

# 基于 Android 的一种主动监控系统设计与研究

许强键, 杨 飞, 翁玲瑜

(广东工业大学 信息工程学院, 广东 广州 510006)

**摘 要:**针对传统网络视频监控中视频数据体积大和安装复杂的不足,文中提出了一种基于 Android 智能终端的主动监控的方案。介绍了系统整体结构,通过对该系统进行分析和研究,介绍了系统的软件设计思路及运动物体检测方法,并在 Tiny6410 开发板上搭建移植前端交叉编译环境。对异常帧视频图像进行压缩处理后,通过 3G 网络传输到客户端 Android 手机或平板电脑。在无线局域网覆盖区域对该系统进行测试,结果表明,该方案能达到远程主动监控目的,操作友好,易于维护和升级。

**关键词:**主动监控; Android; 帧差法

中图分类号: TP302.1

文献标识码: A

文章编号: 1673-629X(2014)04-0189-04

doi: 10.3969/j.issn.1673-629X.2014.04.048

## Design and Research of an Active Surveillance and Control System Based on Android

XU Qiang-jian, YANG Fei, WENG Ling-yu

(Faculty of Information Engineering, Guangdong University of Technology,  
Guangzhou 510006, China)

**Abstract:** In accordance with the shortcuts in traditional network video surveillance and control, such as the massive video data and complicated installations, propose a new active surveillance and control scheme based on Android intelligent terminal. Introduce the whole system architecture, software design thought and the moving object detection method by analysis and study of the system. Establish and transplant the front-end cross compile environment on Tiny6410. Then after compression of abnormal frame video image, the 3G network will transfer it to the client-end. Test the system in the wireless LAN environment, and the experimental results show that this scheme can realize the purpose of remote active video surveillance and control, is user-friendly, and easier to maintain and upgrade.

**Key words:** active surveillance and control; Android; frame-difference method

## 0 引 言

随着嵌入式技术、计算机视觉技术、网络传输技术的发展,智能视频监控已成为安防领域备受关注的前沿课题<sup>[1]</sup>。传统的利用图像采集卡采集视频图像的数字视频监控系统有两大不足。

第一:不能自动对视频监控区域中感兴趣的运动目标进行判定<sup>[2]</sup>,其“记流水账”的监控方式,不仅使得视频数据体积大,占用大量存储空间,而且需要有人值守,加大了人工成本;

第二:需要铺设专门的视频线和控制线,不利于系统的扩展。

文中设计的监控系统能实时自动检测监控区域内

运动目标,并通过 Android 手机或平板电脑访问出现异常帧的摄像头,真正实现“人防”向“技防”的转变。

## 1 系统总体结构

该系统在整体上由前端网络摄像头、图像处理、无线传输媒介、远程手机终端 4 部分组成,结构如图 1 所示。

(1)网络摄像头。网络摄像头是集视频编码、摄像、Web 服务于一体的高级摄像设备,其内嵌有 TCP/IP 协议栈,每个摄像头都有一个唯一的 IP 地址,将采集到的视频信号经异常帧检测后,通过 3G 无线通信模块,传送至 3G 网络,再由监控终端访问以实现视频

监控。

(2)图像处理。该系统采用帧差法对进入到监控区域的运动目标进行快速检测,并保留异常帧。

(3)无线传输媒介。系统通过无线 3G 网络传输异常帧到远程手机监控终端。

(4)远程手机终端。视频通过解码后,用户通过 Android 手机等掌上终端设备实时显示远程异常帧画面。

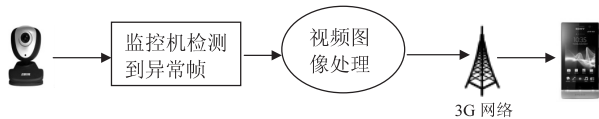


图 1 系统总体结构

## 2 系统的软件设计

### 2.1 系统软件设计流程

主动智能监控区别于传统智能监控的关键在于,前者要能敏锐捕捉到异常帧的出现(如非法人员的闯入),及时向终端设备发送告警信息,同时对视频进行压缩传输;当没有突发事件时,系统不进行视频数据的 3G 传输<sup>[3]</sup>。相对于传统的通过图像采集卡采集视频数据,需要依靠传感器触发非法事件的系统,这种主动的物体监测方法,硬件消耗较低,也更能从全局来判断是否有事件发生。系统软件总体设计流程图如图 2 所示。

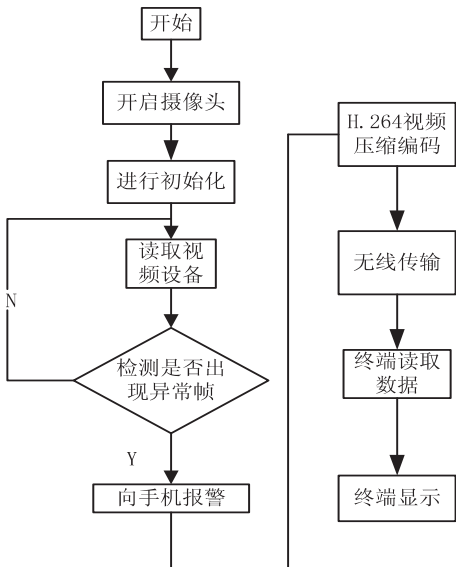


图 2 系统软件总体设计流程图

### 2.2 运动物体检测方法

系统采用帧差法对监控区域中运动目标进行检测。帧差法即是利用图像序列中前后连续两帧或几帧图像的差异来进行运动目标检测的,就是对图像序列中在时间上连续出现的相邻两幅图像求绝对差,然后设定一个阈值判断视频图像的运动特性。其优点是算法简单、无需建模、运算速度快、复杂度低,对缓慢变化

的光照及环境不太敏感,因此具有一定的自适应性。文中采用的编译环境为 VS2008,利用 OpenCV2.1 提供的各类通用算法,在 PC 机上配置好各环境变量,对视频进行仿真测验,如图 3 和图 4 所示。在程序中, cvCreateFileCapture 函数初始化从文件中获取视频, cvNamedWindow 为创建窗口函数,可以创建窗口名字、设置窗口属性标志,函数 cvAbsDiff 为计算两个数组差的绝对值,也即实现前后两帧相减。摄像头初始化完成后,使用函数 cvQueryFrame 从摄像头中得到运动物体视频数据帧<sup>[4]</sup>。程序中设置阈值为 240,表示当前帧灰度图像中大于 240 的像素点数突变时,启动报警机制,向用户告警。

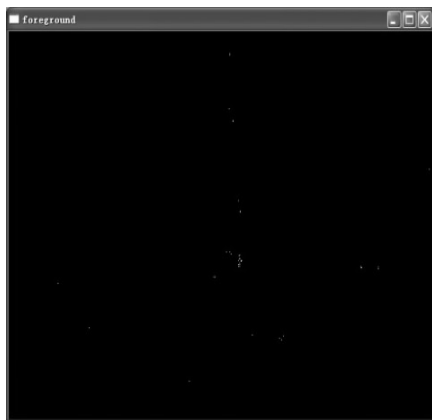


图 3 摄像头检测无运动目标时



图 4 摄像头检测有运动目标时

### 2.3 搭建移植前端编译环境

监控前端的硬件平台是 Tiny6410 嵌入式开发板,软件平台是 Android 操作系统。前端操作系统的搭建和移植可分为以下几步:

①搭建 Linux 开发平台。文中 Linux 操作系统选用虚拟机的安装方式,当安装成功之后即可实现文件共享。

共享文件夹为 D:\VirtualBox VMs\win7share

#mkdir /mnt/linuxshare//设置共享目录

#mount -t vboxsf win7share /mnt/linuxshare//挂载

②建立交叉编译环境。文中选用 arm-linux-gcc-4.3.2 作为交叉编译工具。

```
#tar xvif arm-linux-gcc-4.3.2.tar.bz2 //进行解压命令
```

```
#gedit /root/.bashrc //把交叉编译器路径加入到环境变量中
```

```
#arm-linux-gcc -v //重启系统,使配置生效
```

③解压安装源代码。此步骤为创建一个工作录,将所需要的源代码解压安装到此目录中。

```
#mkdir -p /development/Tiny6410/Android //建立目录
```

```
#tar xvfz /tmp/Android/Android-kernel-2.6.36.tar.gz //解压 Android 内核源代码
```

④配置编译 U-boot。配置和编译 U-boot 需要根据硬件环境进行相应的配置。文中选用的硬件平台的内存容量为 256 M,可支持 NAND Flash 启动的 U-boot 配置编译和支持 SD 卡启动的 U-boot 配置编译。

⑤配置和编译 Linux 内核。在命令行执行以下代码:

```
#cd /development/Tiny6410/Android/linux-2.6.36-Android
```

```
#cp config_Armd_n43.config
```

```
#make
```

代码执行完成后,生成 zImage。

### 3 视频压缩编码

视频监控需要通过网络进行视频数据的传输,这样就对网络的数据传输速率提出了一定的要求<sup>[5-6]</sup>。由于数字化视频数据占用很大带宽,如果不对视频图像进行压缩处理,实现实时传输几乎不可能。视频数据因其有大量的冗余信息,如空间冗余、时间冗余、结构冗余、视觉冗余等,才能进行压缩处理。文中选用 H.264 作为视频压缩编码标准。它有如下优点:编码效率更高、容错能力强、视频画面质量高、网络适应能力强。由 S3C6410 处理器接收摄像头的视频信号<sup>[7]</sup>。采用 S3C6410 作为视频监控服务器的核心芯片,主要是由于很多视频监控系统中视频压缩编码方式采用的是软件编码方式,软件编码方式数据的运算量很大,效率不高,很难实现真正意义上的实时监控。S3C6410 是一个 16/32 位 RISC 处理器,旨在提供一个具有成本效益、低功耗、高性能的应用处理器解决方案,它采用 64/32 位内部总线架构。它具有端口 USB2.0 OTG 高速支持。其自带一个集成的多标准编码器(MFC)提供 MPEG-4/H.263/H.264 编码和解码的速度高达 30 帧/s。视频的 H.264 压缩编码则利用其内部集成的多媒体编解码器(MFC)进行即可。

### 4 网络传输

文中采用实时传输协议(Real Time Protocol,RTP)对压缩了的视频数据进行传输。之所以选择 RTP 协议,是因为 RTP 协议主要是为了实现网络中多媒体数据流的实时传输。该协议可以实现单点或者多点的实时数据传输,提供了相关的时间信息并能够实现数据流的同步。不管是相对于具有延时效果的 TCP 协议还是相对于不保证服务质量的 UDP 协议,RTP 协议都是良好选择。为了将 H.264 的视频数据通过 RTP 协议进行传输,就必须对数据按 RTP 协议进行封装。即将 H.264 的网络抽象层单元(NALU)作为 RTP 数据包的载荷数据组成完整的 RTP 数据帧<sup>[8-9]</sup>。RTP 数据帧的载荷数据定义了三个不同的基本载荷结构,分别是单一 NAL 单元模式、组合封包模式和分片封包模式。H.264 采用的是简单的打包方案,即一个 RTP 分组里面放入一个 NALU,然后设置 RTP 投标值。在接收端,通过 RTP 的序列信息来识别复制包并丢弃,取出有效的 RTP 包里的 NAL 单元。其组合封包模式和分片封包模式可参阅文献[10]。

### 5 Android 客户端解码及显示

Android 操作系统是内核基于 Linux 平台的源码开放的移动操作系统,它自 2007 年发布以来,就以其代码开源性有利于系统的升级换代和市场推广而受到各行各业青睐。因此,从发布至今,Android 一直以顽强的生命力和惊人的速度前进,得到了众多手机制造商的支持<sup>[11]</sup>。其体系结构分为四层,由下到上依次是由 C 语言开发,仅提供基本功能的 Linux 内核、包括虚拟机 Virtual Machine 在内的由 C++语言开发的核心类库、应用程序框架和各公司利用 Java 语言开发的应用程序。

该系统设计的客户端软件是基于 Android 操作系统智能手机开发的,其作用为完成对 H.264 的解码和显示。文中利用 Android 手机模拟器 SDK 开发的登陆界面如图 5 所示。

#### 5.1 客户端解码

视频数据的解码部分,采用 Android NDK+C 的 C 语言实现机制,它是整个解码的核心部分,需要移植 H.264 解码库,通过本地 C 语言的实现和解码库对码流数据进行处理,完成 H.264 解码,实现图像重建。

#### 5.2 客户端显示

文中的视频显示采用的是 Android 提供的 Bitmap 类显示 BMP 图像,H.264 标准支持 4:2:0 的逐行和隔行的视频采样格式。为了能够在应用层完整地显示图像,图像需要在完成色彩空间从 YUV 到 RGB 的转换后,数据传给 Java 层。Android.graphics.Bitmap 类实现

视频显示功能。Android. view. view 类重载 onDraw() 方法,然后调用 canvasdrawBitmap() 方法,将图片连续播放显示在手机或平板屏幕上。当用户首次打开终端登录界面时,需输入远程主机 IP 地址和网络端口号。如果与监控前端摄像头信息一致,则终端与前端的视频连接建立成功,否则表明建立失败,用户需再次进行正确连接。当终端接收到告警信息时,用户即进行登录显示操作,只有在成功登录的情况下,终端才显示非法事件发生界面。



图 5 远程终端登录界面(截图)

## 6 测试结果

当手机接收到报警后,打开登陆界面,在无线覆盖区域,输入监控主机的 IP 地址和网络端口号,实现与监控前端的通信连接。前端为带有 USB 网络摄像头的 PC,客户端为 Android 模拟器。远程主机 IP 地址设为 192.168.1.8,网络端口号设为 8080。终端显示界面如图 6 所示。实现结果表明,该系统运行稳定,达到了预期监控效果。

## 7 结束语

该系统利用 VS+OpenCV 和 Android eclipse 开发环境进行设计及研究。该系统可应用于家居安防系统中,由于手机客户端完全利用开源代码实现<sup>[12]</sup>,故有很强的移植性,便于系统升级。实践表明,在环境变量要求不大的区域,该系统实时反馈效果好,稳定性高。随着智能化、数字化、信息化进程的加快,该视频监控系統极有市场前景。

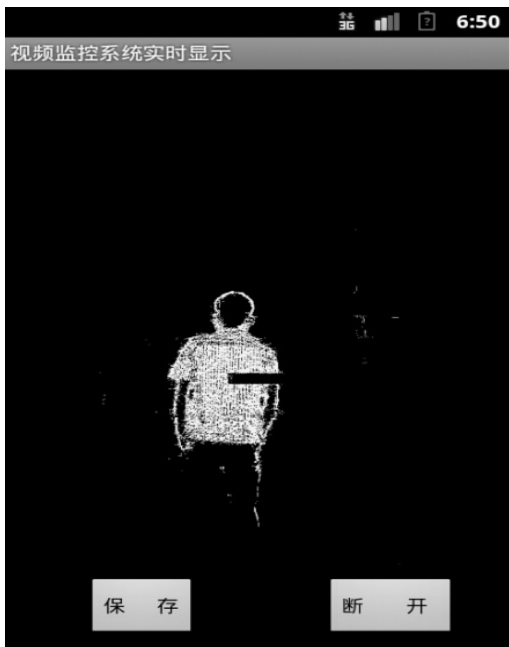


图 6 终端显示界面(截图)

### 参考文献:

- [1] 赵俊. 智能视频监控关键技术研究[D]. 西安:西安电子科技大学,2007.
- [2] 赖明亮. 基于 ARM 的嵌入式监控摄像机设计与实现[D]. 长沙:中南大学,2011.
- [3] 付存宇. 基于 3G 网络的嵌入式无线视频监控系统设计[J]. 计算机与现代化,2013(4):184-186.
- [4] 常丹华,杨冬冬,韩夏. OpenCV 在智能监控方面的应用研究[J]. 电视技术,2009,33(9):101-102.
- [5] 毕厚杰. 新一代视频压缩编码标准-H. 264/AVC[M]. 北京:人民邮电出版社,2005.
- [6] 刘峰. 视频图像编码技术及国际标准[M]. 北京:北京邮电大学出版社,2005.
- [7] 郭巧云,许雪梅,李岸,等. 基于 ARM11 的无线视频监控系统的的设计[J]. 计算机测量与控制,2010,18(8):1786-1788.
- [8] Wenger S, Wang Yekui, Hannuksela M M. RTP payload format for H. 264/SVC scalable video coding[J]. Journal of Zhejiang University Science A, 2006, 7(5):657-667.
- [9] 李校林,刘利权,张杰. 基于 RTP 的 H. 264 视频流实时打包传输的研究[J]. 计算机工程与科学,2012,34(5):168-171.
- [10] Wenger S, Hannuksela M, Stockhammer T. RTP payload format for H. 264 video[S]. RFC3984,2005.
- [11] Gavalas D, Economou D. Development platforms for mobile applications; Status and trends[J]. IEEE software, 2011, 28(1): 77-86.
- [12] 李琴,陈立定,任志刚. 基于 Android 智能手机远程视频监控系统的的设计[J]. 电视技术,2012,36(7):134-136.

基于Android的一种主动监控系统设计与研究

作者：[许强键](#)，[杨飞](#)，[翁玲瑜](#)，[XU Qiang-jian](#)，[YANG Fei](#)，[WENG Ling-yu](#)

作者单位：[广东工业大学 信息工程学院, 广东 广州, 510006](#)

刊名：[计算机技术与发展](#)

ISTIC

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2014(4)

本文链接：[http://d.g.wanfangdata.com.cn/Periodical\\_wjfz201404048.aspx](http://d.g.wanfangdata.com.cn/Periodical_wjfz201404048.aspx)