

# 智能监控中实时预警系统<sup>①</sup>

刘从军, 嵇金荣, 杨 辉

(江苏科技大学 计算机科学与工程学院, 镇江 212003)

**摘 要:** 目前国内的视频监控系统主要用于被动的视频录像, 依靠人工进行查看、分析, 效率非常低. 本文借助智能分析设备设计了一种智能监控中实时预警系统, 该系统采用 H.264 进行编码, 实现了传输与编码的分离, 提高了编码的效率, 利用 RTP, RTSP 和 RTCP 等协议实现了视频流的传输与控制. 该系统真正意义上实现全天 24 小时的实时监控.

**关键词:** 智能监控; 实时预警; 流媒体; H.264; RTP

## Real-time Warning System in Intelligent Monitoring

LIU Cong-Jun, JI Jin-Rong, YANG Hui

(School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhengjiang 212003, China)

**Abstract:** Presently, domestic video monitoring systems are applied for video record. It relies on artificial check and analysis and is of low efficiency. In this paper, with the analysis of intelligent equipment, we design a real-time warning system in the intelligent video surveillance. The system adopts the H.264 coding, realizing the separation of transmission and coding, improving the efficiency of coding. With the RTP, RTSP and RTCP protocol, we achieve the transmission and control of video stream. The system in the true sense brings about real-time monitoring of 24 hours a day.

**Keywords:** intelligent monitoring; real time early warning; streaming media; H.264; RTP

随着移动通信技术和多媒体技术的发展, 视频监控正走向数字化, 智能化. 传统的视频监控已经无法满足人们随时随地获得监控现场情况的需求<sup>[1]</sup>, 为了让用户能随时随地获得监控现场情况以及实现全天 24 小时实时监控, 提出了一种实时预警系统<sup>[2]</sup>. 本系统采用 H.264 标准进行编码, 提高了编码的效率, 采用流媒体技术对视频数据进行传输和控制, 提高了监控视频画面的质量, 并借助智能分析仪, 对摄像头采集的视频图像进行实时分析, 检测到有目标入侵时, 立刻发动警报并发送预警短信等一系列操作, 从而有效进行事前预警, 事中处理, 事后及时取证的全天 24 小时的实时监控<sup>[3]</sup>.

## 1 系统结构

本文中的实时预警系统采用 B/S 和 C/S 相集合的方式, 该系统由以下几个部分组成: 网络摄像头, 智能

分析设备, 流媒体服务器, 短信平台, 电脑, 智能终端等. 该系统不仅可以 PC 访问也可以使用智能终端访问, 系统总体结构如图 1 所示.

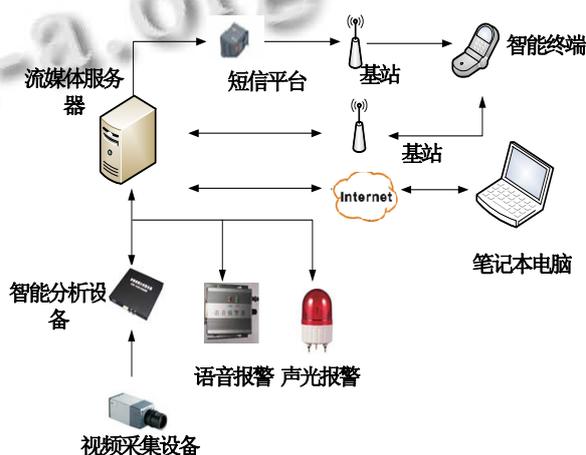


图 1 系统总体框图

① 收稿时间:2013-09-17;收到修改稿时间:2013-10-21

视频数据经过采集模块采集之后,一方面通过智能分析设备对视频数据进行分析,如果检测到入侵目标时,会产生报警信号,服务器检测到报警信号时,会触发现场语音报警和声光报警并将实时预警视频放到指定文件夹下同时通过短信接口给手机终端发送预警短信提示用户;一方面进入编码压缩模块进行 H.264 压缩编码,并将分离出的视频数据添加 RTP 包头和封装 RTP 数据包最后发送 RTP 数据包.在 RTP 会话期间,通过 RTCP 协议统计、管理和控制 RTP 数据包的传输.系统客户端程序的主要工作:①接收 RTP 数据包,从 RTP 数据包中解析出视频数据单元,并送入解码模块进行解码最后通过显示模块进行显示②拦截报警短信查看实时报警视频.该系统使用 RTSP 协议控制服务器和客户端之间的交互.整个系统的工作流程如图 2 所示.

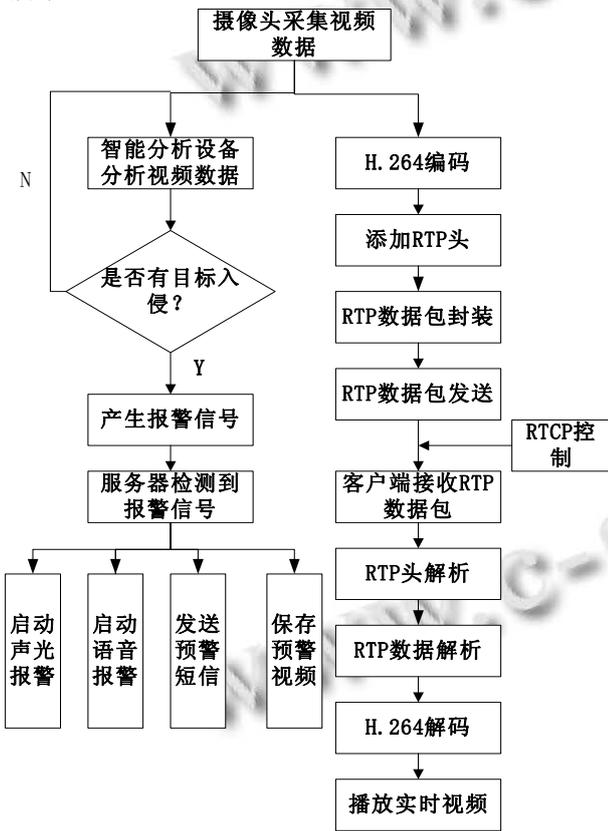


图 2 系统工作流程图

本文采用的智能分析仪是一款网络嵌入式设备,采用了 H.264 视频编码技术,已嵌入区域入侵检测、区域离开检测、绊线周界检测、物品遗留检测、物品丢失检测、逆向运动检测、徘徊检测、速度异常检测、

跌倒检测等算法.该设备能够对获取的监控图像进行自动的分析,根据预设的规则,检测监控场景中的事件,一旦检测出有入侵、物品丢失等可疑事件发生,将能够实时触发报警.该智能分析仪具有两排设备状态指示灯,一个以太网口,一个 RS232 串口,两个报警信息输出口:GPIO 及 RS485 口,两个视频输入口:Video In1、Video In2/Out,两个音频口:音频输入口 Audio In、音频输出口 Audio Out.

## 2 系统客户端设计

### 2.1 实时视频监控模块设计

实时视频监控模块主要实现对监控现场画面进行实时查看,实时视频监控主要由以下几个模块组成:视频通信、解码、显示,通信模块完成服务器数据的接收、分析,将数据载入缓冲区,视频解码模块完成对 H.264 编码的数据进行解码.视频数据的解码与接收过程非常复杂,本文采用多线程同时处理.最后通过显示模块将实时视频进行显示<sup>[4]</sup>.

实时预警系统对现场进行实时监控时,手机客户端接收的数据需进行 H.264 解码,本文中是移植了视频解码库 FFmpeg 进行 H.264 解码<sup>[5]</sup>.

### 2.2 实时预警视频浏览模块设计

实时预警视频浏览模块主要对实时预警视频进行查询,浏览以及删除等操作.查询功能支持模糊查询和按日期进行查询,浏览功能对实时预警视频进行实时浏览,删除功能对历史预警视频进行删除操作.

实时预警视频浏览模块的使用与预警短信拦截程序密不可分,如果没有预警短信拦截程序,用户不知道何时查看实时预警视频,所以预警短信拦截程序对本系统非常重要.预警短信拦截程序实现过程如下:当服务器端检测到入侵目标时,服务器向用户手机发送预警短信,服务器端发送预警短信是调用短信平台接口进行发送,客户端预警短信拦截程序对接收到的短信进行内容匹配,如果短信内容等于事先设置的内容,系统启动指定的预警音乐提示用户有目标入侵,用户登录系统点击实时预警视频浏览,进入实时预警视频列表,列表中最上面一条记录是最新预警视频,点击该记录播放预警视频.如果短信内容不等于事先设置的内容,不做处理.具体流程如图 3.

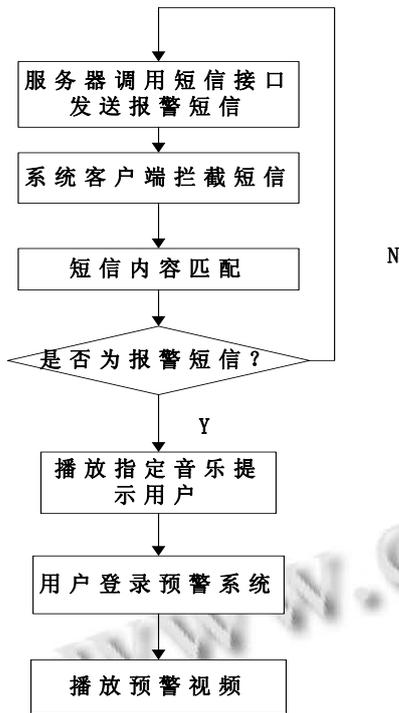


图 3 报警短信拦截流程图

实时预警视频保存的格式是 mp4 格式，程序中采用 MediaPlayer 类对它进行解码。MediaPlayer 播放视频文件时，需要 SurfaceView 来对显示画面<sup>[6]</sup>。由于篇幅限制本文重点介绍系统服务器的设计。

### 3 系统服务器设计

#### 3.1 H.264 视频编码的设计

H.264 标准使用了抗误码技术，可以使用在很复杂的环境下。H.264 标准是目前低码率下压缩率最高的编码标准，在带宽不稳定的无线网络上有着无法比拟的优势，这是本文采用该编码方式的主要原因。

当视频采集模块采集的原始码流进入编码模块时，编码模块需创建编码器以及创建编码通道，创建编码通道时需设置编码属性，编码器和编码通道最多可创建 3 个<sup>[7]</sup>，由于有编码控制接口的控制，所以在编码过程中也可以动态调整编码属性。具体编码流程如图 4 所示。

#### 3.2 预警短信模块设计

预警短信模块是调用短信接口来实现，该模块的代码是使用 C++语言实现的，开发工具使用的是

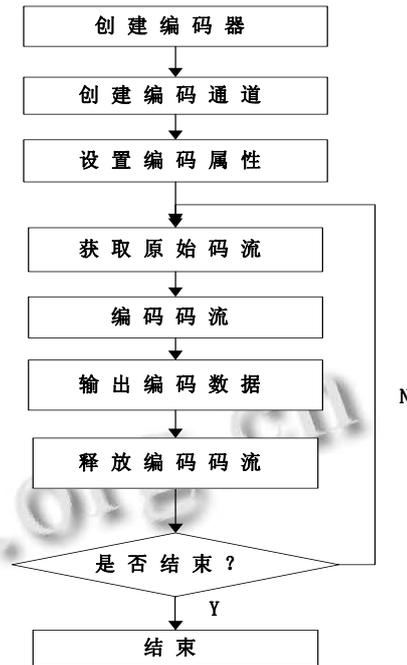


图 4 编码流程图

Visual Studio2008，数据库使用的是 SQL Server2008 数据库，数据库主要保存用户的信息。

当服务器检测到报警信号时，会触发预警短信模块向手机终端发送预警短信，该模块的部分代码如下：

```

int CSendMessage::SendMessage(CString strPhoneNumber, CString strContent)
{
    VARIANT variantSeqID;
    VariantInit(&variantSeqID);
    this->m_descnum = strPhoneNumber;
    this->m_content = strContent;
    if (SUCCEEDED(hr) && spTmp != NULL)
    {
        CComBSTR str;
        char ch[10] = {0};
        sprintf_s(ch, "%d", m_port);
        hr = spTmp->SendBusinessPkg(L"send",
            m_account.GetBuffer(0),
            m_password.GetBuffer(0),
            m_sourcenum.GetBuffer(0),
            m_descnum.GetBuffer(0),
            m_ipaddr.GetBuffer(0),
            ch,
            m_content.GetBuffer(0),
    
```

```

        m_sendtype.GetBuffer(0),
        m_comid.GetBuffer(0),
        m_needreceipt.GetBuffer(0),
        &variantSeqID);
    }
    return hr;
}

```

### 3.3 质量保证 QoS 的设计

在监控系统中影响 QoS 质量的方面有: 丢包、拥塞、抖动等。本文设计了一种根据网络状况自动调整传输码率的方案。该方案使用 RTP 和 RTCP 相结合的策略[7], 通过反馈机制, 计算丢包率, 生成 RTCP 数据包定期向服务器端反馈。服务器端根据反馈信息来控制传输码率。

### 3.4 会话控制设计

会话控制模块主要使用 RTSP 协议对服务器和客户端的会话进行建立和控制。具体设计方法如下[7-8]:

① OPTIONS: 获得服务器提供的可用方法的集合。

② DESCRIBE: 得到会话描述信息。

③ SETUP: 客户端提醒服务器建立会话, 并确定传输模式。

④ PLAY: 客户端发送播放请求。

⑤ PAUSE: 客户端发送临时中断请求。

⑥ RECORD: 客户端发送录制请求, 如果本地没有存储设备, 则返回一个错误给客户端。

⑦ TEARDOWN: 客户端请求停止会话, 服务器断开会话并释放所分配的资源。

⑧ KEEPALIVE: 客户端定期向服务器报告在线信息(扩展方法)。

⑨ GETLOG: 客户端发送获取日志信息请求(扩展方法)。

客户端向服务器请求视频时, 发送 4 个交互过程: Option、Describe、Setup、Play。流媒体服务器收到请求后, 响应请求, 发送视频数据包<sup>[9]</sup>。当客户端停止视频播放时, 发送停止请求 TEARDOWN 给流媒体服务器, 流媒体服务器收到停止请求后, 停止发送视频数据包。

### 3.5 系统测试结果

在 3G 网络和无线网络覆盖范围内, 对系统进行了整体测试, 系统非常稳定, 达到预期的目标, 实时预警视频浏览界面如图 5 所示。



图 5 实时预警视频浏览界面

## 4 结 语

本文对智能监控中实时预警系统进行了详细设计, 采用 H.264 对视频数据进行编码, 使用 RTP 和 RTCP 相结合的方式对数据进行传输和控制, 并使用 RTSP 进行会话建立和控制, 该系统不仅提高了监控画面的质量, 而且真正意义上实现了全天候 24 小时实时监控。

### 参考文献

- 1 常志沛. 基于 Android 的智能手机视频监控系统的设计与实现[学位论文]. 大连: 大连海事大学, 2011.
- 2 Shiddiqi AM, Pratama H, Ciptaningtyas HT. A video streaming application using mobile media application programming interface. *Telkonnika*, 2010, 8(3): 293-300.
- 3 杨欧. 基于 Android 平台的视频智能监控报警系统. *中国高新技术企业*, 2012, (7): 28-29.
- 4 曹晓芳, 王超, 李杰. 一种基于 Android 智能手机的远程视频监控的设计. *电子器件*, 2011, 34(6): 709-712.
- 5 鲍轩, 章坚武. 基于 Android 的音视频监控软件的设计. *杭州电子科技大学学报*, 2012, 32(4): 61-64.
- 6 陈泽恩. 基于 Android 平台的移动监控系统设计与应用分析. *软件*, 2012, 33(10): 52-54.
- 7 朱龙成. 多路视频监控中流媒体服务器的设计与实现[学位论文]. 武汉: 华中科技大学, 2009.
- 8 章民融, 徐亚锋. 基于 RTSP 的流媒体视频服务器的设计与实现. *计算机应用与软件*, 2006, (7): 93-95.
- 9 曾金, 毛燕琴, 沈苏彬. 嵌入式流媒体服务器的设计和实现. *计算机技术与发展*, 2011, (7): 81-84.