从而保证其可以不断调整窗口的大小，如此一来，当目标的大小发生变化的时候，该算法就可以自适应地调整目标区域继续跟踪。

在 OpenCV 自带的 camShift 的例子当中，是通过计算目标在 HSV 空间下的 H 分量直方图，通过直方图反向投影得到目标像素的概率分布，然后通过调用 OpenCV 的 CAMSHIFT 算法，自动跟踪并调整目标窗口的中心位置与大小。该算法对于简单背景下的单目标跟踪效果较好，但如果被跟踪目标与背景颜色或周围其它目标颜色比较接近，则跟踪效果较差。另外，由于采用颜色特征，所以它对被跟踪目标的形状变化有一定的抵抗能力。

OpenCV 自带例子中的 camShift 算法，可以分为 3 个部分：

1）计算色彩投影图（反向投影）：

（1）截取跟踪目标矩阵 ***rect***。为了减少光照变化对目标跟踪的影响，首先将图像从 RGB 颜色空间转换到 HSV 颜色空间；

（2）求取跟踪目标的加权直方图 hist1。对 H 分量进行直方图统计，直方图代表了不同 H 分量取值出现的概率，或者说可以据此查找出 H 分量的大小为 x 时的概率或像素个数，即得到颜色概率查找表；

（3）将图像中每个像素的值用其颜色出现的概率进行替换，由此得到颜色概率分布图；

以上 3 个步骤称之为反向投影，需要提醒的是，颜色概率分布图是一个灰度图像；

2）meanShift 寻优

读取视频序列中的一帧，先随机取一块与 rect 等大的矩形，计算加权直方图 hist2。meanShift 算法是一种非参数概率密度估计方法，它通过不断迭代计算得到最优搜索窗口的位置和大小。

3）camShift 跟踪算法

计算两者比重函数，如果后者差距过大，更新新的矩阵中心 Y，进行迭代（MeanShift 是一种变步长可以迅速接近概率密度峰值的方法），直至一定条件后停止。camShift 其实就是在视频序列的每一帧当中都运用 meanShift，并将上一帧的 meanShift 结果作为下一帧的初始值，如此不断循环迭代，就可以实现目标的跟踪了。

本系统是实现跟踪摄像头中目标物体，目标物体初始位置用鼠标指出，矩形区域确定跟踪目标，跟踪的时候以椭圆为代表目标，其跟踪窗口大小和方向随着目标物体的变化而变化。

3　系统测试

以上对系统的实现进行了详细的说明，接下来就对该系统进行测试。本次测试系统使用开源的 VLC 播放器访问嵌入式流媒体服务器地址，该播放器支持大量的文件格式和流媒体播放格式，可安装在 Windows、Linux、Android 平台。在局域网环境下，本文对系统的性能做了进一步测试。本系统播放终端界面如图 3 所示，远程摄像头的视频信息成功地得到了解码与播放。通过以移动的人脸为跟踪目标，观察系统跟踪时的稳定性和实时性，得到效果如图所示结果。结果表明，本系统对于纯色物体在黑白背景下的跟踪效果是很好

的，但是如果背景的颜色与目标相近，或者目标附近有与目标的色调相近的算法比较物体，则Camshift会自动将其包括在内，导致跟踪窗口扩大，甚至有时会将跟踪窗口扩大到整个视频框架。

图3 实时视频播放界面

4 结束语

本系统利用具有ARM、DSP双内核的处理器搭建了图像处理能力极强的硬件平台，并采用H.264作为视频压缩算法，使得实时视频传输应用中视频压缩时间大大缩短。系统基于RTSP流媒体协议搭建了流媒体服务器，使得视频传输稳定性有很大提高。camshift能有效解决目标变形和遮挡的问题，对系统资源要求不高，时间复杂度低，在简单背景下能取得良好的跟踪效果。但当背景较为复杂，或者有许多与目标颜色相似像素干扰的情况下，会导致跟踪失败。因为它单纯的考虑颜色直方图，忽略了目标的空间分布特性，所以这种情况下需加入对跟踪目标的预测算法。目前，我们已经基于本系统实现了火点的识别及跟踪并搭载到实验室小型无人机上。该系统的性能满足了监控、视频会议等应用的要求，但系统在视频高分辨率情况下的传输稳定性仍存在不足，需要进一步优化以满足高分辨率要求下的应用场合。

参考文献：

[1] Ren Z K，Liu M H，Ye C，et al. The Real Time Video Transmission System Based on H.264 [A]. International Conference on Web Information Systems and Mining [C]，Shanghai：IEEE Press，2009，63：270－274.

[2] Ren Z K，Wei Z Q. Design and realization of video real time video transmission system [J]. Computer Engineering and Application，2007，28 (11)：2607－2610.

[3] Guo J，Zhao J. Image Capture and Display of Embedded Linux [J]. Modern Electronics Technique，2006，7：129－131.

[4] Corbet J，Robini A. Linux Device Drivers [M]. Beijing：China Electric Power Press，2006.

[5] Yan X Z，Chen Y. Design of image acquisition and transmission based on embedded ARM [J]. Foreign Electronic Measurement Technology，2009，11：57－59，80.

[6] Li H J. H.264 video compression based on network video monitoring system technical design [J]. Hebei Journal of Industrial and Technology，2011，28 (4)：236－239.

[7] Wu Q. Video Encoding and Decoding Based on H.264 and Its Realization and Optimization on DSP [J]. Modern Electronics Technique，2010，4：55－57.

[8] Richard W. Stevens，Stephen A. Rago. Advanced Programing in the UNIX Environment (Second Edition) [M]. Beijing：Post and Telecom Press，2006.

[9] Gao J S，Chen Y W，Li L L. Video Order Program System Design Baed on RTSP [J]. Chinese Journal of Electron Devices，2006，29 (4)：1143－1146.

[10] Chen L，Pei H L. Programming for the Server of Real Time Based on RTSP [J]. Microcomputer Information，2009，28 (5－3)：65－67.

(上接第1515页)

5 总结

本文研究了基于钝头体推导的FADS空气动力学模型在锥头体中的适用性问题。针对该动力学模型解算锥头体大气数据存在模型误差的问题，本文提出了一种改进的校准算法，有效提高了算法精度和工程实现性。

参考文献：

[1] Srivastava A，Meade A J，Mokhtarzadeh A A. A Hybrid Data－Model Fusion Approach to Calibrate a Flush Air Data Sensing System [R]. AIAA 2010－3347，2010

[2] 方习高，陆玉平. 嵌入式大气数据传感系统求解算法研究 [J]. 计算机测量与控制，2008，16 (3)：398－400.

[3] Zheng C J，Lu Y P，He Z. Improved Algorithms for Flush Airdata Sensing System [J]. Chinese Journal of Aeronatics，2006，9 (4)：334－339.

[4] 郑守铎，陆宇平，叶 玮. 基于χ^2检验的FADS系统故障检测与管理技术研究 [J]. 计算机测量与控制，2007，15 (11)：1449－1551.

[5] 叶 玮，郑守铎，温瑞珩. FADS/INS组合法迎角、侧滑角测量方法研究 [J]. 飞机设计，2007，27 (6)，14－17.

[6] 杨 雨，陆宇平，吴在桂. 嵌入式大气数据传感系统中的组合滤波技术 [J]. 传感器与微系统，2009，28 (5)：117－120.

[7] Cobleigh B R，Whitmore S A，Haering E A. Flush airdata sensing (FADS) system calibration procedures and results for blunt forebodies [C]. California：Dryden Flight Research Center Edwards，1999.

[8] Whitmore S A，Cobleigh B R，et al. Design and Calibration of the X－33 Flush Airdata Sensing (FADS) System [A]. California：Dryden Flight Research Center Edwards [C]，1998.

[9] Artz E J，Dona N W，Yechout D T R. NASA Orion Flush Air Data Sensing System Feasibility Determination and Development [A]. Maryland：52nd Aerospace Sciences Meeting [C]，2014.