单位代码	10602
学　　号	2014011688
分 类 号	TN948.6
密　　级	公开

广西师范大学
GUANGXI NORMAL UNIVERSITY

硕士学位论文

基于 H.265 的无线视频监控系统设计与实现

Design And Implementation Of Wireless Video Monitoring System Based On H.265

学　　院：电子工程学院

专　　业：电子与通信工程

研究方向：嵌入式系统开发

年　　级：2014 级

研 究 生：钟俊文

指导教师：宋树祥　教授

完成日期：2017 年 4 月

基于 H.265 的无线视频监控系统设计与实现

专业名称：电子与通信工程

申 请 人：钟俊文

指导教师：宋树祥 教授

论文答辩委员会

主席：

委员：

基于 H.265 的无线视频监控系统设计与实现

研究生：钟俊文　　导师：宋树祥 教授

专业：电子与通信工程 研究方向：嵌入式系统与应用　年级：2014 级

摘要

近年来，随着中国社会与经济的飞速发展，安防系统在中国乃至世界显的尤为重要。回顾近十几年来，在视频编码的领域，市场上绝大多数所利用的视频编码标准基本都是 H.264，但是随着目前高清化、智能化以及网络化的快速发展，H.264 不能完全满足应有的需求。所以本文通过研究现有的视频监控系统，设计了一种基于 H.265 的无线视频监控系统。该系统主要由两部分组成，分别是监控服务器和监控客户端。该系统具有所呈现画面更加清晰、编码效率更高、性能更加稳定等优点。并且克服了有线监控系统的大面积布线的问题。本设计主要完成了以下几个方面的研究工作：

（1）本文对系统开发过程中用到的视频压缩编码算法 H.265、无线传输介质以及流媒体传输协议 RTP/RTCP 和会话协议 RTSP 进行了详细的分析。

（2）研究和学习了 Hi3516A 的架构、特性、基本组件以及常使用的外围模块，针对本系统设计了系统的整体方案和主要硬件的电路设计。

（3）监控摄像端的嵌入式摄像服务器平台搭建，包括 Bootloader 的移植，Linux 内核的移植以及文件系统 YAFFS2 的制作等。

（4）在监控服务器端对媒体处理软件进行开发，包括视频的采集、视频的处理、视频的编码以及多媒体处理平台 MPP 的实现。

（5）视频传输的开发，包括对应用在 UDP 之上的 RTP 传输协议进行开发以及会话协议 RTSP 进行开发。通过对 4G 模块驱动的移植来实现监控服务器与客户端进行 4G 信号无线传输的方式。

（6）在监控客户端实现对视频信号的接收以及对 FFmpeg 进行开发实现 H.265 压缩视频信号的解码；通过对 Directdraw 开发实现视频信号的显示；通过对 OpenCV 的开发实现对监控视频的目标检测功能。

（7）将各个模块测试后对整个系统进行了测试，并对测试结果进行了分析。结果表明，将新一代视频编码 H.265 到 4G 无线传输开发融合在一起，无线视频监控系统能够正常运行，并具有高压缩性和实时性。

关键词：视频监控；Hi3516A；H.265；4G；FFMPEG

Design And Implementation Of Wireless Video Monitoring System Based On H.265

Graduate Student: Junwen Zhong Supervisor: Prof. Shuxiang Song
Major: Electronics and Communications Engineering Grade: 2014
Research Direction: Embedded system development

Abstract

In recent years, with the rapid development of China's society and economy, security systems is particularly important in China and even the world. Review the past decade, in the field of video coding, most of the video coding standards used on the market are mainly H.264. But with the rapid development of current high-definition, intelligent and networked, H.264 can't fully meet their needs. By studying the existing video surveillance system, this paper designs a kind of wireless video monitoring system based on H.265. The system mainly consists of two parts which are respectively the monitoring server and monitoring client. The system has the advantages of clearer rendered images , higher coding efficiency and more stable performance, etc. It also overcomes the problem of cable monitoring system of large area of wiring. This design mainly finished the following several aspects of research work:

1. This paper makes a detailed analysis of the video compression coding algorithm for H.265, wireless transmission medium, and streaming media transmission protocol RTP/RTCP protocol RTSP session that used in the system development.

2. Research and study the Hi3516A's architecture, features, basic components and commonly used peripheral modules. This paper designs the overall scheme of the system and the main hardware's circuit design.

The embedded video server platform construction of monitoring camera, involving the transplanting of Bootloader, the transplantation of the Linux kernel and file system YAFFS2 production.

4. Develop the media processing software in monitoring the server-side, including video capture, video processing, video coding and multimedia processing platform for the realization of the MPP.

5. The development of video transmission, including the development of the application on UDP transmission protocol RTP and RTSP session agreement. The 4G module driver transplant and implementation to realize the monitoring server and the client for 4G wireless signal transmission way.

6. To realize the H.265 compressed video signal decodino through receiving the video signal inn monitoring client implementation and the FFmpeg development ; to realize the display of the video signal through the Directdraw development; to realize the monitor video target detection through the OpenCV development.

7. Test the whole system through connecting each module and analyze the test results.The results show that after combining the new generation of video coding H.265 and 4G wireless transmission fuses in together, the wireless video monitoring system can run normally, and has the real time and high compressibility.

Key words: video monitoring; Hi3516A; H.265; 4G; FFMPEG

目录

第 1 章 绪论

1.1 研究背景及意义

近年来，随着中国社会发展迅速，经济实力逐渐增强。网络视频监控在安防领域也取得了巨大成就。回望过去几年，从 2004 年的科技强警示范城市建设与全国平安城市建设到 2008 年北京奥运会的完美安全结束，从“十二五”计划的完满完成到《“十三五”平安中国建设规划》起草实施。无一不在刺激中国安防的发展与拉动。就现在而言，中国安防各专业领域全面发展，形成了相对完整的产业链体系。所以 2016 年 9 月 4 日在中国杭州首次举办的二十国集团领导人第十一次峰会(G20 峰会)非常成功。在 G20 峰会举行中负责是安保任务承担者也可以说是安防设备供应商就是中国安防的龙头企业：海康、大华、宇视等等。在 G20 峰会时期这些安防的较强企业靠着良好的服务能力和产品质量，将峰会期间的安防任务很圆满地完成[1]。对于来自二十个国家领导人的考验算基本完成。经过这次峰会使国际市场看到了我们中国在有很好的安防企业，并且安防的产品具有较好的表现力。这必定有利于中国安防以及安防的产品走出国门迈向世界。

虽然中国乃至全球的安防日益加强，但是恐怖活动依旧存在，在近两年的时间，公共场所依旧发生了一些让人意想不到的事情，比如上海滩发生的多人踩踏事件、法国巴黎出现的恶意袭击事件以及含有三位死亡中国人的马里人质劫持事件。这成为国内外关注的重点。在中国通过改革开放使中国人民的物质文化水平不断提高，与之同时，我们对人身安全以及财产安全的要求也不断提高。这就要求我们借助高效先进的安防工具来保障我们公共场所的安全和我们需要保护的人身安全以及财产安全。所以提高国内乃至世界的各方面安防的有效措施之一就是在可允许的范围安装视频监控设备，这不仅可以预防危害人类的事件发生，对已经发生的危害事件进行罪犯的抓获也有良好的帮助。

视频监控的应用范围从银行、安防部门等行业延伸到家庭、交通安全、通信安全以及金融保障等。视频技术的飞速发展，尤其在在视频的高清(HD)甚至超高清(UHD)以及 3D 视频和多视点(multi-view)视频技术发展的带动下，各种各样的视频信息已经无所不在的出现在我们的生活中。据估计，视频流数据将在 2018 年占到整个互联网流量的 90%以上，说夸张一点，整个互联网络差不多就是视频网了[2]。但是视频监控依然存在两方面的问题，第一个问题是尽管近年来网络带宽和存储能力增加迅速，但是也远不能满足以海量信息为特征的视频数据的传输和存储的要求。其次传统的有线视频监控对于公园、校园、社区等地依旧需要大量的布线，并且对于死角不能完全覆盖[3]。监控室不能随便的变动，否则需要再次重新布线并且耗费人力物资，大大的增加了使用成本。

针对传统视频监控系统的这两个问题，首先视频数据量的高效压缩无论在过去还是现在以及在未来一直是加快信息的传输速度以及减少信息存储内存的重要技术措施之一。在 2013 年发布的新一代高效视频编码标准 H.265/HEVC 具有明确的目标就是提高视

频的编码效率，保证一样的图像质量下，压缩率相对 H.264/AVC 的高档次(high profile)翻一番；支持的视频格式更广泛，支持的分辨率不仅是从 QVGA(320×240)到 1080p(1920×1080)，还支持超高清视频 4320p(7980×4320);在计算复杂度、压缩率、鲁棒性和处理延时之间妥善折中处理[4]。对于有线视频监控存在的问题，我们的解决办法为通过无线进行视频传输的方式。随着 4G 信号以及 WIFI 信号的普及，对无线视频监控的发展做了良好的铺垫。而中国的 4G 信号已经逐渐成熟，具有覆盖面积广、传输速度快、兼容性强、所需成本低的优点。所以在视频监控系统中采用 4G 网络传输可以实现传输速度快、稳定性强、所需成本低等优点。再有新一代的高效视频编码标准 H.265 的加入，传输高分辨率和高质量的视频图像就不成问题。对于将来的超高清 4k 视频实现无线传输应用在现在所流行的直播、视频聊天以及高清无线监控做了良好的铺垫。所以开发基于 H.265 的无线视频监控系统具有很大的潜在价值和实用价值。

1.2 视频监控技术的发展历史和对未来的展望

最早的视频监控系统作为一种报警复核手段出现在安全防范系统。由于它可以通观全局和一目了然的对事件进行判断，与此同时还拥有很高的准确性和实时性,现在已经成为安全防范系统技术集成的核心[5]。并且在不久的将来，成为安防主导技术的必定是由机器解释（通过机器识读，用存储的图像信息进行对比）替代目前的目视解释（当前安全防范系统主要方式）。视频监控系统按照其技术的发展经历了四个阶段：

（1）模拟视频监控系统

在 20 世纪 70 年代到 90 年代之间,我们所用的监控系统被称为全模拟的视频监控系统，也就是我们所说的第一代监控系统。

我们也将第一代视频监控系统也称为闭合电视监控系统。其主要以模拟视频矩阵和模拟磁带录像机为核心。图像信息是通过采取视频电缆模拟的方式进行传输，所以传输的距离比较短，这种模拟视频大多应用于小范围的集总式的网络构架，其监控的图像基本只能在监控中心查看[6]。第一代视频监控系统所具有的优点是视频、音频信号的采集、传输、存储都是模拟的形式。经过这些年的发展，具有成熟的技术和稳定的系统[7]。但是也存在传输距离短、易受干扰、与信息系统不易交换数据以及灵活性差的缺点。

（2）数字视频监控系统

到了 20 世纪 90 年代，我们所用的视频监控系统已经发展为数字视频监控系统。在第二代的视频监控系统中，主要是以 DVR（数字硬盘录像机）为核心，在 DVR 中进行视频图像的长时间的录像、录音、远程监视和控制的功能。

在第二代视频监控系统中，视频从摄像机采集视频到数字硬盘录像机还是通过同轴电缆进行模拟传输的，但是在数字硬盘刻录机上则进行数字化并且在本地压缩存储。第二代视频监控系统的构架首先是采用 PC 和 windows OS 平台的。后期随着信息处理技术和数字技术的发展出现了嵌入式数字硬盘刻录机，而且是建立在一体化的硬件结构上，

整个音视频的压缩、显示和网络等功能都通过一块单板来实现，很大的提高的整个系统硬件的可靠性和稳定性。

（3）网络视频监控系统

当监控视频系统发展到20世纪90年代末期时，由于网络水平不断的发展，所以第三代视频监控系统浮出水面，第三代视频监控系统又叫做全数字视频监控系统。全数字视频监控系统在网络视频服务器和网络摄像机的共同使用下，把图像的采集、压缩、协议转换和传输设置在监控点[8]。由于互联网和局域网的发展可以实现分布式网络的即插即用，这就将视频监控这一流程完全实现数字化。

（4）智能视频监控系统

在视频监控系统发展到21世纪时，人们对视频监控系统更加注重的是其网络化、高清化、智能化。而恰好随着嵌入式以及人工智能的高速发展将大数据、高效压缩芯片以及物联网和云计算等高科技技术融入到新一代的视频监控系统中。

虽然我们一直在向网络高清智能的方向迅速发展，但是在视频监控系统这些年的发展上也遇到了不少问题。并且我们无论在以前、现在还是不久的将来一直面临的解决并且存在的问题。问题所在就是视频所占存储空间和网络传输带宽相互矛盾。解决这个问题必将对视频监控以后的发展有着巨大的影响和意义。

1.3 视频编码国内外的发展

视觉是我们感知世界的重要方式之一，因此我们一直努力的让人们看到更清晰更好的图像信息。特别是现在我们进入数字时代以后，数字视频更是紧随IT技术的浪潮飞速的发展。但是视频作为一种数据量非常大的信息载体想获得实际应用，必须采取高效的数据编码。自20世纪80年代以来，国际标准化组织一直在持续研究视频编码方法，每一次视频编码国际标准的颁布，都会很大程度上促进视频技术的发展，催生更多的视频应用。相应的，视频应用的不断涌进也为视频编码提出了更高的要求，进而推动着视频编码标准向更高的压缩效率不断挺近。

国际广播电视视频音频编解码四大标准：VC-1、国际化标准化组织运动图像专家组的MPEG系列、国际电信联盟（ITU）的H.26x系列和我国具备自主知识产权的第二代信源编码标准AVS系列[9]。目前视频编码研究的热点是MPEG系列和H.26x系列。编码标准国内外主要历程如图1.1所示。

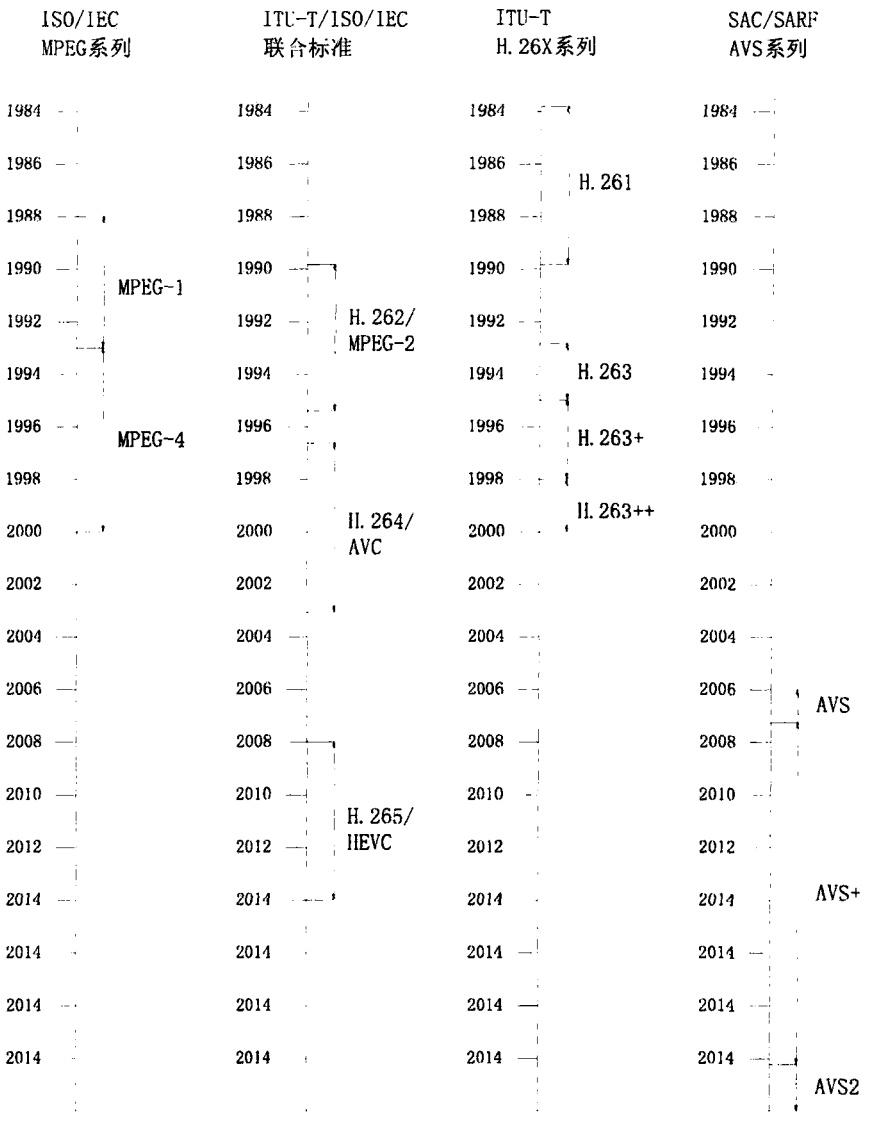

图 1.1 编码标准历程

对于 H.26x 系列，首先是 1993 年 3 月制定的针对在带宽为 64kbit/s 的倍数的综合业务数字网（ISDN）上实现电信会议应用，特别针对于面对面的可视电话和视频会议而设计的 H.261。由于世界各国所采用的电视制式不同，主要有 PAL 和 NTSC 两大类，如果这些国家想建立可视电话和视频会议的业务，是不能直接使用电视信号进行传输的，因此 H.261 提出一种中间格式的视频（CIF）来解决这个问题。在 1998 年 2 月，国际电信联盟为了解决低码流通信而设计了 H.263 标准。H.263 标准在原理上基本继承了 H.261 标准，但也做了一些改善，提高了性能，在 H.263 标准与 H.261 标准的对比下，对于压缩低码率的图像具有更好的图像效果。在 2003 年 ITU-T 的 VCEG 和 ISO/IEC 的 MPEG 首次合作成为联合视频组并共同开发了数字视频编码标准 H.264。H.264 标准可分为 3 类：基本类属于简单的版本，其应用面广。主类采用了多项提高图像质量和增加压缩比的技术措施用于 SDTV、HDTV 和 DVD，扩展类用于各种网络的视频流传输[10]。同时

H.264 标准仍然沿用了混合编码的方法，在视频编码层(VCL)进行具有方向性的帧内预测、运动补偿、数据分割和自适应的熵编码等技术[11]。在定义的网络抽象层（NAL）中将 H.264 视频编码层的网络数据与下层的传输协议有机连接。通过这些技术的应用，H.264 标准所编码的视频数据比之前的 H.263 标准编码的数据减少大约 50%的码流，所以 H.264 标准的出现，大多替代了之前的 H.263 标准以及 MPEG 标准[12]。成为主流的视频编码标准。在国际市场上，以 H.264 标准为编码的芯片发展迅速，并且在市场上发挥着相当重要的作用，主要以 Sigma Design、Broadcom 和 Conexant 作为主要的芯片生产公司代表。并且 H.264 标准的应用面是相当广泛的，不仅在 IPTV 中作为视频编码的标准编码算法，在一半以上的生产机顶盒的企业也应用的 H.264 标准。在编码网站 Encoding.com 中可以看到在 2009 年时对五百万段视频数据的编码标准进行统计显示：H.264 标准在全球市场的应用率极速上升，从之前的所占市场的三分之一上升到占有市场应用的三分之二。但是在国内市场上，我们虽然有像清华大学和中国科学院等知名学校对 H.264 标准进行了深入的研究，也有像深圳市海思半导体公司和大华等知名企业对其进行生产编码芯片，但是由于国外对其编码芯片的操纵，我们基于 H.264 标准的产品没有得到很好的扩展，大多还是用的 MPEG-2 标准。

但是随着最近几年高清、超高清视频在我们生活中出现的越来越频繁，H.264 标准在高清视频使用时显得有些吃力，不能在有限带宽下满足我们对视野中超高清的要求。如今，数字视频广播、移动无线视频、远程监控以及医学成像等，都是和我们的生活息息相关的，而这些都需要我们实现视频的高青化和智能化。所以在 2010 年 4 月 ITU-T 的 VCEG 和 ISO/IEC 的 MPEG 第二次组建了视频编码联合组（JCT-VC）,为新一代的视频编码标准 H.265/HEVC 进行了合力制定(这里的 H.265 为 H.265 的视频编码协议标准)。并且 ISO/IEC 在 2013 年 11 月正式发布了 H.265/HEVC 标准。虽然新一代视频编码标准 H.265 在算法计算的复杂程度上是过去十年间占据大部分市场应用标准 H.264 标准的两倍以上，但 H.265 在并行处理能力和网络适应能力上则大大加强，而且在对相同质量的图像数据进行压缩的情况下，H.265 标准可以比 H.264 标准节约大约一半以上的码流。更重要的一点，也是其他编码标准不能实现的一点就是 H.265 标准还支持 4K(分辨率为 4096x2160)和 8K(分辨率为 8192x4320)的超高清视频[4]。但是由于芯片制作周期长、持有主要专利权的三星、联发科等知名企业并没有同意将 H.265 放进 MPEGLA 的专利池内、芯片的制作成本以及对其稳定性的考察等因素一直制约着 H.265 在安防行业大面积使用。不过通过技术的成熟以及 H.265 在小面积安防成功的影响下，再加上新一代标准的种种优点。H.265 无论是在现在流行的直播上面实现超高清直播，还是在实时会话领域替代之前的 H.264 编码标准都将是一个发展趋势。特别是在安防系统中，在减少大量数据存储成本的同时，还可以增加清晰度，所以 H.265 编码标准给安防系统开了一扇新的大门。

1.4 论文的主要工作

在对视频监控系统的背景和意义进行了全面了解和对视频压缩算法的历程进一步学习后。本文设计了一种基于 H.265 的无线视频监控系统。该系统主要由两部分组成：监控服务器和监控客户端。监控服务器是以 Hi3516A 作为主要控制器进行开发，实现对视频信号的采集编码以及发送。而客户端是在 PC 机上实现的，将接收到的视频信号通过开发 FFmpeg 对视频进行 H.265 解码。并通过对 DirectDraw 开发实现视频的显示功能等以及进一步实现了对视频信号的目标检测。主要工作如下所述：

（1)对现阶段视频监控系统技术做了大量研究，对视频监控系统中压缩算法的研究、无线传输介质的研究、处理器方案的研究等。根据研究对比设计了一套视频监控系统。

（2)）根据本论文对监控系统的要求，对系统进行整体设计。对 Hi3516A 芯片进行学习开发，以 Hi3516A 为中心服务器设计外围电路，实现系统的硬件电路。

（3）对 Hi3516A 进行嵌入式开发环境的搭建，包括交叉环境的搭建、内核的移植以及根文件系统的制作等。

（4）对 Hi3516A 的媒体处理软件的开发，其中主要包括对视频图像的采集、处理、编码以及对 MPP 进行移植实现。视频编码对 H.265 进行开发使采集到的视频通过 H.265 编码。

（5）对本系统中传输模块进行开发，主要包括 RTSP 的开发、对 4G 模块进行开发使其能够进行在客户端与服务器间进行 4G 无线通信。

（6）对 FFmpeg 进行开发实现对 H.265 编码视频的解码；通过对 DirectDraw 进行开发显示出来并实现视频信号的录像、录像回放等功能；对 OpenCV 进行开发实现对监控视频的目标检测。

（7）对整个系统进行了测试，并对测试结果进行了分析。

第2章 视频监控主要技术研究对比与分析

2.1 视频压缩与编码技术对比与选择

2.1.1 现代视频编码技术分析

在多媒体普遍应用的今天，高清化和智能化围绕着我们生活的点点滴滴。我们生活的方方面面都在提高的同时，我们使用的多媒体的数据量也在相应增加，虽然存储性能也在不断增强，但是远远不及我们用多媒体产生数据量的增加速度。这时我们就要着手于对视频的编码。尽最大努力去减少多媒体所占用的数据量来减少存储成本和时间，但又不能影响人们的视觉感受。

一直以来，视频监控系统所面对的最重要的挑战无非于庞大的原始视频数据量的存储和传输。所以视频编码技术对于多媒体的应用特别是视频监控系统中占据着重要的位置。通过使用现有的视频编码标准来对视频原始图像进行编码，使其在视频的存储时占有较少内存和在视频的传输时占有较少的带宽。我们所看到的视频信息其实是由一帧帧图像组成的，加上人体视觉本身的暂留效应。所以当连续帧不间断出现时就组成了视频。对视频进行编码其实就是对视频中的图像信息进行编码。也就是对视频中帧信息以及相邻帧进行编码。但是不同的视频信息具有不同的特性，我们还应该将特性考虑在视频编码中，这样可以更好的实现编码效率。视频编码技术的核心是利用人的视觉心理特性（如视觉暂留、大块着色原理等）以及信息论的理论，去除视频图像中的信息冗余部分[13]。我们在视频编码中需要解决的基本问题同时也是最重要的问题就是怎样去除图像信息中的时间冗余、空间冗余以及统计冗余。在市场上所用产品的视频编码标准中主要应用了以下三种技术：用于去除时间冗余的基于运动补偿的时域预测、用于去除空间冗余的基于块的变换编码、用于去除前两个技术所生成数据的统计冗余的基于熵编码[14]。以上所述的三种技术相组合后形成基于块的预测变换混合编码框架如图 2.1 所示：

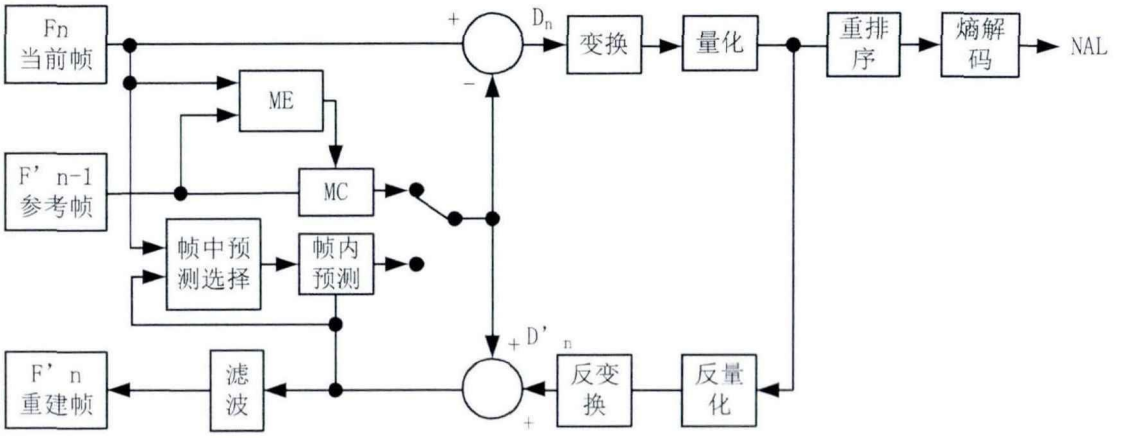

图 2.1 混合编码框架

① 图像进入编码器后，被分割成由一个 16*16 像素的亮度分量和两个 8*8 像素的色度分量所组成的宏块。并且通过光栅扫描来进行依次处理后进行编码。

② 通过码流控制模块来决定是采用帧内编码还是帧间编码，主要的依据是根据宏块特性以及当前的信号带宽速率。

③ 若进行帧内编码，则根据同一帧内进行频域或者空域预测，预测残差进行变换和量化。

④ 若进行帧间编码，在解码缓存中找到已编码相邻帧间进行预测，进行运动估计然后重构图像获得运动信息以后获得运动矢量信息，并且将预测残差进行变化和量化。

⑤ 无论进行的哪种方式编码，将其编码后的数据放进熵编码器后生成最终码流。

⑥ 为了减少误差，我们在编码器末端接入一个解码器来进行反量化和反变换来重构图像，并且将重构图像保存在解码缓存中，给下一帧或者需要帧来使用预测参考。

⑦ 对于解码端的重构图像，一定会存在我们留下的编码痕迹，我们可以采取环内区块滤波来有效减少。或者可以在显示前进行后处理工作，能在不影响编码预测的同时进行生成图像的质量改进。

2.1.2 主流视频编码技术对比与选择

国际广播电视视音频编解码四大标准：VC-1、国际标准化组织运动图像专家组的 MPEG 系列、国际电信联盟（ITU）的 H.26x 系列和我国具备自主知识产权的第二代信源编码标准 AVS 标准[9]。现在市场上利用率最高的也是将来的发展趋势的压缩标准就是 MPEG 系列和 H.26x 系列，下面主要分析这两种当下最热的压缩标准。

MPEG(Moving Picture Exports Group)根据英文全写翻译为活动图像专家组。MPEG 是第一个得到授权允许来制定编码标准的组织。目前为止 MPEG 已经制定了 MPEG-1、MPEG-2、MPEG-4、MPEG-7 以及 MPEG-21 标准等，下面主要介绍这几种常用标准的特性。

MPEG-1 是该系列的第一个标准，于 1993 年 8 月发布。主要用于传输 1.15Mbit/s 数据传输速率的数字存储媒体运动图像以及伴随的音频编码。主要应用于数字媒体上动态图像与音频的存储和检索，例如 VCD 和 MP3 等。虽然当时应用较多，但是其只支持逐行视频而不支持隔行视频。

MPEG-2 编码标准不仅支持逐行视频也支持隔行视频，并且在传输速度上可以达到 3Mbit/s 以上。比较特别的是在这个标准中开始引入了档次和等级，可以针对不同的应用要求来进行不同编码模式的选择。按图像的分辨率分成四个“等级”，而按不同的编码程度分为五个“档次”。对于某一输入格式的图像，采用特定集合的压缩编码工具。产生特定速率范围内的编码码流。MPEG-2 的应用范围包括卫星电视、有线电视以及 DVD 产品的核心技术上。

MPEG-4 相比之前介绍的编码标准涵盖的内容更加广阔，它包含多达 31 个部分，分

别定义了系统、音视频编码、多美图传输集成框架、知识产权管理、动画框架扩展和3D图形压缩等内容[16]。其中第10部分就是大名鼎鼎的H.264。主要有三大特性，分别为基于内容的交互性、高效的压缩性、通用的访问性。其编码效率可达上一版本的1.4倍之高。其广泛应用于互联网音视频广播、无线通信、计算机图形仿真、电视电话等等。

MPEG-7主要应用于多媒体管理，描述多媒体的特征并提供可用的最全面视听描述工具[15]。而MPEG-21为多媒体框架，主要为多媒体信息的用户提供透明且有效的电子交易和使用环境，这里不做仔细介绍。

（2）H. 26x系列标准

H.261标准是1993年3月制定的针对在带宽为64kbit/s的倍数的综合业务数字网（ISDN）上实现电信会议应用特别是面对面的可视电话和视频会议而设计的。由于世界各国所采用的电视制式不同，主要有PAL和NTSC两大类，如果这些国家想建立可视电话和视频会议的业务，是不能直接使用电视信号进行传输的，因此H.261提出一种中间格式的视频（CIF）来解决这个问题。H.261标准开拓新领域应用在数字视频编码标准中，其主要编码方法包括帧间预测（以运动补偿为基础）、空域变换编码（以离散余弦变换（DCT）为基础）、熵编码、量化、zig-zag扫描的混合编码框架等，这些编码技术组合在一起就形成了沿用至今的混合编码（Hybird）框架。

H.263标准是国际电信联盟在1998年2月为了解决低码流通信而设计的。H.263标准基本原理继承了H.261标准，但是做了一些改善，增加了更高级的预测模式并且在算术编码的语法上都有很大的改进，提高了性能，使H.263标准相比H.261标准之下对于压缩低码率的图像具有更好的图像效果。并且在其发展的过程中，出现了两个增强功能的改进版本，分别为H.263+和H.263++,相比H.263标准可以允许更多的图像输入格式，扩展了视频编码的范围，为视频编码的发展奠定了基础。

H.264标准是ITU-T的VCEG和ISO/IEC的MPEG在2003年首次合作成为联合视频组并且共同开发了数字视频编码标准。H.264标准可分为3类：基本类属于简单的版本，其应用面广。主类采用了多项提高图像质量和增加压缩比的技术措施可用于SDTV、HDTV和DVD，扩展类用于各种网络的视频流传输[10]。同时H.264标准仍然沿用了混合编码的理念，在视频编码层(VCL)进行具有方向性的帧内预测、运动补偿、数据分割和自适应的熵编码等技术。在定义的网络抽象层（NAL）中将H.264视频编码层的网络数据与下层的传输协议有机连接。通过这些技术的应用，H.264标准所编码的视频数据比之前的H.263标准编码的数据减少大约50%的码流，所以经过H.264标准的出现，大多替代了之前的H.263标准以及MPEG标准。成为主流的视频编码标准。

2010年4月ITU-T的VCEG和ISO/IEC的MPEG第二次组建了视频编码联合组（JCT-VC）,为新一代的视频编码标准H.265/HEVC进行了合力制定。并且ISO/IEC在2013年11月正式发布了H.265/HEVC标准。在新标准中，虽然编码框架没有革命性的改进，但是在每个模块下都引入了新的编码技术。不仅如此，H.265具有自己的编码特

性。例如将块灵活的进行四叉树方式分割、帧内预测模式的角度增加、运动矢量预测具有自适应功能、离散余弦变换可以变化尺寸和最新的样点自适应补偿滤波器等[17]。因为具有这些优良的特性，是新的编码标准在视频编码领域具有新的天地。不仅支持 1080p 格式视频编码，更是延伸到超高清的 4k 视频以及 8k 视频编码。测试表明，H.265 标准可以在 1.5Mbit/s 以内的带宽下无卡顿的传输 1080p 的高清视频[17]。这为我们高清甚至超高清视频实现在网络中传输成为可能。更为我们在现有的无线传输的条件下实现相同带宽下传输更清晰的视频。

对比上述所述的两大系列视频编码标准的性能及应用，我们选用 H. 26X 系列的新一代视频编码标准 H. 265 作为视频监控的视频压缩标准。

2.2 新一代视频编码 H. 265 的技术分析

H.265 无论与之前的 H.264 相比还是与 MPEG 系列相比在编码性能上都有相当大的提升，这主要是新编码工具的使用和具备了特色的核心技术。H.265 具有不少新的编码技术，例如将块灵活的进行四叉树方式分割、帧内预测模式的角度增加、运动矢量预测具有自适应功能、离散余弦变换可以变化尺寸和最新的样点自适应补偿滤波器[17]。下面对新一代视频编码 H.265 进行主要技术分析：

（1）在编码器构架上基本与之前的 H.26X 构架基本相同，依旧是混合编码的方式。但是在每个模块上基本都添加了新的编码技术，主要的编码模块有帧内编码、帧间编码、变换量化、去方块滤波、样点自适应补偿、熵编码等。H.265 的编码框架如图 2.2 所示。

图 2.2 H.265 的编码框架

（2）H.265 在编码单元上有了很大的改进，它使用的编码树单元（CTU）和编码树块（CTB）作为编码单元，相对于传统的 H.264 的宏块而言具有更大的灵活性和高压缩性。如图 2.3 所示，可以对图像根据图像内容进行灵活分割，主要是 CTU 由编码单元

（CU）、预测单元（PU）、变换单元（TU）来组成。在 CTU 的分割上可以根据每一部分所携带的信息来决定作为什么单元。所以这种分割可以在携带信息少的画面或区域提高压缩效率。

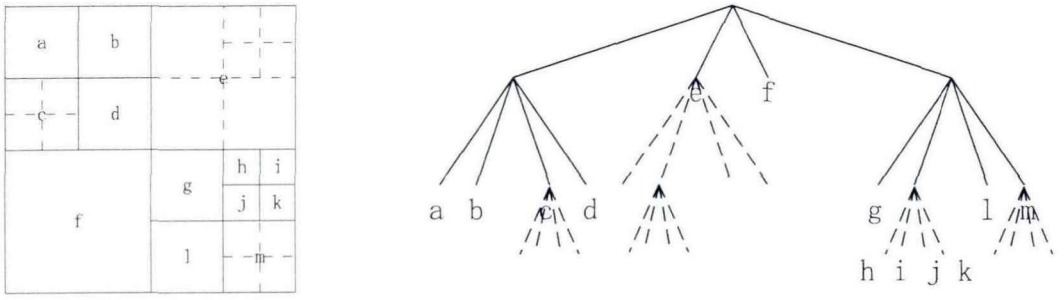

图 2.3 CTU 的分割

（3）在图像分割的技术上有了很大的改进，在传统的图像分割上基本为均匀分割，而在新的编码标准上引入了片（Slice）的概念，就是根据图像组的信息进行分割成片，片是由一个或者多个片段（SS）组成，而片段是由一个或多个 CTU 所组成。片的分割如图 2.4 所示。

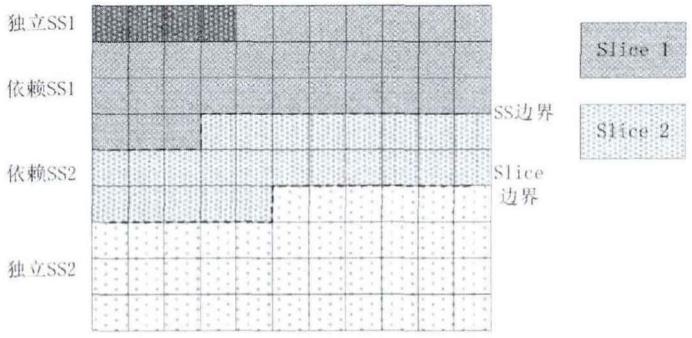

图 2.4 片的分割

（4）对于都要进行帧内与帧间预测的视频编码标准，H.265 在这两方面也做了相应的改进。帧内预测的技术相对更加准确，对于亮度信号进行 33 种角度预测和 DC 预测模式以及 Planar 预测模式。使视频图像信息中比较复杂的纹理能够进行更好的预测。对于帧间预测则引入了运动信息融合技术（Merge）和先进的运动矢量预测技术（AMVP）以及基于运动信息融合技术的 Skip 技术。增加的技术主要应用于对于相邻块间所存在空域以及时域的相关运动参数冗余。

（5）在 H.265 编码标准中的熵编码中使用的是将编码和模型相结合的基于上下文的自适应二进制算数编码(CABAC)。大幅度增加编码效率的原因是每一个符号的编码都与以前的编码结果有关，根据符号流的统计特性来自适应的为每一个符号分配码字。二进制化、上下文建模以及二进制算数编码构成了 CABAC 的整个编码过程。

（6）在 H.265 中引入了一种可以减少振铃效应的滤波方法像素自适应补偿技术（SAO）[18]。主要应用于滤波器之后来对像素值进行补偿与重构来达到减少振铃效应的

目的。主要实现是通过以 CTB 为基本单位，通过分类器对重建像素进行划分。然后对不同类别的像素值进行不同的补偿进而提高视频的质量。主要分为边界补偿和边带补偿两种补偿方式。

2.3 视频处理器实施方案对比与选择

视频监控系统中的视频图像编码主要由三部分组成。第一部分是视频信息采集模块（Sensor 模块），该模块主要任务是在 CMOS 或者 CCD 图像传感器中进行影像信号转变成光电信号。第二部分是视频信息处理模块（ISP 模块），该部分的主要任务是对已经采集的图像视频信号进行 bayer 至 RGB 再到 YUV 的转换，并且对其进行一系列的处理，比如 3A（自动白平衡、自动聚焦、自动曝光）处理、图像裁剪以及对图像亮度色度调节等等。第三部分是视频编码压缩模块（Video Encode），该部分的主要任务是把通过视频采集模块和视频处理模块的大量图像数据进行压缩处理，在能保证恢复我们所需要的图像数据信息的情况下便于我们对其进行传输和存储[19]。依据所述视频采集的重要组成部分和整理现阶段主要搭配方案，主要常用以下三种方案。

第一种方案是视频处理模块和视频压缩模块分别在两个不同芯片上的，如图 2.5 所示，视频处理模块可以用专用的 nextchip 的 NVP2400 芯片或者通过 FPGA 来处理视频图像。而视频压缩模块可以进行 ASIC 硬件编码或者通过软压缩（运行在硬件平台的代码实现）对图像信号进行压缩。

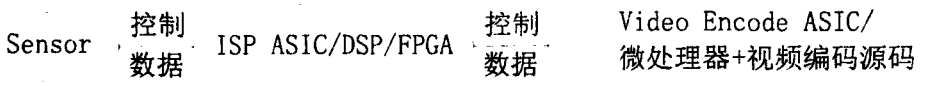

图 2.5 第一种搭配方案

第二种方案是视频处理模块和视频压缩模块集成在同一芯片中[20]。比如海思 3516、海思 3517 以及处理高清视频并支持 H.265 压缩编码的海思 3516A 芯片等多媒体处理芯片。所举以上这类芯片基本都有很多外围扩展接口，对我们进行芯片的开发提供了方便。方案搭配如图 2.6 所示。

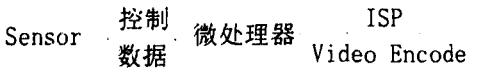

图 2.6 第二种搭配方案

第三种方案是可以对 FPGA 芯片进行开发来实现图像数据的处理和压缩，但是这种方案不仅开发成本高不易成功，而且难保证编码后的图像质量。方案搭配如图 2.7 所示。

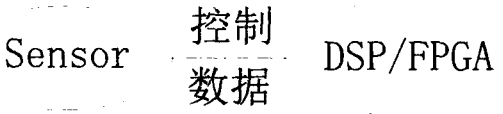

图 2.7 第三种搭配方案

本视频监控系统对视频编码设备的要求能够高效压缩图像信息并且通过无线信号进行传输。该系统对芯片要求较高，并且需要丰富的外围电路设计。所以整体考虑以上所列三种视频编码方案，选择第二种方案来进行该系统的完成。考虑到视频需要高编码率和芯片易于开发的要求，所以我们选择海思 3516A 来进行对视频信号编码处理。

2.4 网络视频传输主要技术方案对比与选择

在视频监控系统中，视频信号的传输处理是一个比较重要的环节。在视频传输的过程中需要选择合适的传输方式以及适应具体应用环境的协议。此节主要进行了在传输过程中主要应用方案的选择。

在 3G 网络出现之前，传统视频监控系统中是通过有线电缆进行视频的传输。通过有线视传输的视频监控系统必然带来复杂的布线和较差的灵活性。所以 3G 信号开启了无线视频监控系统的大门。由于无线视频监控系统相比传统视频监控系统具有低成本、高灵活性、方便维护以及较好移动性等优点，使无线传输方式在视频监控系统中很快的普及。但是在利用 3G 信号进行视频传输的同时会出现较高的误码率，并且稳定性也不是很好。加上 3G 信号应用中带宽的限制不足以支持高分辨率图像的传输，所以选择近年来普及的 4G 信号具有更大的优势[21]。具体对比数据如表 2.1 所示。

表 2.1 3G 信号与 4G 信号网速对比

3G 和 4G 理论网速对比（换算成生活中的下载速度需将以下数值除以 8）					
标准	3G			4G	
制式	TD-SCDMA	WCDMA	CDMA2000	TD-LTE	FDD-LTE
下载速度	2.8Mbps (358.4KB/S)	14.4Mbps (1.8MB/S)	3.1Mbps (396.8KB/S)	100Mbps (12.5MB/S)	150Mbps (18.75MB/S)
上传速度	2.2Mbps (281.6KB/S)	5.76Mbps (737.2KB/S)	1.8Mbps (230.4KB/S)	50Mbps (6.25MB/S)	40Mbps (5MB/S)

提到无线传输的应用，不得不联想到如今无所不在的 WLAN，也就是无线局域网络。无线局域网具有三种形式的构成，分别是点对点型、点对多点型和完全分布型。从 WLAN 的构成和分布可以看出具有无线传输的灵活性高、免去复杂布线以及成本低的普遍性优点。并且对于没有 4G 网络接收能力的电脑等应用设备也可方便的应用无线传输与接收。但毕竟是应用在局域网中的，并且还不能够完全脱离有线网络。对于现在需要高智能化和高移动性的无线视频监控系统来说，覆盖面广的 4G 网络更适合用于无线视频监控系

统。

4G 网络技术全称为第四代移动电话行动通信标准，也就是指第四代无线蜂窝电话通讯协议，是将 3G 网络与 WLAN 优点相结合到本身，能够传输与高清晰度电视差不多的高质量视频图像的技术产品。发展到现在有两种制式，分别为 TD-LTE 与 FDD-LTE。通过 3G 网络和 4G 网络速率的对比可以明显看出。4G 网络在速率上具有明显的提升，令其可以支持高分辨率视频的传输。4G 信号所具有的优点如下所述：

（1）较高的速率。上表可看出 4G 网络在速率上具有很大的优势，具有较高的传输速率。下行速率的峰值可达到 150Mbps，上行速率的峰值也可达到 50Mpbs。

（2）较好的兼容能力。具有开放接口并且真正意义上实现全球标准化服务。由于 4G 信号可以兼容之前的 2G 信号和 3G 信号，所以所有移动通信用户都可以使用 4G 网络信号。

（3）较强的灵活性。由于 4G 网络信号采用智能技术，能自动适应资源分配。所以用户在使用 4G 信号时可以享受到其针对通信中变化的业务需求而采取相应的处理。

（4）支持多类型用户。4G 网络信号可以通过动态的网络和不一样的信道采取自适应技术。能够令无论高速还是低速的不同类型的用户设备共存和互通。

（5）支持业务广泛。由于 4G 网络具有较高的速率。同时带来了广泛的业务应用。例如语音通话、视频会议、移动采访以及高清视频监控等。

4G 信号无论从速率还是应用都有较强的优势，特别是全球标准化服务覆盖面广的特性使用户可以在不同的地点都可以使用 4G 进行他们所需要信息的采集[22]。所以对主流的无线信号进行对比分析，我们采用 4G 网络信号来作为本视频监控系统的传输介质。

2.5 本章小结

本章首先对历代压缩编码进行了简介与分析，然后对主流的视频编码标准进一步的进行了分析与对比，选出适合本系统的新一代的压缩算法 H.265 作为本系统对采集到的视频压缩编码算法。并对 H.265 进一步的研究和使用。对视频处理器的实施方案进行研究和对比，从三种主流方案中选择一种最适合本系统的进行开发。无线监控系统中避免不了传输介质，对可用的几种传输方式进一步的研究对比选出最合适的传输方法进行开发使用到本系统中。

第 3 章 视频监控系统整体设计与硬件设计

3.1 系统整体设计

近年来，随着社会经济的发展，安防的重要性也随之提升。无论在校园、公路、停车场还是仓库，都需要较强的安全防范。而视频监控系统不仅能防患于未然，还能对出现犯罪现象后进行后期的跟踪调查并提供线索。本系统根据具体要求，设计了一套基于 H.265 的无线视频监控系统。本设计主要由两部分组成。第一部分是由视频采集模块、视频编码模块和视频传输模块所构成的监控服务器。该部分主要完成的任务是通过视频采集模块进行视频的采集以及对视频进行格式的转换，然后通过视频编码模块对采集到的视频数据进行压缩以便减少在视频传输模块传输数据的数据量。第二部分主要由视频接收模块、视频解码模块、视频显示模块、视频录像存储模块以及目标检测模块所构成的监控客户端。此部分的主要完成的任务是接收监控服务器所传输过来的视频数据，并完成对数据进行解码、显示、存储和目标检测等主要功能。本设计的监控服务器端所用硬件平台为微智高科公司所提供的海思 3516a 开发板。此开发板以芯片 Hi3516A 作为主要控制器，该控制器基于 Cortex-A7 内核，具有高达 600MHz 的主频率，具有很快的运行速度[23]。并且包含分别具有 32KB 的指令缓存和数据缓存，它的二级缓存为 128KB,Hi3516A 芯片具有重要的特点之一就是不仅支持主流的 MJPEG 编码、JPEG 编码和 H.264 编码。还支持新一代的编码标准 H.265 编码。该设计的摄像头采用的 OV3640，该摄像头可以支持 1080P 以内的编码。满足该设计对采集到视频分辨率的要求。视频的传输采用上海域格信息技术有限公司的 4G 模块 CLM920_CN3 模块，CLM920_CN3 模块可以达到传输下行速率为 150Mbps 的峰值，上行速率为 50Mbps，可以作为本系统的传输模块。本设计所选 PC 机作为客户端的宿主机，整体框架如图 3.1 所示。

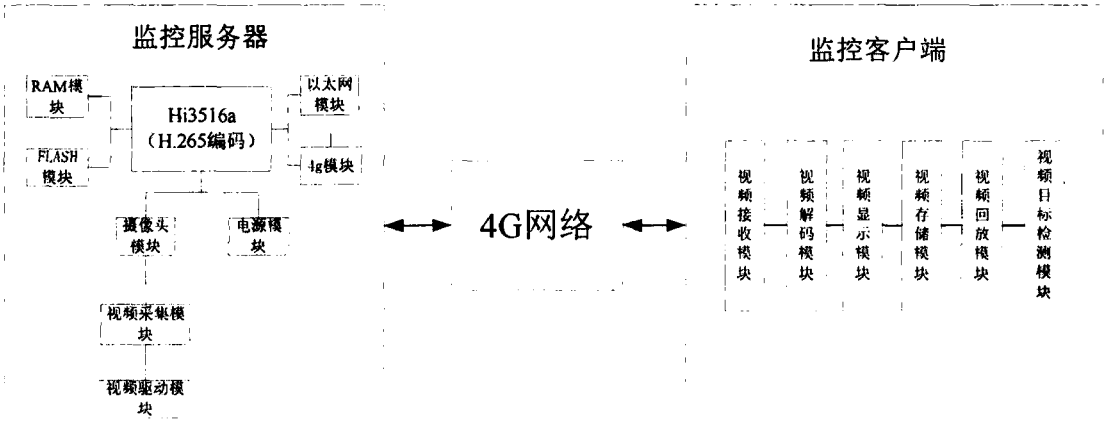

图 3.1 系统整体框架

3.2 主要硬件电路设计

本系统的设计是围绕 Hi3516A 进行扩展开发，如图 3.2 所示，外围模块主要包括 COMS 摄像头模块、内存 RAM 模块、电源电路模块、Flash 模块、以太网接口模块、4G 传输模块和 SD 卡模块等。在本设计中的 Hi3516A 芯片即支持 16bitDDR3 又支持 32bitDDR3,在 32bitDDR3 的模式下可以发挥更好的性能，所以我们采用海力士公司的 H5TC2G63FFR 芯片来进行设计的实现。由于 H5TC2G63FFR 芯片数据总线位宽为 16bit。所以我们采用共用地址总线和控制总线的两片 H5TC2G63FFR 芯片相连作为内存 RAM。我们所有所需要运行的程序都是在这两片内存 RAM 中运行。FLASH 模块在该系统中主要任务是保存系统在运行时产生的数据以及系统运行所需要的操作系统和应用程序。Hi3516A 支持 SPI FLASH 与 NAND FLASH 两种接口，本系统使用荣晟泰科技有限公司的 mx25L122804 芯片来实现 FLASH 模块功能。4G 传输模块使用上海域格信息技术有限公司的 CLM920_CN3 模块，该模块支持移动联通电信三大通信运营商，也同时支持 2G、3G 和 4G 通信。CMOS 采集模块选择的是型号为 OV3640 的摄像头,具有 400 万像素功能[24]。本系统中的以太网接口模块是使用的 DAVICOM 公司的能够 10/100M 自适应收发数据功能的 DM9000 芯片[25]。监控服务端的主要芯片如图 3.3 所示。

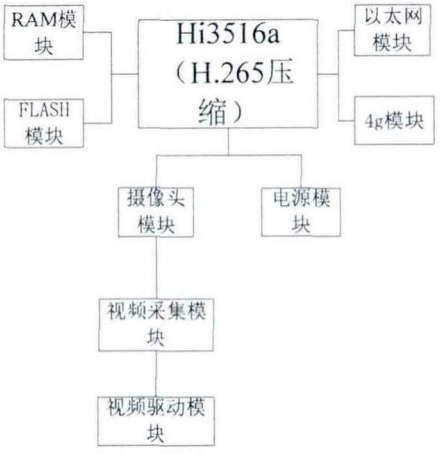

图 3.2 监控服务端

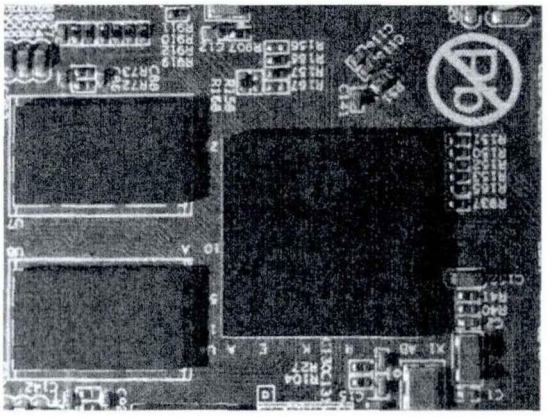

图 3.3 Hi3516a 芯片

3.2.1 Hi3516A 控制芯片

Hi3516A 是海思半导体开发的一款专业高端芯片。它是基于 Cortex-A7 内核并且主频率最高可以达到 600MHz 的微处理器。运行速度完全可以满足视频处理的要求。在视频编码和处理性能上，Hi3516A 可以进行 H.264/H.265 编码最大分辨率为 5M Pixel 的视频图，并且具有多码流实时编码能力[26]。Hi3516A 作为新一代摄像机芯片，集成新一代 ISP，采用业界最新的 H.265 视频压缩编码器，同时采用先进低功耗工艺和低功耗架构设计，这一切将使得 Hi3516A 在低码率、高图像质量和低功耗方面持续引领行业水平。创新性的硬件支持 90 度/270 度旋转功能和镜头几何校正功能，可以满足监控应用的各种场景需求。Hi3516A 还全格式支持 3A 算法，用户可以基于此实现包含一体机机芯在内

的各种机型设计。集成 POR、RTC、Audio Codec、并支持多种 sensor 电平及各种时钟输出等功能，将极大的降低基于 Hi3516A 的摄像机成本。与海思 DVR/NVR 芯片一样稳定和易用的 SDK 设计，能够支撑客户快速产品量产，并实现 DVR/NVR 和 IP 摄像机的系统布局。 不仅如此，Hi3516A 在智能视频分析方面也有很大的进步，由于 Hi3516A 集成了智能分析加速引擎，使其不仅支持智能运动侦测、周界防范还能进行视频诊断等多种智能分析应用。由于 Hi3516A 各方面具有的特点，使其应用面非常广泛。在医疗、学校、车载、通讯以及安防等方面都有其用武之地。功能框图如图所示，基于 Hi3516A 控制器的核心控制板如图 3.4 所示。

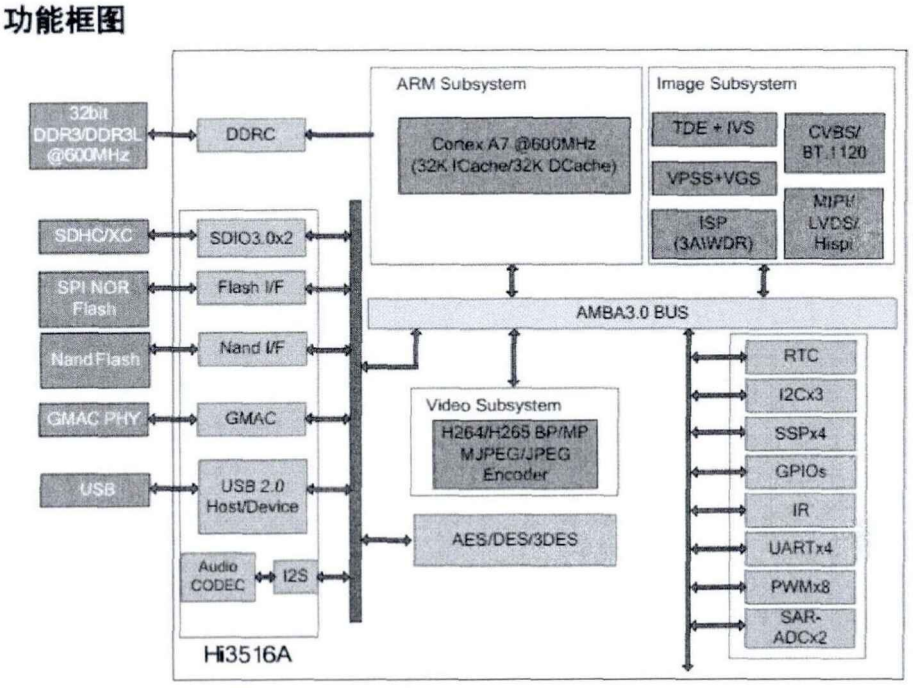

图 3.4 Hi3516a 功能框图

3.2.2 CMOS 摄像头模块

基于 Hi3516A 处理器的核心控制板的 CAMERA 接口支持 ITU RBT-601/656 模式、DMA 模式等多种输入模式，并且支持最大输入分辨率为 8192x8192 的图像。CAMERA 接口具有数据缩放、编解码/预览图像镜像、视频同步信号的可编程极性等功能。同时，CAMERA 接口支持有 XY 翻转、90° 旋转、180° 旋转和 270° 旋转等旋转功能。CAMERA 接口支持图像捕捉帧控制功能，可以通过接入该接口的摄像头进行图像捕捉，CAMERA 接口还支持扫描线消除、支持 LCD 控制器直接路径、支持交错 CAMERA 输入等功能。因此，把 CMOS 摄像头接入 CAMERA 接口进行图像的采集。

本文使用的 CMOS 摄像头是 OV3640，具有 300 万像素，满足本文采集的图像分辨率的要求。CMOS 摄像头模块电路如图 3.5 所示：

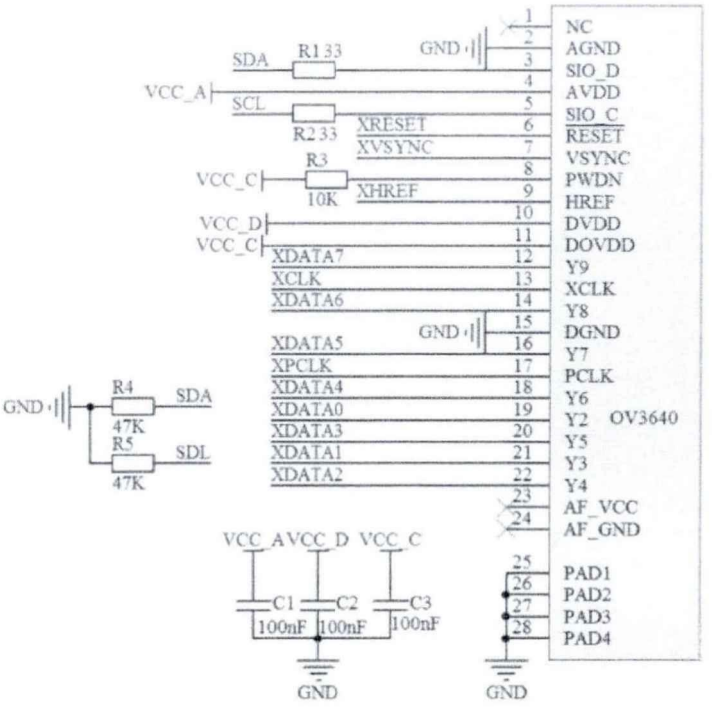

图 3.5 摄像头模块电路

3.2.3 4G 传输模块电路

本系统所采用的 4G 模块是上海域格信息技术有限公司的 CLM920_CN3 模块。CLM920 系列模块是一个 PCI Express Mini Card 1.2 标准的模块，基带芯片采用高通的 MDM9X15，支持的主要操作系统有 Windows 7、Windows 8、Android 4.0 等。CLM920_CN3 模块支持 TDD-LTE、TD-SCDMA、GGE 和 FDD-LTE 等通信方式。4G 信号的 TDD-LTE 模式和 FDD-LTE 都支持并且最大速率分别为 UL18Mbps/DL61Mbps 和 UL50Mbps/DL100Mbps，3G 信号 TDD-SCDMA 的 HSUPA+支持的最大速率为 UL2.2Mbps/DL4.2Mbps。

CLM920_CN3 模块按模块功能可以分为射频收发单元、存储单元、基带处理单元和电源管理单元。硬件整体系统结构如图 3.6 所示：

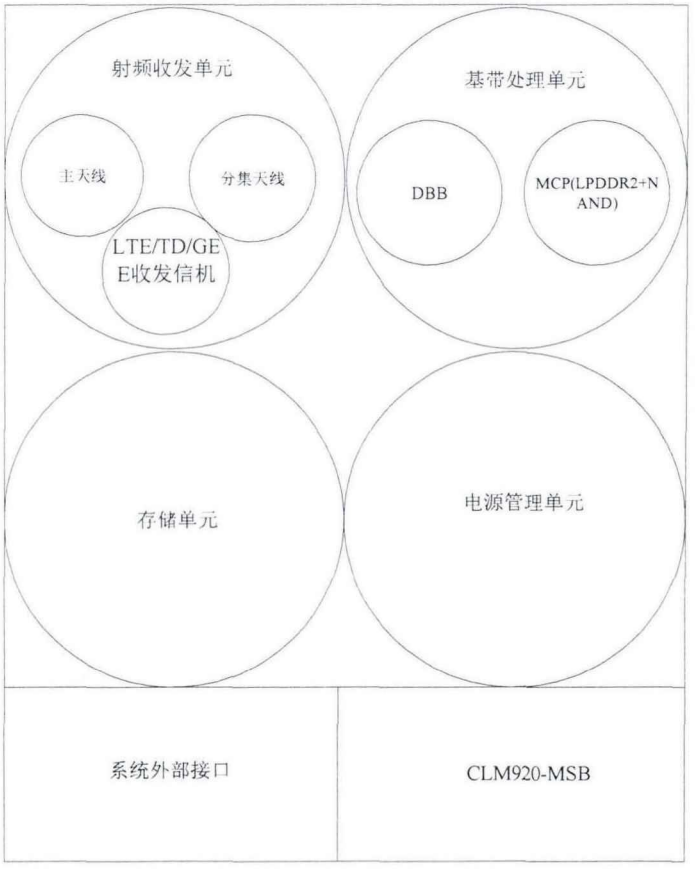

图 3.6 4G 模块系统结构

CLM920_CN3 模块主要包含的交互接口有电源接口、USB 接口、USIM 接口、休眠唤醒接口、WWAN 接口、LED 接口和射频天线接口等。CCN3 模块可以通过一路的 USB 模块进行与主控制芯片进行大量的信息传输。LM920 系列 4G 模块的电源接口采用单电源供电模式，供电范围为 3.3V~4.2V。USB 模块支持 USB2.0 高速协议并且支持一路 USB 接口，USB 通用串行总线控制器主要用于与外围 USB 主机之间的大批量高速数据交换。该 USB 接口支持 USB 2.0 高速协议。并且 USB 接口还能够根据信息包的动态分配 FIFO 存储空间的功能。CLM920_CN3 模块具有一个兼容 ISO 7816-3 标准的 USIM 卡接口，可以实现和 SIM 卡交换数据。并且 CLM920_CN3 模块可以实现下行速率 150Mbps 以及上行速率达到 50Mpbs,能够实现本设计的内容。CLM920_CN3 模块如图 3.7 所示。

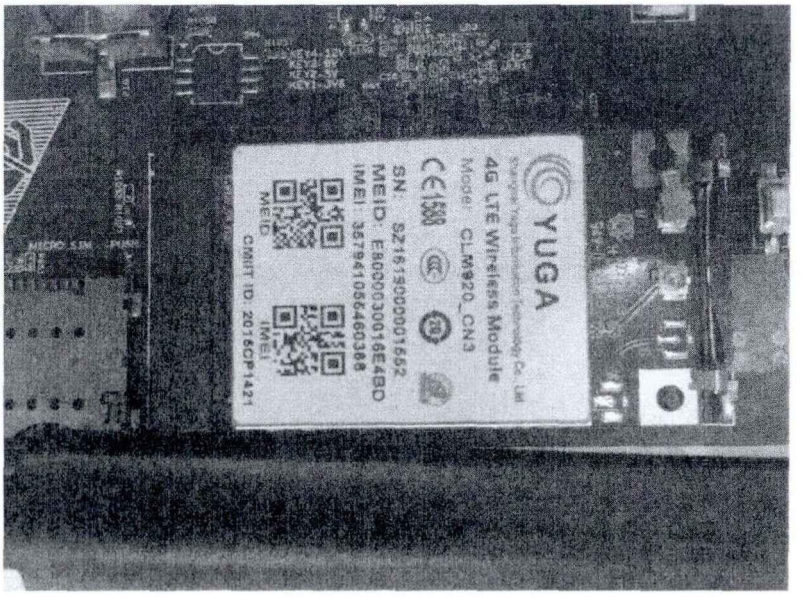

图 3.7 CLM920_CN3 模块

3.3 本章小结

本章首先根据第二章中对系统中所用主要技术以及主要模块搭配方案的研究和结果设计出整个视频监控系统的搭建方式，并根据整体设计方案设计出主要的硬件电路。其中最主要的是对处理器硬件电路的设计，选用 Hi3516A 作为主控制器，然后根据主控制其的电路来进行对 4G 模块和 COMS 摄像头的选择和电路设计。完成了体统的整体设计和硬件电路的设计与搭配。

第 4 章 嵌入式软件开发环境的搭建

4.1 交叉编译环境的搭建

由于 Hi3516a 芯片的资源以及运算能力的限制，不能够像计算机一样直接进行程序的开发编译。因此，采用在计算机的虚拟机上进行软件的编译和在目标机上进行软件的运行来共同进行程序的软件开发。在计算机的宿主机上安装交叉编译环境 arm-hisiv300-linux，然后编译出可以在运行在目标机的可执行程序。交叉开发模式如图 3.1 所示。

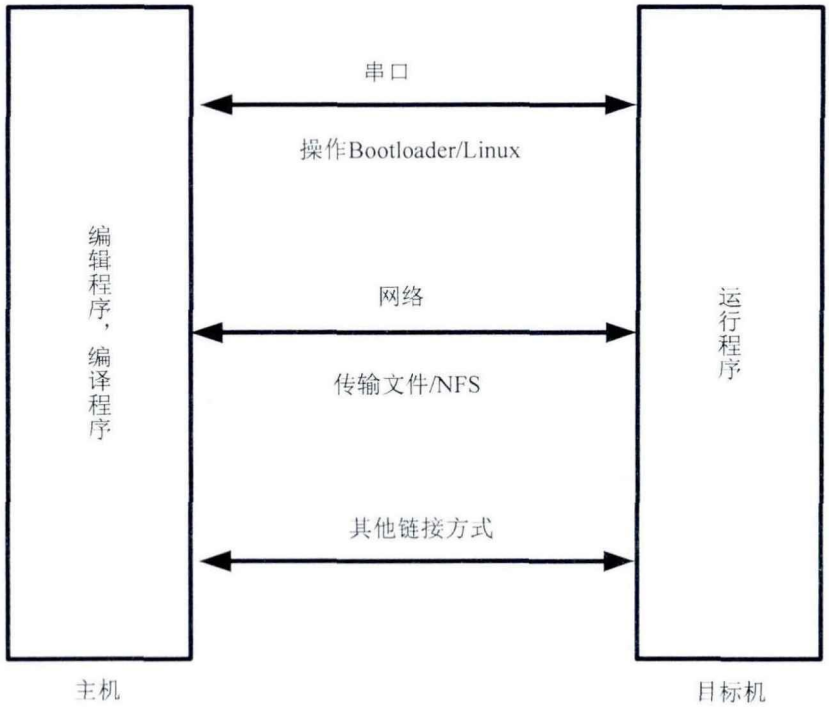

图 4.1 交叉开发模式

首先在 ThinkVision 计算机上下载安装由 Vmware 公司所开发的一个专用的虚拟机 VMware，并在上面搭建虚拟机软件 ubuntu 系统，并且需要在 ubuntu 系统上安装软件开发所需的交叉编译工具链来进行程序的编辑以及编译。最后依靠串口、网络或者其他传输方式把在虚拟机上编译好的可执行程序传输至目标机 Hi3516a 处理器上运行。本系统开发所用的交叉编译工具链为海思半导体公司所提供的。该公司提供了两种交叉编译工具链 arm - hisiv300 - linux 和 arm - hisiv400 - linux [27]。本系统选择的是 arm - hisiv300 - linux 作为交叉编译工具链。在 ubuntu 上安装 arm - hisiv300 - linux 的步骤如下:

(1)在装有 linux 系统 ThinkVision 计算机上建立一个路径/usr/local/arm，然后通过命令 # tar - xzvf arm - hisiv300 - linux.tar. bz2 将需要用的交叉编译工具链 arm - hisiv300 - linux 解压到相对应的目录下。

(2)回到主目录下，执行命令 sudo ./cross/install 来进行交叉编译工具的安装。

(3)配置环境变量（以免再次使用时人工配置环境变量），把交叉编译器的路径加入到

PATH。利用 gedit 命令打开/ect/environment,在里面加上 arm-hisiv300-linux 的安装路径 /opt/hisi-linux/x86-arm/arm-hisiv300-linux/target/bin。然后执行 source /etc/profile，安装交叉编译器的脚本配置的环境变量就可以生效了。

(4)测试交叉编译器 arm-hisiv300-linux -gcc -v-4.8.3 有没有成功安装：

arm-hisiv300-linux- gcc -v

运行命令后，看到在命令运行之后出现所需要的 arm-hisiv300-linux 的版本信息如图所示 gcc version 4.8.3,显示表明交叉编译工具链 arm-hisiv300-linux 已经成功安装可用。

```
wen@ubuntu:~$ arm-hisiv300-linux-gcc -v
Using built-in specs.
COLLECT_GCC=arm-hisiv300-linux-gcc
COLLECT_LTO_WRAPPER=/opt/hisi-linux/x86-arm/arm-hisiv300-linux/bin/../libexec/gc
c/arm-hisiv300-linux-uclibcgnueabi/4.8.3/lto-wrapper
Target: arm-hisiv300-linux-uclibcgnueabi
Configured with: '../gcc-linaro-4.8-2013.12/configure' --host=i386-redhat-linux
 --build=i386-redhat-linux --target=arm-hisiv300-linux-uclibcgnueabi --prefix=/ho
me/sying/wucaiyuan_toolchain/v300/uclibc_gcc4.8_linaro_toolchain_optimized/insta
ll/arm-hisiv300-linux --enable-threads --disable-libmudflap --disable-libssp --d
isable-libstdcxx-pch --with-arch=armv5te --with-gnu-as --with-gnu-ld --enable-la
nguages=c,c++ --enable-shared --enable-lto --enable-symvers=gnu --enable-__cxa_a
texit --enable-nls --enable-clocale=gnu --enable-extra-hisi-multilibs --with-sys
root=/home/sying/wucaiyuan_toolchain/v300/uclibc_gcc4.8_linaro_toolchain_optimiz
ed/install/arm-hisiv300-linux/target --with-build-sysroot=/home/sying/wucaiyuan_
toolchain/v300/uclibc_gcc4.8_linaro_toolchain_optimized/install/arm-hisiv300-lin
ux/target --with-gmp=/home/sying/wucaiyuan_toolchain/v300/uclibc_gcc4.8_linaro_t
oolchain_optimized/install/host_lib --with-mpfr=/home/sying/wucaiyuan_toolchain/
v300/uclibc_gcc4.8_linaro_toolchain_optimized/install/host_lib --with-mpc=/home/
sying/wucaiyuan_toolchain/v300/uclibc_gcc4.8_linaro_toolchain_optimized/install/
host_lib --with-ppl=/home/sying/wucaiyuan_toolchain/v300/uclibc_gcc4.8_linaro_to
olchain_optimized/install/host_lib --with-cloog=/home/sying/wucaiyuan_toolchain/
v300/uclibc_gcc4.8_linaro_toolchain_optimized/install/host_lib --with-libelf=/ho
me/sying/wucaiyuan_toolchain/v300/uclibc_gcc4.8_linaro_toolchain_optimized/insta
ll/host_lib --enable-libgomp --disable-libitm --enable-poison-system-directories
 --with-libelf=/home/sying/wucaiyuan_toolchain/v300/uclibc_gcc4.8_linaro_toolcha
in_optimized/install/host_lib --with-pkgversion=Hisilicon_v300 --with-bugurl=htt
p://www.hisilicon.com/cn/service/claim.html
Thread model: posix
gcc version 4.8.3 20131202 (prerelease) (Hisilicon_v300)
wen@ubuntu:~$
```

图 4.2 交叉编译工具链成功安装

4.2 嵌入式 linux 中 Bootloader 的移植

在个人计算机上，系统的启动需要 BIOS 来引导。同样的在嵌入式 Linux 系统中，通过引导加载程序 Bootloader 来进行 Linux 系统的加载和启动。Bootloader 的主要任务有三个，分别是完成目标端硬件设备的初始化、软件环境中内存的映射使其被别的模块调用以及系统内核的调用。除此之外，Bootloader 还具有利于开发的控制命令[28]。比如镜像烧写以及设备状态查看等。嵌入式 linux 系统开发中经常使用的 Bootloader 是 u-boot、vivi 以及 Redboot，本系统所使用的是通用的 u-boot。u-boot 的主要目录结构如表 4.1 所示。

表 4.1 u-boot 的主要目录结构

目录名	描述
arch	各种芯片架构的相关代码、U-boot 入口代码。
board	各种单板的相关代码，主要包括存储器驱动等。
board/hi3516a	Hi3516A 单板相关代码。
arch/xxx/lib	各种体系结构的相关代码，如 ARM、MIPS 的通用代码。
include	头文件。
include/configs	各种单板的配置文件。
common	各种功能（命令）实现文件。
drivers	网口、Flash、串口等的驱动代码。
Net	网络协议实现文件。
fs	文件系统实现文件。

根据海思半导体公司提供的 Hi3516a 的 u-boot 移植应用开发指南，主要步骤如下所述：

(1)编译环境的配置：为了使其能应用在 Hi3516A 中，通过执行命令 make ARCH=arm CROSS_COMPILE=arm-hisiv300-linux-hi3516a-config 来进行 u-boot 参数的配置。

(2)u-boot 的编译：执行命令 make ARCH=arm CROSS_COMPILE=arm-hisiv300 -linux-编译源代码。编译成功后，在 U-boot 目录下生成 u-boot.bin 文件(此文件是中间件，不是最终执行的 U-Boot 镜像)。

(3) DDR 存储器的配置：打开目录 osdrv/tools/pc_tools/uboot_tools/，根据目录下的配置表格，对其中的标签页 ddrc0-init 进行修改。

(4) 最终使用的 u-boot 镜像的生成：第三步中完成配置表格的修改后，然后保存表格。单击表格第一个标签页上的按钮 Generage reg bin file 后生成临时文件 reg_info.bin。将临时文件 reg_info.bin 和编译 u-boot 得到的 u-boot.bin 都拷贝到 SDK 中的 osdrv/tools/pc/uboot_tools/ 目录下，最后执行命令：mkboot.sh reg_info_hi3516a.bin u-boot-hi3516a.bin 最后生成的 u-boot-hi3516a.bin 就是能够在单板上运行的 U-boot 镜像。

4.3 嵌入式 linux 中内核的移植

嵌入式 Linux 在开发过程中是整个系统的主要软件开发平台，在整个系统开发中占据着重要的地位。Linux 内核主要由内存管理、CPU 和进程调度、虚拟的文件系统、设备管理和驱动、通信进程这五大部分组成[29]。根据表格 4.2 中 Linux 内核的目录结构，对所需要的硬件开发平台进行 Linux 内核的配置以及裁剪，这样即可以实现我们所需要的功能又能减少相应的内核代码，以便于缩小内核并将嵌入式系统的资源利用最大化[30]。本系统根据系统所需裁剪了 Linux3.4.35 内核进行，大量的减少 Linux3.4.35 内核的代码量，最后将经过裁剪的 Linux3.4.35 内核移植至基于 Hi3516A 的处理器开发平台里。

表 4.2 Linux 内核目录结构

目录	解释说明
/include	建立内核代码时所需的大部分包含文件
/init	内核的初始化代码
/arch	所有硬件结构特定的内核代码
/drivers	内核中所有的设备驱动程序
/fs	所有的文件系统的代码
/net	内核的连网代码
/mm	所有内存管理代码
/ipc	进程通信代码
/kemel	主内核代码

内核 Linux3.4.35 对应的定制移植和编译的步骤如下：

首先对内核 Linux3.4.35 进行针对于本系统需要进行对应的配置：通过命令 cp arch/arm/configs/hi3516a_full_defconfig.config 复制已经解压的 Hi3516A 硬件平台所需配置文件到内核 Linux3.4.35 主目录中。完成这项工作所使用的。完成上步后运行 ARCH=arm CROSS_COMPILE=arm-hisi300-linux- menuconfig 命令后选择所需模块并进行保存。执行命令 make menuconfig 显示内核配置界面如图 4.3 所示。

对内核进行编译：当进行了内核的配置后，直接运行命令 make ARCH=arm CROSS_COMPILE=arm-hisiXXX-linux- uImage 命令进行内核的编译。等大约两分钟后内核生成镜像文件为 uImage。

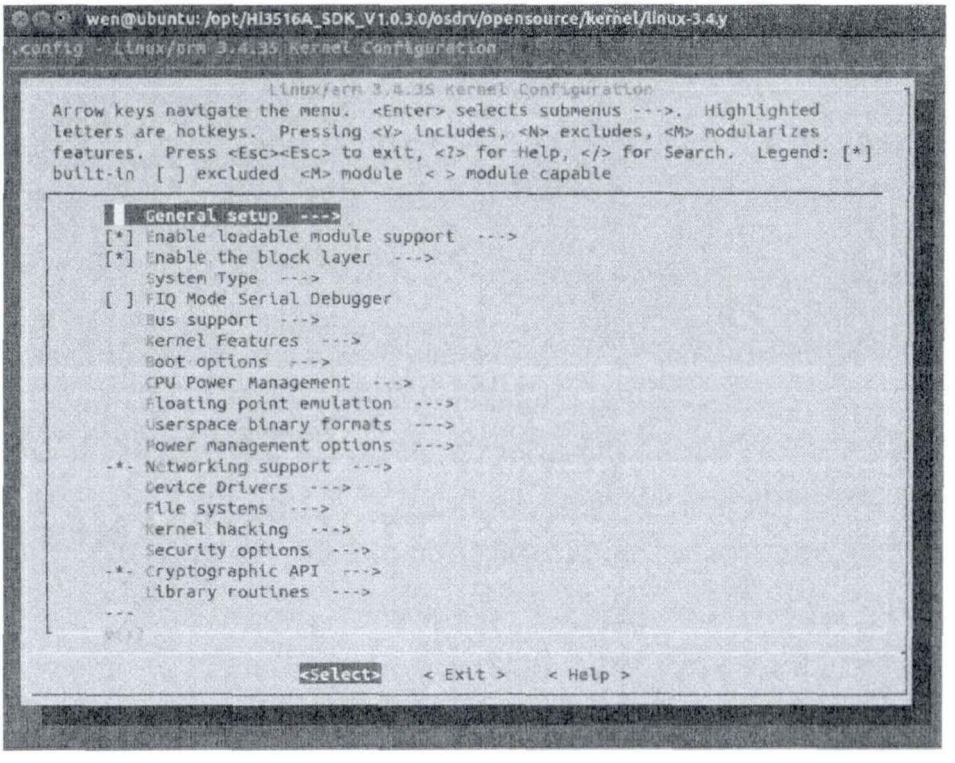

图 4.3 编译内核

4.4 嵌入式 linux 中根文件系统的制作

当内核配置编译后，就需要对根文件系统进行制作。根文件系统是在文件系统中的顶端挂载。在个人计算机 windows 系统中，我们的磁盘分为 C 盘、D 盘、E 盘以及 F 盘来进行文件的存储。同样的在 linux 系统中，文件是根据根文件系统的目录分区来进行存储的，其独有的树形结构使根文件系统更利于数据的存储与管理。而根文件系统中的文件包含了用户和系统所需要的应用程序、配置文件以及保护的信息等。Linux 可用很多种文件系统，例如 YAFFS（YAFFS2）、RAMDISK、EXT2（EXT3）、Gramfs 以及 JFFS2 等文件系统。其中 YAFFS2 文件系统具有易移植且高性能的优点，并且主要为存储设备 Nand Flash 进行制作的，能够用于大页（2KB/page）的 Nand Flash[30]。所以本系统采用通常使用的 busybox（版本为 busybox-1.20.2）来进行根文件系统 YAFFS2 的制作。Linux 根文件系统中主要包含的目录如表 4.3 所示。

表 4.3 根文件系统目录结构

目录	内容
bin	必要的用户命令
etc	系统的配置文件
dev	设备文件
home	用户主目录
mnt	挂载点
lib	必要的链接库
opt	附加的软件套件
proc	提供进程信息和内核的虚拟文件系统
sbin	必要的系统管理命令
root	root 用户的主目录
tmp	暂时的文件
usr	用户的大量应用程序的文件
var	存放可变数据

根文件系统的制作步骤：

(1)运行命令 mkdir -p /nfsroot/rootfs 来制作 YAFFS2 文件系统的主目录 rootfs，并在 rootfs 主目录下创建 YAFFS2 文件系统必须安装的子目录。比如 bin、dev、lib、proc、home、sys、mnt 等子目录。

(2)下载制作根文件系统所需的 busybox 源码并对其进行解压，然后对 busybox 源码目录的 Makefile 文件进行修改，复制 busybox_cfg_hi3516a_v300 的配置文件至 busybox 的主目录下，并将其命名为.config。运行命令 make menuconfig，进行对 busybox 进行配置。如图 4.4 所示为 busybox 配置界面。

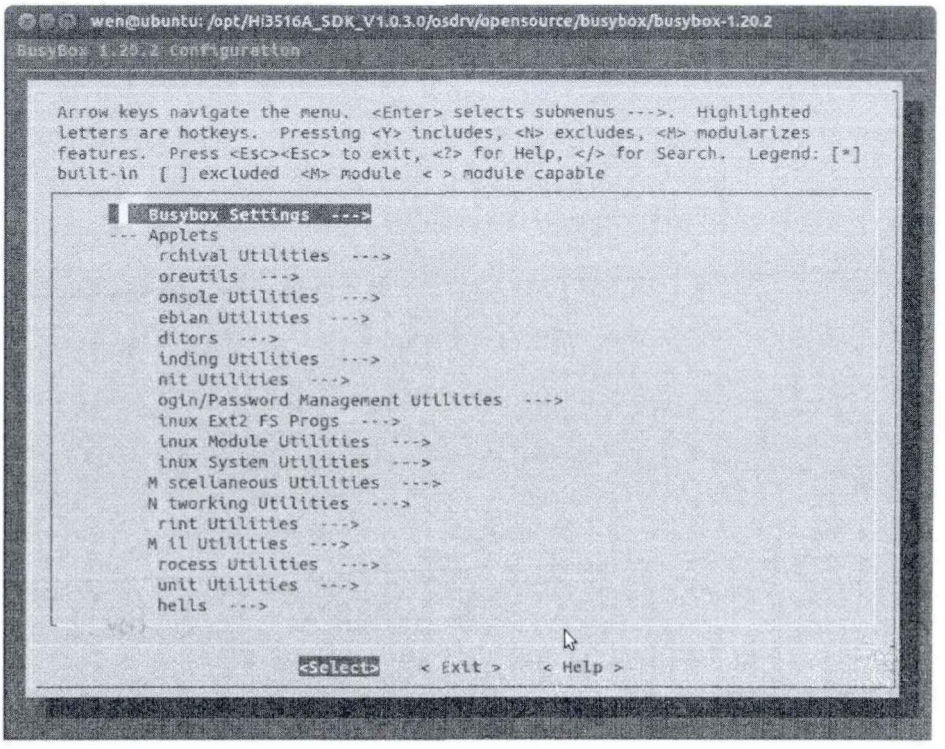

图 4.4 busybox 配置界面

大部分的 busybox 的选项都默认选择，但是 busybox installation prefix 是指定所安装的路径，所以要令 YAFFS2 根文件系统中 rootfs 的绝对路径作为安装路径[32]。以上工作做完后，运行 make 和 make install 进行 busybox 的安装与编译。在 busybox 目录下的_install 目录下生成以下目录及文件：bin、sbin、linuxrc -> bin/busybox 以及 usr。将生成的文件全复制在新建目录 rootbox 中。

(3)通过如下图所示命令实现在 rootbox 目录中添加 etc dev tmp lib var home mnt proc 目录，同时在 etc、dev、lib 的目录中添加所用文件。这些文件包含系统配置文件、库文件、设备文件等。能够令文件系统在嵌入式平台上稳定运行。

```
hisilicon$mkdir rootbox
hisilicon$cd rootbox
hisilicon$-R packet/os/busy/box_1.20.2/_install/*.
hisilicon$mkdir etc dev lib tmp var mnt home proc
```

(4)利用编译工具 makeyaffs2image 来进行 YAFFS2 文件系统镜像的制作，运行命令 mkyaffs2image ./rootbox yaffs2-root.img pagesize ecctype 生成根文件系统镜像文件 yaffs2-root.img。

4.5 本章小结

本章主要对系统的服务器端嵌入式软件进行开发，首先完成了交叉编译环境的搭建可以在 PC 机的虚拟机上进行对嵌入式板端的操作。然后完成了对嵌入式 Linux 的 Bootloader 进行开发移植。还完成了对 Linux 内核的移植以及根文件系统的制作。

第 5 章 软件的整体设计

5.1 媒体处理软件的开发

5.1.1 系统媒体处理概述及控制

在视频监控系统中，对服务器端媒体处理软件的开发关系着视频图像的采集和传输。而视频图像的采集和传输是系统中至关重要的两个部分。由于本系统用的主要处理器为 Hi3516A，所以对媒体软件的开发主要根据海思半导体公司所提供的针对媒体开发的软件处理平台（Media Process Platform）。英文简称为 MPP（以下所述不特加说明都是指海思媒体软件处理平台）[33]。在本系统中通过对 MPP 的开发来实现对视频信号的采集、编码以及传输。

MPP 的一大优势就是能够快速开发软件。在应用软件中，不可避免的会出现一些与芯片有关的底层处理，并且这些底层处理是相对复杂的。MPP 不仅对其屏蔽，而且还具有 MPI（MPP Programe Interface）接口来实现应用软件所需功能。MPP 不仅能够加快开发视频图像的采集、编码（支持的格式有 H.265/H.264/MJPEG/JPEG/MPEG4）、解码（支持的格式有 H264/VC1/MPEG4/MPEG2/AVS）以及显示输出。还能对视频图像前处理（图像去噪、图像 Deinterlace、图像增强、图像锐化）、智能分析、编码码流叠加 OSD、视频侦测分析等模块进行加速开发[23]。

针对 Hi3516A 的芯片特点，系统控制主要做的工作有三点。第一是使硬件进行复位和初始化。第二是初始化媒体处理平台系统的所有模块、提供版本信息以及管理业务模块的工作状态。第三是对大块物理内存进行管理。无论在应用程序启动前还是退出后，都要进行对媒体处理平台系统进行初始化来释放资源。在系统控制中提供的对大块物理内存的管理是依靠视频缓存池来进行的。这里所说的视频缓存池是由一组一样大小的缓存块而构成并且这些缓存块具有连续的物理地址。在系统中的主要任务是对内存进行合理的分配和回收。视频输入模块 VI 通过公共视频缓存池 A 中得到一块视频缓存块 Bm，这一步的主要目的是对图像信息进行保存。然后再将视频缓存块 Bm 传送给视频处理模块 VPSS 并对视频信号进行图像加强、去噪以及锐化的功能。在经过必要的视频处理后，根据视频信号的需求利用 bypass 通道将视频缓存块 Bm 发送给视频解码（VDEC）模块或者视频侦测分析(VDA)模块以及视频输出(VO)模块来进行对视频信号相应的视频编码、视频侦测以及视频输出。当处理完成后，释放视频缓存块回到视频缓存池中等待再利用。这样就实现了内存的合理利用和分配回收。

系统控制模块是媒体控制软件开发很重要的一步，对视频缓存池的管理是通过对结构体 VB_CONF_S 的定义和设置来进行的。主要代码如下：

```
typedef struct hi VB_CONF_S{
        HI_U32 u32Max Pool Cnt;//最大允许缓存池个数（本系统中设置为
128）
         Struct hi VB_CPOOL_S{
            HI_U32 u32Blk Size;//缓存块大小（本系统中设置为 601344）
             HI_U32 u32Blk Cnt;//缓存块个数（本系统中设置为 20）
            HI_CHAR ac Mmz Name[MAX_MMZ_NAME_LEN];
        }ast Comm Pool[VB_MAX_COMM_POOLS];
```

在本系统中我们定义了一个视频缓存池来针对视频信息进行 H.265 的编码，并设置最大缓存池的个数为 128。根据媒体软件处理平台的底层数据存储格式设置 20 个视频缓存块并设置每块的大小为 601344。然后调用 MPI 接口 HI_MPI_VB_SetConf 和 HI_MPI_VB_Init 对视 MPP 频缓存池进行设置和初始化。进行初始化后再通过调用 MPI 接口 HI_MPI_SYS_Set Conf 和 HI_MPI_SYS_Init 对系统控制参数进行设置以及对 MPP 系统进行初始化。若不能够对其进行初始化则进行去初始化并放回到视频缓存池中。具体过程如图 5.1 所示。

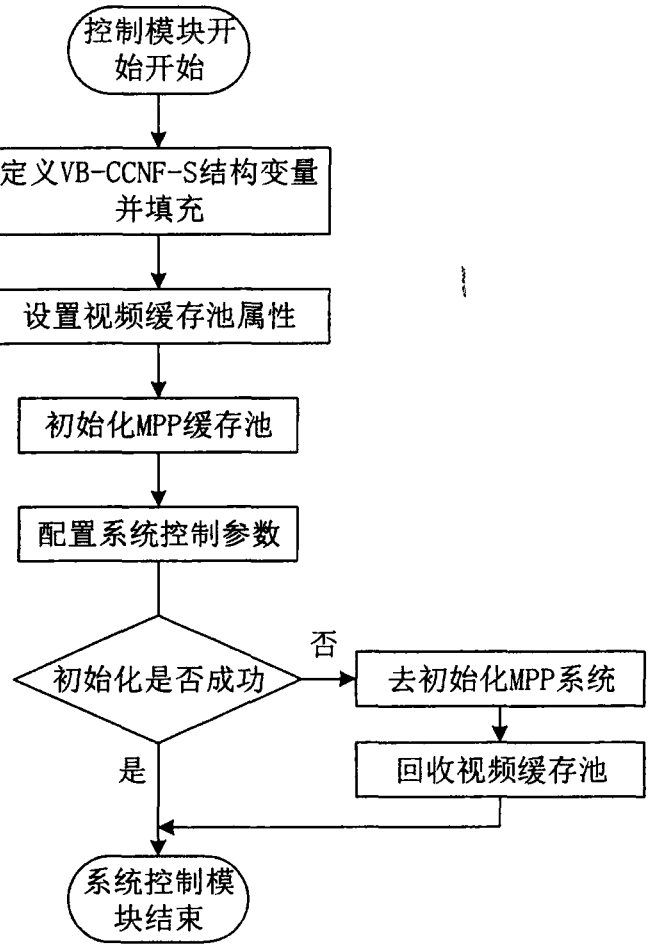

图 5.1 系统控制流程图

5.1.2 视频图像的采集

视频输入模块（VI）主要包含一个视频输入设备和一个视频物理通道来实现它的功能。虽然只有一个视频物理通道，但是他可以进行视频通道的扩展，最多能够扩展到16个通道来实现对来至物理通道数据的缩放活覆盖功能。而视频输入设备和视频物理通道所实现的主要功能为对视频时序进行解析和将解析后的视频进行裁剪等操作后传输至DRR。具体的处理过程如图5.2所示

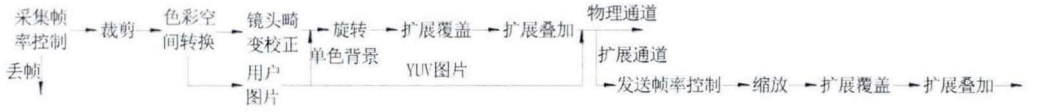

图5.2 输入模块处理过程

本系统通过摄像头OV3640采集视频数据来进行处理。如图5.3所示，通过调用函fd=open("/dev/hi_mipi",O_RDWR);ioctl(fd,HI_MIPI_SET_DEV_ATTR,&MIPI_BT1120_ATTR)来打开MIPI，并将模式设置成BT1120。然后通过对结构体VI_DEV_ATTR_S进行填充设置对图像捕获的第一信息做标为（0,20）扫描模式为逐行扫描以及捕获的图像分辨率。在通过使用接口HI_MPI_VI_Set Dev Attr和HI_MPI_VI_Enable Dev进行属性的设置和视频输入设备的启动后，使用结构体VI_CHN_ATTR_S设置与之前对应相同的参数但是起始坐标设置为(0,0)。最后通过使用HI_MPI_VI_Set Chn Attr和HI_MPI_VI_Enable Chn对视频物理通道进行属性设置和启动。

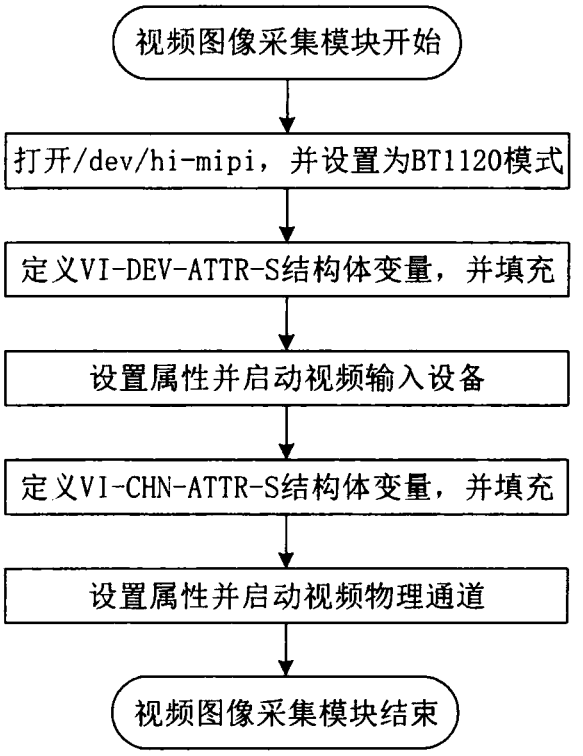

图5.3 视频采集的实现

5.1.3 视频图像的处理

视频图像处理模块 VPSS 可以对一张图片信息进行去噪和去隔行的统一预处理，过后再实现对每个通道的图像信息进行缩放和锐化等功能。视频处理模块可以根据不同的方案设置不同的组（GRUOP）和通道(CHANNEL)。每个组有多个通道组成，并且分时复用视频图像处理模块的硬件。视频图像处理的具体流程如图 5.4 所示：

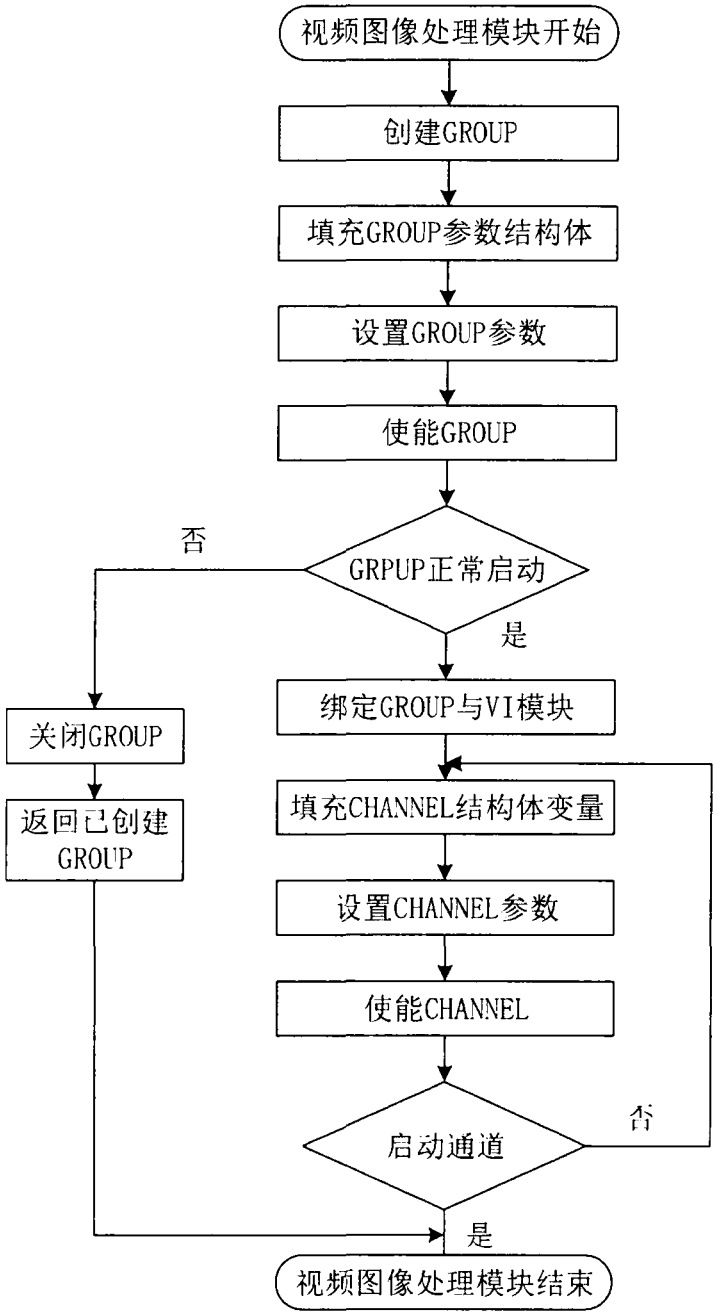

图 5.4 视频处理的实现

在系统中，VPSS 处于连接视频输入与输出的位置。如图 5.5 所示，通过对 SYS 的绑定接口，对输入的或者已编码的视频进行相应的处理后传输给输出端，并且还可以通过 MPI 接口对组进行管理。视频处理模块接收到来至摄像头采集到的 H.265@30fps 的视频信息，对此设置一个组和一个通道。然后对通过对结构体 VPSS_GRP_ATTR_S 进行和之前视频采集模块对应相同的设置然后对其设置参数后启动，如果启动失败则关闭创建的组并返回。若成功则调用结构体 VPSS_CHN_MODE_S 进行通道属性的设置进行 H.265 压缩。在本节所用的主要结构体如表 5.1 所示。

图 5.5 处理模块的连接作用

表 5.1 视频处理所用结构体

创建组	设置组参数	开始组	关闭组	返回组	设置通道属性	设置通道模式	开始通道
HI_MPI_VPSS_Create Grp	HI_MPI_VPSS_Set Grp Param	HI_MPI_VPSS_Start Grp	HI_MPI_VPSS_Stop Grp	HI_MPI_VPSS_Destroy Grp	HI_MPI_VPSS_Set Chn Attr	/HI_MPI_VPSS_Set Chn Mode	HI_MPI_VPSS_Enable Chn

5.1.4 视频图像的编码

系统中视频图像的编码是通过视频编码模块 VENC 来实现的，图 5.6 显示其主要组成部分为编码通道子模块及编码协议子模块。它可以同时进行不同协议、不同模式的多路实时编码。在编码时，可以对 YUV 格式的图像进行编码，但是针对本系统中使用的 H.265 压缩编码只支持 Semi-planar YUV 4:2:0。在此模块通道接收到所需编码的图像信息后根据图像的大小和编码通道大小来进行对图像的编码。

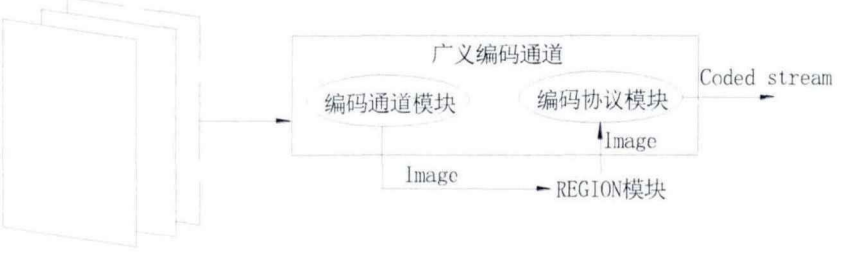

图 5.6 视频编码模块

本系统中，首先设置视频编码模块具有一路编码通道，并且接收由视频处理模块发送过来的视频数据，以此得到 H. 265 码流。第一步，对结构体 VENC_CHN_ATTR_S 进行开发，主要代码如下：

```
typedef struct hi VENC_CHN_ATTR_S{
VENC_ATTR_S st Ve Attr; //编码器属性
VENC_RC_ATTR_S st Rc Attr; //码率控制属性
}VENC_CHN_ATTR_S;
```

针对本设计，将设置的主要参数为 H.265 编码、帧方式得到码流、30fps 的帧率。然后通过使用结构体 HI_MPI_VENC_Create Chn 和 HI_MPI_VENC_Start Recv Pic 进行对 H.265 编码通道的建立和开启。其次，使用结构体 HI_MPI_SYS_Bind 进行对编码通道和视频处理模块的绑定。最后一步是对编码通道是否完成的判断，若未成功则使用结构体 SAMPLE_COMM_VENC_UnBind Vpss/SAMPLE_COMM_VENC_Stop 对编码通道进行解绑与关闭。最后，完成对视频信号进行 H.265 的编码通道,具体的流程如图 5.7 所示。

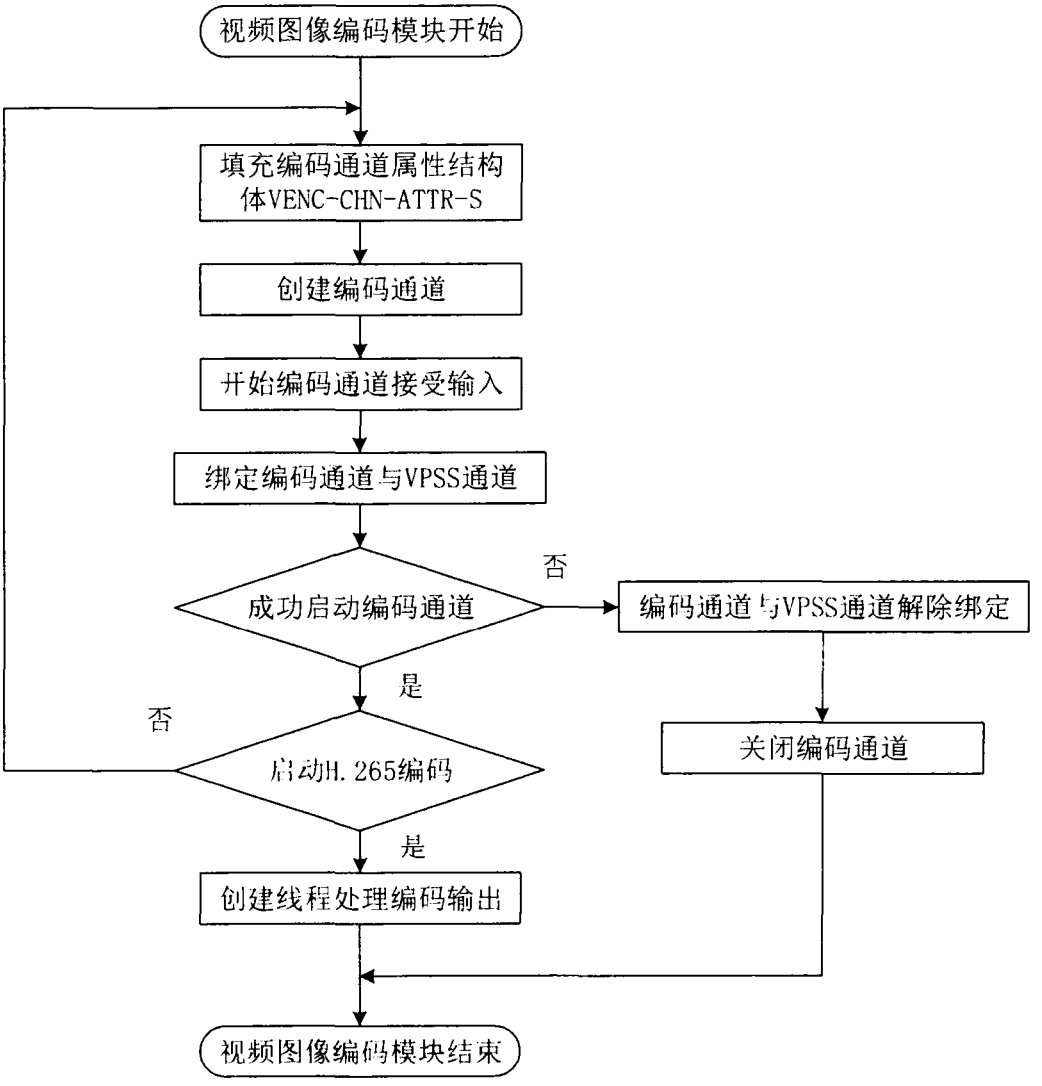

图 5. 7 视频信息 H. 265 编码的实现

由上面可知，视频编码模块是通过线程的建立得到 H.265 码流并推送至流媒体服务器。首先需要使用 open 函数来对视频文件进行打开，然后将得到的编码通道设备描述符加入与之对应的映射集中。使用 select 函数进行监测视频通道是否有新数据进入，若有则根据写入数据设置通道、数据包的数量以及空间地址。最后，使用结构体 HI_MPI_VENC_Get Stream 把视频缓存快地址给相应的成员指针变量，并将数据写入到视频文件中。具体的代码实现代码如下：

```
fwrite(pst Stream->pst Pack[i].pu8Addr+pst Stream->pst Pack[i].u32Offset,\
pst Stream->pst Pack[i].u32Len-\
pst Stream->pst Pack[i].u32Offset, 1, fp H265File);
fflush(fp H265File);
while(One Packet Send Flag){
        sem_wait(semw);
memcpy(shm_add,pst Stream->pst Pack[i].pu8Addr+\
pst Stream->pst Pack[i].u32Offset+Data Have Send,(Data Send Last Time=\
pst Stream->pst Pack[i].u32Len-pst Stream->pst Pack[i].u32Offset-\
Data Have Send>BUF_SIZE?BUF_SIZE:\
pst Stream->pst Pack[i].u32Lpst Stream->pst Pack[i].u32Offset-\
Data Have Send));
        Data Have Send += Data Send Last Time;
        if(Data Have Send==pst Stream->pst Pack[i].u32Len-\
pst Stream->pst Pack[i].u32Offset){One Packet Send Flag = 0;}
        sem_post(semr);
```

5.1.5 MPP 的实现

本节主要进行对多媒体开发软件进行整体实现自动加载 MPP 驱动程序。由于 H.265 编码、视频输入设置都在媒体处理平台的 ko 文件中。所以首先复制所需的媒体处理平台的 ko 文件夹到 Hi3516A 的/etc/init.d/目录下进行驱动加载及对应的脚本文件。然后对 HI356A 中增加命令./load3516a -a -sensor bt1120 -osmem 来进行多媒体处理平台驱动的加载。如图 5.8 所示，MPP 加载实现成功。

```
Hisilicon Media Memory Zone Manager
Module himedia: init ok
load sys.ko for Hi3516A...OK!
Load tde.ko ...OK!
load region.ko ....OK!
load vgs.ko for Hi3516A...OK!
ISP Mod init!
load viu.ko for Hi3516A...OK!
load vpss.ko ....OK!
load vou.ko ....OK!
Load hifb.ko OK!
load rc.ko for Hi3516A...OK!
load venc.ko for Hi3516A...OK!
load chnl.ko for Hi3516A...OK!
load h264e.ko for Hi3516A...OK!
load h265e.ko for Hi3516A...OK!
load jpege.ko for Hi3516A...OK!
load vda.ko ....OK!
ive mod init success!
```

图 5.8 MPP 的实现

5.2 视频图像的网络传输

5.2.1 传输协议和会话协议的应用开发

视频传输模块在视频监控系统中是尤其重要的一部分，通过视频传输部分连接客户端和服务器，并且使客户端和服务器可以进行双向交流。这里一定用到 OSI 模型中的传输层协议，也就是我们所说的传输协议。

传输层模块在本系统中主要的目的是建立服务器和客户端之间可以进行数据的传输。传输层协议具有两种不同方式连接的服务协议，一种是面向连接并且具有可靠性的传输控制协议 TCP（transmission control protocol），另一种是无面向连接可靠性低但是具有相对传输效率的用户数据协议 UDP（user datagram protocol）[35]。

对于我们熟悉的 TCP 而言，通过“三次握手”使目标双方建立连接，并且只要在双方进入连接后才能进行真正的数据传输。这样虽然能具有极高的可靠性，而且因为传输两端都有缓冲校验机制能够对所传输的数据包进行顺序的接收和解码[35]。可是为这付出的代价就是不利于系统进行实时性。因为在传输过程中数据包出现问题，则接收处会要求重新发包，这样的话不易实现实时性，进一步就会影响客户端对视频信息的解码。另一方面，TCP 报头文所占的内存是 UDP 报头文所占内存的三倍之多，并且还缺少客户端解码需要的时间戳服务信息。UDP 虽然不可靠，有丢包的几率，但是丢包率也不会很高。正是由于不需要对丢包进行检测重发而提高数据传输的效率，相对适合应用于视频监控系统。但是为了检测数据包在传输过程中所出现类似于丢包或乱序的问题，在本系统中选用应用在 UDP 协议之上的 RTP 作为本系统是视频传输协议。RTP 的传输过程如图 5.9 所示。

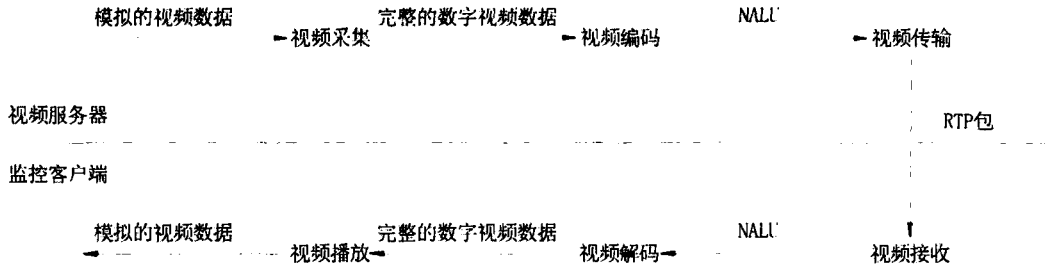

图 5.9 RTP 传输流程

RTP 标准是 1996 年由 IETF 多媒体传输小组发布的介于应用层与传输层的一个网络传输协议。它由两个子协议组成，一个是封装与发送所需传输的数据的 RTP 协议，另一个是发送在传输数据包过程中含有发送的和丢失的字节数等重要信息的 RTCP 协议[36]。在本系统中将 RTP 应用在 UDP 之上的过程如下所述：

（1）服务器端将 RTP 的执行程序装进应用程序（创建 RTP 信息包）中。

（2）应用程序发送含有 RTP 的信息包至 UDP 的套接借接口。

（3）客户端通过 UDP 套接接口将 RTP 信息包送至应用程序。

（4）通过 RTP 执行程序的写入，提取出媒体数据的应用程序。

在 RTP 的使用中，只完成了传输层功能的部分，所以大大降低了对传输层的处理，从而增加了灵活性。而且在其使用过程中，数据有与控制流不是利用的同一端口，而是两个相邻的端口，这样可以分开工作，互不干扰。另一个方面由于它提供了一个协议框架，我们可以根据自己的系统来进行针对的开发，加上可以作为一个应用程序来为系统提供服务，所以具有较好的扩展性和实时性。RTP 应用于 UDP 协议层次图如图 5.10 所示。

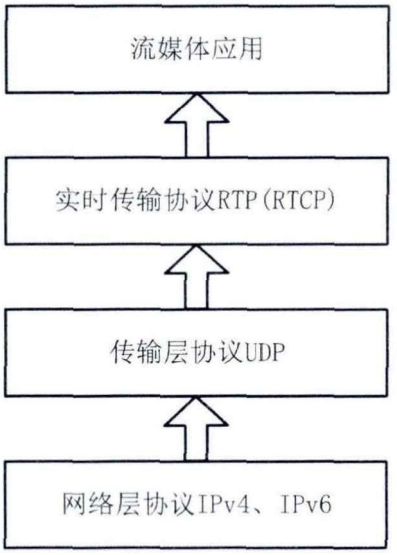

图 5.10 RTP 应用 UDP

数据在传输的同时避免不了对数据的控制和管理，这就需要会话控制协议。常见的会话控制协议有 HTTP 协议和 RTSP 协议。在 HTTP 协议的使用中，一方面由于具有较少的定义方法，所以在较复杂的实际应用中不太好实现。而且不能与另外的承载协议并存，在使用中灵活性差。另一方面 HTTP 的应用不是双向的，只能由客户端来发送请求，所以其应用的范围比较窄[37]。我们在本系统采用的是 RTSP 作为本系统的会话控制协议。

RTSP 协议被用于建立的控制媒体流的传输，它为多媒体服务扮演“网络远程控制”的角色。尽管有时可以把 RTSP 控制信息和媒体数据流交织在一起传送，但一般情况 RTSP 本身并不用于转送媒体流数据[38]。媒体数据的传送可通过 RTP/RTCP 等协议来完成。

根据服务器的资源和在本系统实时性的要求下，本系统完成了所需要的 RTSP 信息交互时的三种状态。第一种为初始化状态（RTSP_STATUS_INIT），第二种为 RTSP 服务器和客户端成功链接的状态（RTSP_STATUS_READY），第三种为播放状态（RTSP_STATUS_PLAYING）。

本系统中所建立 RTSP 的信息交流的具体过程图 5.11 所示。

图 5.11 RTSP 信息交流

在以上的图中所示，第一个状态的建立就是对系统进行初始化。第二种状态的建立是客户端首先发送一个 SETUB 的请求，其中所完成是任务是开放一个 RTSP 的通道，并且传达能够双方使用的数据传输协议集和使用的端口号范围以及播放模式的信息。服务端接收到 SETUB 的请求后回应相应的信息要求，并建立好应有的连接，也就是 SETUB 的响应过程。当这套请求响应完成后进入到第二种状态。

第三种状态的建立是客户端发送 PLAY 请求消息，该消息的主要作用是要求服务端与之建立通道并发送流信息。服务端对其响应不断的发送数据到客户端完成要求进入第三种状态。当是客户端发送 TEARDOWN 信息要求对所建立连接进行关闭时，则当前会话关闭进入到初始化状态，具体状态变化图如图 5.12 所示。

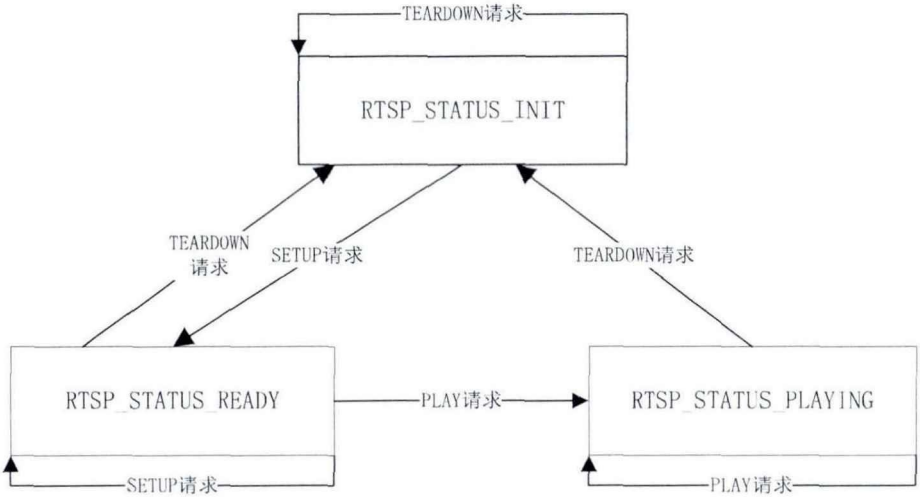

图 5.12 RTSP 状态变化图

测试两台计算机是否建立连接，本设计中利用抓包工具在局域网内进行抓包。证明电脑间可以进行数据交流。状态如图 5.13 所示。

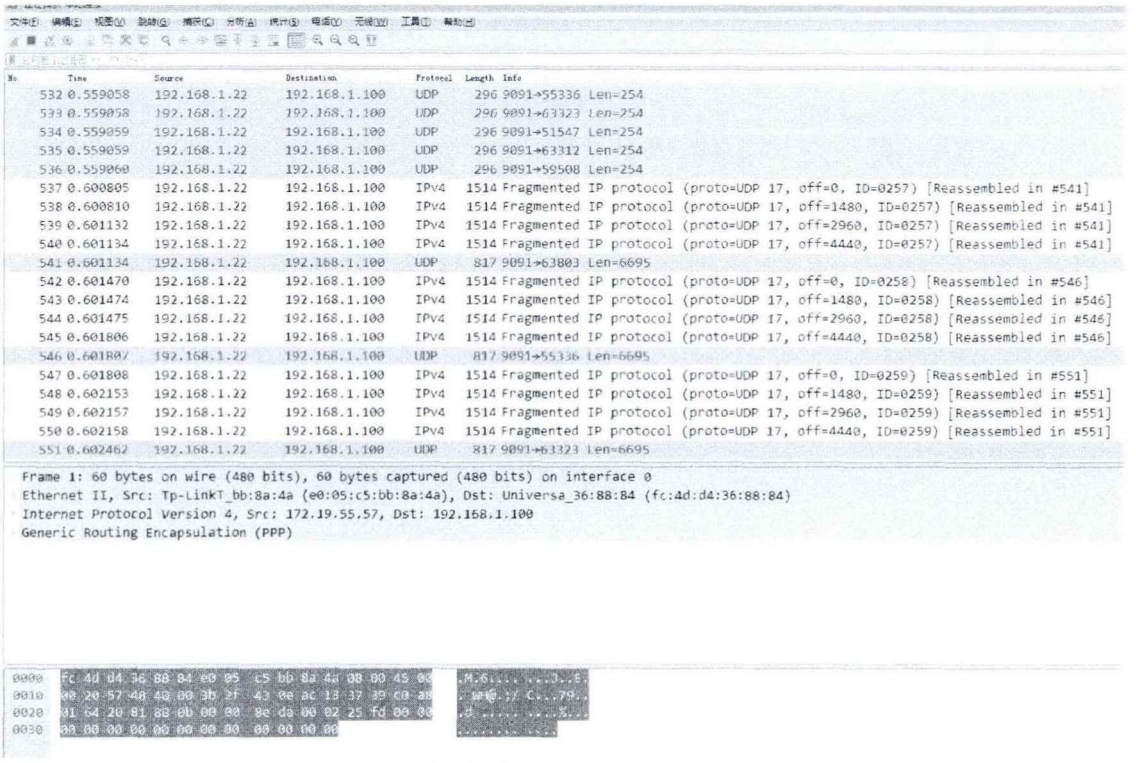

图 5.13 局域网内视频传输的实现

5.2.2 基于 Hi3516A 系统的 4G 模块应用

本节主要介绍在本系统中实现无线传输的 4G 模块进行驱动的移植、开发拨号上网程序以及在 4G 模块上 USB 转串口进行数据传输的实现。最终实现系统在 4G 无线信号下进行视频的传输。

在服务器所使用的 linux 系统上，控制外围硬件的方法是对硬件驱动程序的使用。USB 驱动由两部分组成，分别为主机控制器驱动程序和 USB 设备驱动程序。从逻辑结构上看，USB 设备分为四个级别，依次为设备、配置、接口和端点[39]。USB 驱动具体组成如下图 5.14 所示。

图 5.14 USB 的组成

在本系统所使用的4G模块中，其驱动程序由两部分组成。一部分为USB设备网卡驱动，它由USB驱动与网卡驱动所构成。它的主要任务是将所需要传输的数据进行封装以及发送，并对接收到的数据传送到网络层，在传输之前对其进行应有的处理。另一部分是USB转串口驱动，此部分完成的任务是将高速传输的USB接口变成相比速度较低的串行接口[40]。Linux系统加载域格模块驱动USB具体的操作过程如下：

① 通过以下主要命令进行添加Linux串口驱动组件配置内核。选中USB driver for GSM and CDMA modems组件并进行添加。

```
cd kernel
make menuconfig
device drivers -> usb support -> usb serial converter support
```

保存设置并对通过命令 hisilicon$make ARCH=arm CROSS_COMPILE=arm-hisiv300-linux- uImage 其进行编译后得到如图5.15。

```
usbcore: registered new interface driver usbserial
usbserial: USB Serial Driver core
usbcore: registered new interface driver option
USB Serial support registered for GSM modem (1-port)
```

图5.15 加载4G模块过程

② 通过以下代码添加ppp支持。保存配置并通过命令 hisilicon$make ARCH=arm CROSS_COMPILE=arm -hisiv300-linux- uImage 进行编译。

```
Device Drivers -> Network device support  -> <*> PPP (point-to-point protocol)//在其中选中所有ppp选项：
Device Drivers ->
Network Device Support->
<*> PPP (point-to-point protocol) support
[*]  PPP multilink support
<*> PPP support for async serial ports
<*> PPP support for sync tty ports
<*> SLIP (serial line) support
```

③ 使用命令ls usb查看usb设备，然后查看已安装的设备。

④ 对内核目录drivers/usb/serial下的option.c进行修改，添加域格的USB驱动设备。

⑤ 通过命令make ARCH=arm CROSS_COMPILE=arm-hisiv300-linux- modules编译模块驱动，在drivers/usb/serial目录下生成option.ko和usb_wwan.ko，然后将这两个文件复制到Hi3516A平台上加载，如图5.16所示。

图 5.16 4G 模块驱动的加载

⑥ 通过命令 insmod usb_wwan.ko 和 insmod option.ko 对驱动程序进行安装后在目录/dev 下生成 ttyUSB0 到 ttyUSB4 这 5 个设备如图 5.17 所示。

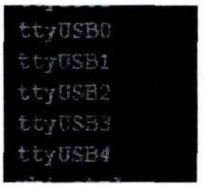

图 5.17 4G 模块驱动安装成功

由上图可知完成了对 4G 驱动的移植安装。然后在 Linux 下进行 AT 指令的交互。首先将 SIM 卡插入到 Hi3516A 的 SIM 卡槽中，并将 4G 全频天线连接到模块的射频连接器。对并将模块进行开机操作再次加载 USB 驱动并获取上图五个端口。然后运行命令 #minicom -s 并在其菜单下，配置“Serial device”为/dev/ttyUSB2 修改完毕后退出到 minicom 菜单，选择“Save setup as dfl”保存配置后选择“exit”后退出 minicom 配置。然后通过 minicom 发送 AT 指令对系统进行测试。

对于 4G 模块，我们需要对其进行拨号连接。主要是对 ppp 进行移植并编译实现首先通过命令# tar zxvf ppp-2.4.5.tar.gz 对 ppp 进行解压然后通过命令# cd ppp-2.4.5 和 # ./configure 对其进行配置最后通过命令# make CC=arm-hisiv300-linux-gcc 对其进行编译。编译完成后，进入 pppd chat 目录，将生成的 pppd 和 chat 可执行文件拷贝到目标板文件系统的，然后将 pppd 目录下的已经编译好的 pppd 文件拷贝至开发板短的文件系统的/usr/sbin 目录下，然后通过# . ./yuge.lte-pppd& 加载拨号脚本并测试网络。

5.2.3 传输协议端口的设置

在本系统中使用了一种逻辑意义上的端口来进行实现将软硬件连接起来实现数据的传输。而具体协议的端口是通过端口号来进行标记的。各软件在应用之前与设置的或者我说预设的端口进行连接就可以各自得到自己所需要的数据。

在本系统中 RTSP 端口号的设置在 ARTSPConnection.cpp 文件中进行。首先从 url 中获取端口号，然后获取不到就为其通用的端口号 554，所以在本系统中 RTSP 使用的端口号为 554。对 RTSP 端口进行设置的具体实现过程如下：

```
ARTSPConnection::ParseURL(
    const char *colonPos = strchr(host->c_str(), ':');
    if (colonPos != NULL) {
        unsigned long x;
        if (!ParseSingleUnsignedLong(colonPos + 1, &x) || x >= 65536)
{               return false;
        }
        *port = x;
        size_t colonOffset = colonPos - host->c_str();
        size_t trailing = host->size() - colonOffset;
        host->erase(colonOffset, trailing);
    } else {
        *port = 554;
    }
```

在本系统中 RTP 和 RTCP 端口号的设置在 ARTPConnection.cpp 文件中进行。首先进行函数声明，创建一对相邻端口的 UDP 数据报套接字。RTP 套接字是偶数端口，RTCP 套接字端口比 RTP 端口号大一个号。然后进行函数的定义内容完成 RTP 和 RTCP 端口的具体设置。

5.3 客户端图像的解码与显示

5.3.1 多媒体视觉处理工具 FFmpeg

在视频监控系统的客户端中，需要对视频数据进行记录、转换、编码等操作。FFmpeg 恰好可以完美的实现这些功能。并且其能够支持的操作系统很广泛，包括我们常用的 Windows 系统、Mac 系统以及 Linux 系统等。FFmpeg 完全支持视频监控系统中所需要的功能，它具有对视频数据的采集、视频数据格式的转换、对所需视频画面的截图以及给视频图像加上水印等[41]。

FFmpeg 主要由各种功能模块组成，并且在 H.265 发布以后，FFmpeg 即使更新了针对 H.265 的解码[43]。对本系统中使用的 FFmpeg 的主要模块解析如下：

libavformat：在 FFmpeg 中主要功能是对所获取的视频进行生成和解析其封装格式，并对下一步的视频编码进行码流的提供。

libavcodec：对接收到的视频流进行 H.265 的解码（FFmpeg 已经完美支持对 H.265 的解码功能）。

libavdevice：令硬件采集加速显示等。

ffmpen：对接收到的视频格式进行转换以及抓取电视卡画面进行格式转换。

ffsever：在本系统中支持 RTSP 实时协议的广播串流服务器。

ffplay：通过 SDL 来显示客户端所接收到的数据，通俗的讲就是一个播放器。

5.3.2 实现 FFmpeg 对 H.265 的解码

客户端接收到的视频为经过 H.265 压缩的视频，若想在客户端显示处理首先要对其进行解码。我们常用的解码器 FFmpeg 包含的编解码库不仅支持前些年流行于市场的 H.264、MPEG-4 的编码器[44]。而且已经支持针对于 H.265 的解码器。对 H.265 解码过程如图 5.18 所示，宏观来看就是对 H.265 编码的反过程，但是还是存在一丝不同。首先是对 NAL 单元进行分析，经过分析后的比特流进行熵编码，H.265 中的熵编码是采用的 CABAC，主要完成的任务是将接收到的已压缩的视频流解码成最初的语法元素。这些元素中包含有运动矢量、量化的变换系数以及已编码帧的序号等，这些的组合为一些量化系数。然后对这些数据实施重排序后进行反量化和反变换。这一步的主要完成的任务为具有相比之前的编码标准更好准确度的整形离散余弦变换进行频空域的转换，生成残差数据[42]。将生成的残差块和参考图像经过运动估计等处理的预测块实行相加得到基本图像。再将刚刚得到的图像进行 H.265 更加优化的去块滤波等减少失真手段的方法得到最终的 H.265 解码图像。

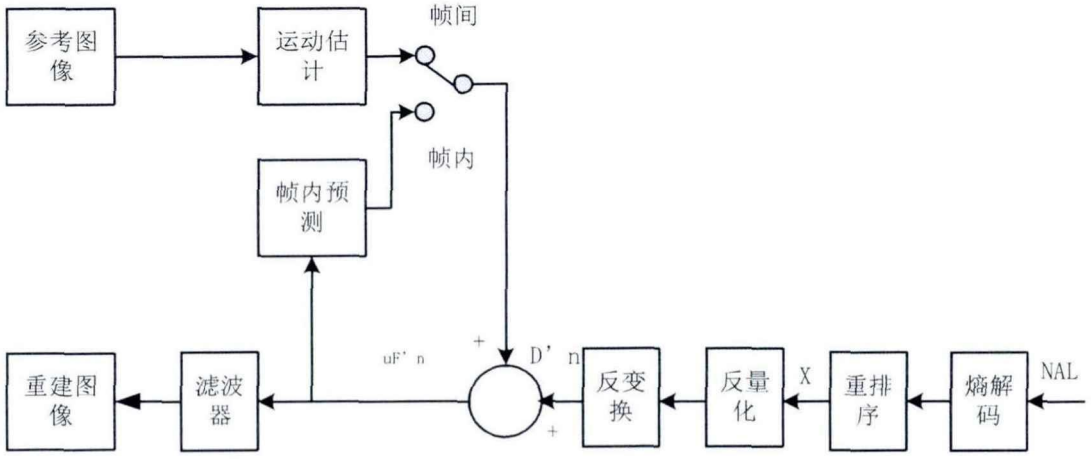

图 5.18 H.265 解码图像流程图

在利用 FFmpeg 内所包含的解码器对经过 H.265 标准压缩的视频流进行解码。其中在 FFmpeg 和 H.265 解码器之间有两个结构体作为接口，分别为 ff_hevc_parser 和 ff_hevc_decoder。在这两个结构体中包含的相关函数如下表 5.2。

表 5.2 解码结构体相关函数

parser_init()	parser_parse()	parser_close()	hevc_init()	hevc_parse()	hevc_close()
初始化解析器	解析	关闭解析器	初始化 HEVC 解析器	解析 HEVC 码流	关闭 HEVC 解析器
init()	decode()	close()	hevc_decode_init(	hevc_decode_frame()	hevc_decode_free()
初始化解码器	解码	关闭解码器	初始化 HEVC 解码器	解码 HEVC 码流	关闭 HEVC 解码器

在 parse_nal_units()和 decode_nal_units()中对解析函数调用完成 H.265 码流中的一些类似于 SPA、PPS 以及 Slice Herder 信息进行解析。

ff_hevc_decode_nal_vps()：解析 VPS。
ff_hevc_decode_nal_sps()：解析 SPS。
ff_hevc_decode_nal_pps()：解析 PPS。
ff_hevc_decode_nal_sei()：解析 SEI。

H.265 的编码中一大特点就是用编码树单元代替了之前的宏块编码，相应的对其进行不同的解码方式和解码函数,其中对编码树单元进行解码主要的处理函数为 hls_decode_entry()。解码函数（Decode）通过帧内预测、帧间预测等方法解码 CTU 压缩数据。CTU 解码模块对应的函数是 hls_coding_quadtree()。hls_coding_quadtree()用于解析 H.265 码流的四叉树结构的句法，是一个递归调用的函数。当解析到单个 CU 的时候，会调用 CU 的解码函数 hls_coding_unit()。hls_coding_unit()会调用 hls_prediction_unit()和 hls_transform_tree()分别对 CU 中的 PU 和 TU 进行处理。hls_prediction_unit()会调用 luma_mc_uni()或者调用 luma_mc_bi()进行预测。hls_transform_tree()用于解析 TU 的四叉树结构的句法，是一个递归调用的函数。当解析到单个 TU 的时候，会调用 hls_transform_unit()对 TU 进行处理。通过以上函数的应用，具体的在 FFmpeg 对 H.265 进行解码的流程图如图 5.19 所示。

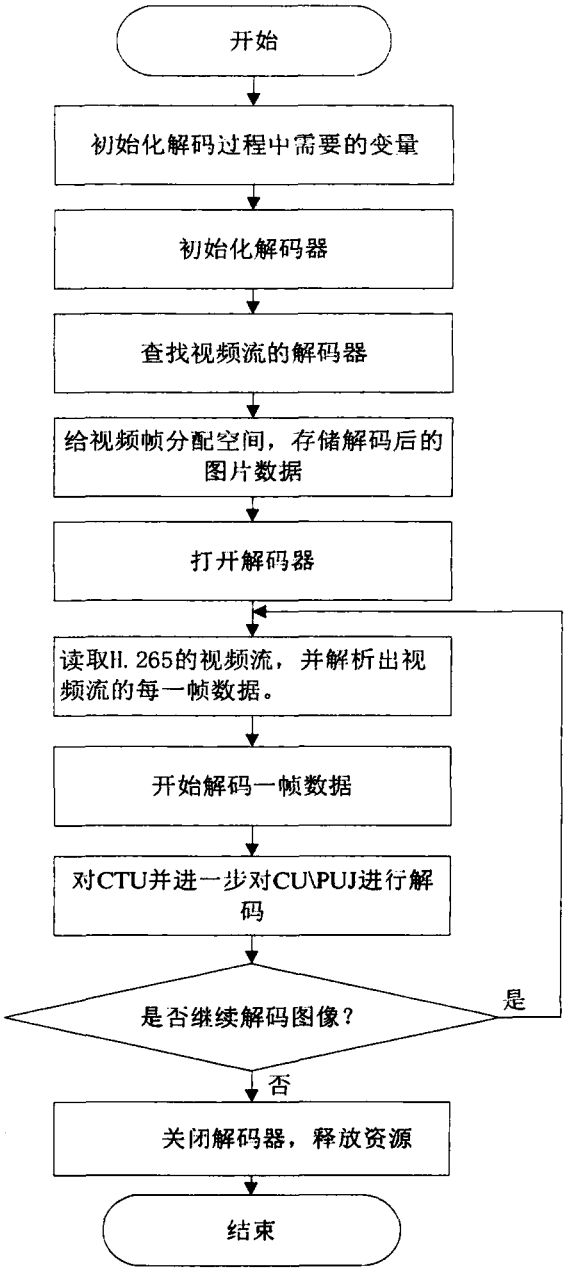

图 5.19 H.265 解码图像的实现

5.3.3 通过 DirectDraw 实现对视频的显示

DirectX 开发包是可以在 Windos 平台上对多媒体进行开发的应用。本监控系统中客户端必定要开发视频显示、视频回放以及视频采集等多种实现。DirectDraw 是 DirectX 开发包中的一部分，能够兼容本监控系统客户端所在的 Windows 系统的应用和驱动程序[45]。Directdraw 具有对应的软件接口可以访问显示设备。而且兼容 Windows 图形接口设备（GDI）的特性使 DirectDraw 能够实现本系统中对视频显示的要求。DirectDraw 主要由以下五种要素构成[46]:

协作级：这一要素可对显示模式进行选择，并能够用 DirectDraw 对应用程序进行可

调性控制。

显示模式：这一要素在 DirectDraw 中提供有可选模式为调色板式和非调色模式，在本监控系统的显示中选择的是后者。

DirectDraw 对象：这一要素为整个显示模块的核心，它是系统设立的首个对象，其他对象都是通过其衍生而成。但是对象与对象间是不存在依赖的关系。一个对象对应一个应用进程。

表面：本要素在 DirectDraw 中比较重要的一个要素，它关系着图像的显示。表面分为主表面和副表面，其中主表面又分为可见的和不可见两种。可见的就是我们在当时所看到的图像，而不可见的是在可见的后方的缓冲区内。当图像翻转时，可见的去掉，而不可见的变成可见的显示出来。这样可以加快视频的显示，并且有利于对其开发。

裁剪器：可以对某一指定区域进行 Bit 操作。通过是指向某一区域并且可进行单独编写的特性对监控客户端进行多窗口化。

在通过 DirectDraw 显示图像的具体过程基本就是通过函数的调用来实现硬件驱动，在所需显示的地方进行图像的显示，具体流程如下图 5.20 所示。

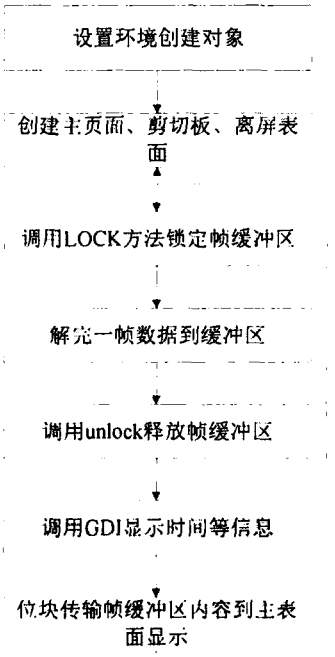

图 5.20 图像显示的实现

本系统中对 DirectDraw 实现视频数据显示的具体过程为：第一步：在使用的 VC 上安装 DX 系统的软件开发包，并对库文件进行相应的连接，具体的操作为将软件开发包内的 lib 和 include 添加到到所用 VC 的对应库文件和头文件下。并将静态连接库 Ddraw.lib 加入到工程的库文件列表。第二步：创建 DirectDraw 对象具体操作如下

```
LPDIRECTDRAW    m_lp DD;       // Direct Draw
Direct Draw Create(NULL，  &m_lpDD，  NULL);
```

其中将函数的首个参数设为 NULL 作为显示设备的唯一标识符（GUID）。为了图像可以显示正常我们把创建的对象的工作模式设置成窗口模式。第三步：创建表面，分为创建主表面和离屏表面，其中主表面是通过 Create Clipper 的使用来设置剪切板对具体区域的尺寸等进行设置，使其能刚好适合所建立的窗口。也就是我们当时所看见的屏幕。而离屏表面的建立是使用函数 IDirectDreaw::CreateSurface 来设置长宽等尺寸与其分辨率相同，存储格式为 YUV4:2:0。第四步：建立 SetClipper 来对参数进行设置。第五步：由于主表面和缓冲画面对其的填充，再加上 GDI 对其时间等信息的打印。在表面上完成绘图。

5.4 监控视频的目标检测

在视频监控系统中，我们要对客户端接收到的图像进行目标检测。比如说仓库的管理、夜间校园的监控以及对重要实验设备的监管等都需要对运动目标进行检测。在监控领域，运动目标的检测一直是活跃的话题之一。特别是对动态图像进行目标检测无论从智能安防、军事目标跟踪还是对人体运动的分析都有很大的作用。

本系统目标检测的功能是通过对 OpenCV 开发来实现的。OpenCV 是由 Inter 建立的开源计算机视觉库。这届计算机库函数基本都是有 C 语言和 C++语言而写，并且提供了像 MATLAB、Ruby、Python 等多种语言的接口。它支持的操作系统非常广泛，不仅可以在 Linux 系统上使，也可以在 Windows 系统上使用[47]。其具有的具体特点如下：

（1）具有开源的特性，方便开发者对其进行开发。

（2）一定的语法规则方使开发者开发式进行交流。

（3）API 接口相对较全，能够提供图像处理以及计算机视觉等技术的共同算法。

（4）支持平台广泛，方便研究开发。

目标检测是最近研究的热点，不仅对安防有重要推动作用，还对图像处理的研究有着很大的重要性。下面对常见的三种运动目标检测方式进行比较。

帧间差分法。该方法是根据相邻帧的图像的不同进行处理，然后根据；两张图片的差来进行运动目标的检测。但是由于较慢运动时，差别不明显。所以检测出效果不完整的情况。

背景差分法：这种是先设置好一张图像作为背景图像，然后根据检测的图像与背景图像进行对比，对比出就是运动目标的检测。但是只适合固定的场景，对于经常有运动的场景不好进行目标检测。

光流法：该方法是通过物体进行运动产生的光流场来进行检测。这种方法虽然准确，但是开发难度较高，而且具有复杂的计算[48]。

所以根据以上三种方式的对比，我们选择第一种方法进行改进来实现本系统的目标检测功能。在本系统中对帧间查分法做的改进的进行三帧差分法，该方法是选取连续的三张图片来进行帧间处理，然后对所得到的信息进行二值化处理以及后处理[49]。接着需要完成对所有像素点在上一步得到的信息进行逻辑运算。得到一样的信息后取出变化的轮廓信息。具体对监控视频进行目标检测过程如下：

首先选择视频信息中连续的三帧图像$I_{i-1}(x,y), I_i(x,y), I_{i+1}(x,y)$,对每相邻的两帧进行差值计算

$$\mathrm{d}_{(\mathrm{i},\ \mathrm{i}-1)}(x,y) = I_i(x,y) - I_{i-1}(x,y)$$

$$\mathrm{d}_{(\mathrm{i}+1,\ \mathrm{i})}(x,y) = I_{i+1}(x,y) - I_i(x,y)$$

对得到的差值图像通过阈值 T 进行二值化

$$\mathrm{d}_{(\mathrm{i},\ \mathrm{i}-1)}(x,y) = \begin{cases} 1 & \mathrm{d}_{(\mathrm{i},\ \mathrm{i}-1)}(x,y) \geq \mathrm{T} \\ 0 & \mathrm{d}_{(\mathrm{i},\ \mathrm{i}-1)}(x,y) < T \end{cases}$$

$$\mathrm{d}_{(\mathrm{i}+1,\ \mathrm{i})}(x,y) = \begin{cases} 1 & \mathrm{d}_{(\mathrm{i}+1,\ \mathrm{i})}(x,y) \geq \mathrm{T} \\ 0 & \mathrm{d}_{(\mathrm{i}+1,\ \mathrm{i})}(x,y) < T \end{cases}$$

然后在每个像素点（x，y）与得到的二值化逻辑图像相“与”，得到三帧图像中的中间帧的二值图像。

$$B_I(x,y) = \begin{cases} 1 & \mathrm{b}_{(\mathrm{i},\ \mathrm{i}-1)}(x,y) \bigcap \mathrm{b}_{(\mathrm{i}+1,\ \mathrm{i})}(x,y) = 1 \\ 0 & \mathrm{b}_{(\mathrm{i},\ \mathrm{i}-1)}(x,y) \bigcap \mathrm{b}_{(\mathrm{i}+1,\ \mathrm{i})}(x,y) \neq 1 \end{cases}$$

对本系统视频图像进行三帧差分法进行目标检测的具体框图如图 5.21 所示。

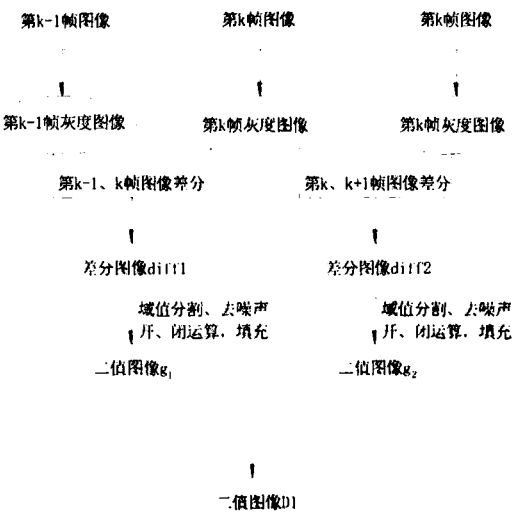

图 5.21 三帧差分法的实现

5.5 客户端软件 MFC 的多线程设计

本系统的客户端是通过 MFC 编程进行设计的。MFC 是我们在项目开发中所用到的基础函数库，是由微软的 VC 开发集成环境所提供的。并且在 MFC 中包含对多线程设计的支持。而线程是进程内部的一个执行单元，每个进程中最少有一个线程，并且这个线程为主执行线程[50]。它是有系统创建，在进程执行是就随之执行。在具体的就需要环境下可以在这个进程下建立多个线程，这些线程都在这个进程的虚拟地址空间中，并且对系统资源、地址空间等都是公用的，所以也带来了同一进程的线程间可以灵活通讯的优点。除此之外，多线程具有并行处理的优点，可以在处理器（CPU）运行时进行减压。

在 MFC 中有工作线程和用户接口线程两种。其中工作线程基本是一个函数作为一个线程，在线程的创建和启动后进会进入到运行状态，处于运行状态时若需要使用共享资源则会进行对资源的同步处理，但是并没有消息循环机制。而用户接口线程可以对用户进行响应，具有消息循环的机制[51]。

在本系统的 MFC 设计中，通过对系统函数 AfxBeginThread()进行调用来实现对线程的建立，所有的线程都是由 CWinThread 类表现。然后对 CreateThread()函数进行调用来进行线程的启动以及初始化，并返回该函数的指针。进而对线程的进行、暂停以及优先级进行设置。在多线程的设计中，优先级是至关重要的。多线程中优先级的设置影响着客户端在运行时的整个调度过程。本系统中是通过 CWinThread 类提供的 GetThreadPriority()函数来进行对多线程优先级的设定。通过对函数 SetThreadPriority()的使用进行对线程优先级记性修改。在工作者线程中，如果控制函数执行至 return 和 exit 等语句以及函数 AfxEndThread()时，则停止工作者线程。本系统主要进行了对图像接收、图像显示以及图像格式转换三个线程的建立，具体设计的流程图如图 5.22 所示。

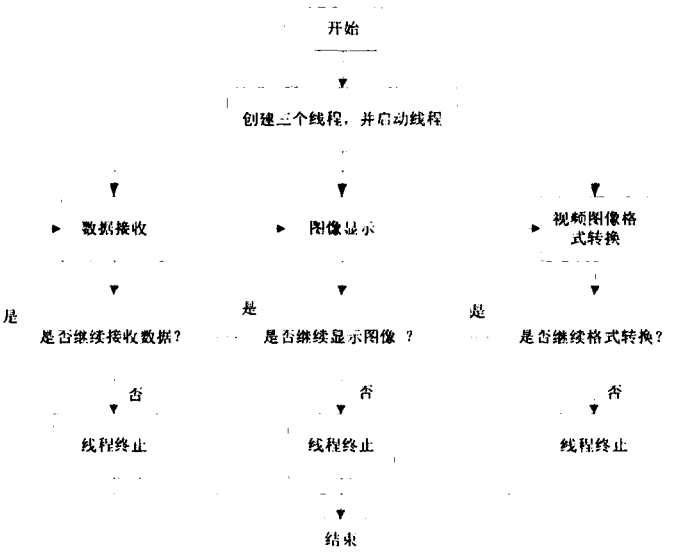

图 5.22 多线程的实现

5.6 本章小结

本章主要针对系统的软件进行开发，完成了对 Hi3516A 的媒体处理软件的开发。包括视频的采集、处理和编码。然后对媒体处理软件进行实现并给出结果。还完成了对视频图像传输的开发，实现了图像可以通过4G信号进行传输到客户端。客户端通过FFmpeg对视频进行 H.265 的解码开发得到几乎不失真原图像。然后通过对 DirectDraw 的开发使图像显示在屏幕上，并使用三帧差分法对图像进行目标检测以及在客户端进行 MFC 多线程的设计。

第 6 章 系统的测试与实现

6.1 测试环境

本系统基于 H.265 的无线视频监控系统所用的服务器是以 Hi3516A 开发板进行开发的。4G 模块使用的是上海域格 CLM920_CN3 进行 4G 无线视频传输。摄像头使用的是基于 CMOS 的 OV3640。客户端使用的是 ThinkVision 牌子的电脑。系统测试环境如图 6.1 所示。

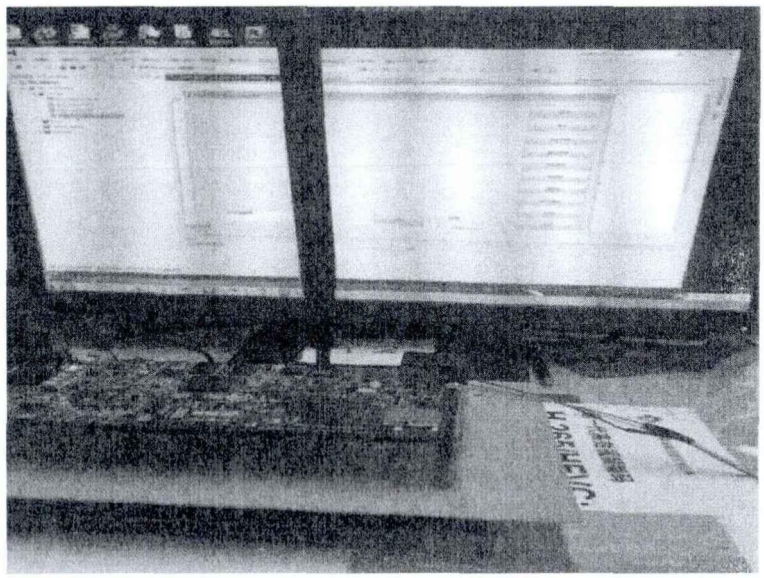

图 6.1 测试环境

本监控系统的客户端是在 VS2012 下进行开发的主要实现对监控视频的播放、监控视频的录像、监控视频的播放录像以及目标检测功能。由图可以看出客户端可对服务器传送过来的视频信号无卡顿播放。客户端主要界面如图 6.2 所示。

图 6.2 客户端界面

6.2 系统测试

6.2.1 对 H.265 硬编码进行测试

监控服务器端处理器 Hi3516A 进行的 H.265 硬编码进行测试，以及与之前通过 H.264 进行编码测试的压缩比进行对比。在实际测试时为了减少传输过程中的数据流量，使用 640x480 分辨率的视频图像进行测试。结果显示通过 H.265 进行编码的视频比通过 H.264 进行编码的视频在字节数上大大的减少，并且大大的提高压缩比。测试结果如表 6.1 所示。

表 6.1 编码测试

第 k 帧图像	图像分辨率	编码前字节数	H.265 编码后字节数	压缩比	H.264 编码后字节数	压缩比
1	640x480	614400	13752	44.76：1	22509	27.29：1
2	640x480	614400	22096	27.80：1	38962	15.76：1
3	640x480	614400	12504	49.13：1	19476	31.53：1
4	640x480	614400	12942	47.47：1	20513	29.95：1
5	640x480	614400	25794	23.81：1	41689	14.73：1
6	640x480	614400	14821	41.45：1	23541	26.09：1
7	640x480	614400	14097	43.58：1	22942	26.78：1
8	640x480	614400	23917	25.69：1	39527	15.54：1
9	640x480	614400	13948	44.05：1	21524	28.54：1
10	640x480	614400	13275	46.28：1	20845	29.46：1

6.2.2 对 4G 信号进行测试

运行 PPP 拨号命令. ./yuge.lte-pppd &测试 4G 是否能够拨号成功。由图 6.3 可知，拨号成功。

```
/mnt # . ./yuge.lte-pppd &
/mnt # Starting pppd
Script chat -s -v "ABORT" "NO CARRIER" "" AT OK AT+CGDCONT=1,\"IP\",\"\" OK ATD*99**1# CON
NECT finished (pid 1407), status = 0x0
Serial connection established.
using channel 1
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB3
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0xe0ebebbf>]
rcvd [LCP ConfReq id=0x0 <asyncmap 0x0> <auth chap MD5> <magic 0x44f86db0> <pcomp> <accomp>
]
sent [LCP ConfRej id=0x0 <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0xe0ebebbf>]
rcvd [LCP ConfReq id=0x1 <asyncmap 0x0> <auth chap MD5> <magic 0x44f86db0>]
sent [LCP ConfAck id=0x1 <asyncmap 0x0> <auth chap MD5> <magic 0x44f86db0>]
rcvd [LCP DiscReq id=0x2 magic=0x44f86db0]
rcvd [CHAP Challenge id=0x1 <4261698e187d70a80c09257ldc2c7536>, name = "UMTS_CHAP_SRVR"]
sent [CHAP Response id=0x1 <34b5ee237c19532a23438e651ab24c02>, name = "card"]
rcvd [CHAP Success id=0x1 ""]
CHAP authentication succeeded
CHAP authentication succeeded
sent [CCP ConfReq id=0x1 <deflate 15> <deflate(old#) 15> <bsd v1 15>]
sent [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0
.0.0>]
rcvd [LCP ProtRej id=0x3 80 fd 01 01 00 0f 1a 04 78 00 18 04 78 00 15 03 2f]
Protocol-Reject for 'Compression Control Protocol' (0x80fd) received
rcvd [IPCP ConfReq id=0x0]
sent [IPCP ConfNak id=0x0 <addr 0.0.0.0>]
rcvd [IPCP ConfRej id=0x1 <compress VJ 0f 01>]
sent [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [IPCP ConfReq id=0x1]
sent [IPCP ConfAck id=0x1]
rcvd [IPCP ConfNak id=0x2 <addr 10.79.155.254> <ms-dns1 221.7.128.69> <ms-dns2 221.7.136.68
>]
sent [IPCP ConfReq id=0x3 <addr 10.79.155.254> <ms-dns1 221.7.128.69> <ms-dns2 221.7.136.68
>]
rcvd [IPCP ConfAck id=0x3 <addr 10.79.155.254> <ms-dns1 221.7.128.69> <ms-dns2 221.7.136.68
>]
Could not determine remote IP address: defaulting to 10.64.64.64
Failed to create /etc/ppp/resolv.conf: No such file or directory
not replacing existing default route via 192.168.1.1
local  IP address 10.79.155.254
remote IP address 10.64.64.64
primary   DNS address 221.7.128.69
secondary DNS address 221.7.136.68
```

图 6.3 4G 拨号的成功

然后在服务器测试能否连接到网络进行 4G 传输，由图 6.4 可知能够通过 4G 信号上网。

```
/mnt # ping 180.97.33.107
PING 180.97.33.107 (180.97.33.107): 56 data bytes
64 bytes from 180.97.33.107: seq=0 ttl=48 time=209.003 ms
64 bytes from 180.97.33.107: seq=1 ttl=48 time=185.485 ms
64 bytes from 180.97.33.107: seq=2 ttl=48 time=175.632 ms
64 bytes from 180.97.33.107: seq=3 ttl=48 time=205.385 ms
64 bytes from 180.97.33.107: seq=4 ttl=48 time=195.379 ms
64 bytes from 180.97.33.107: seq=5 ttl=48 time=186.114 ms
^C
--- 180.97.33.107 ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 175.632/192.833/209.003 ms
/mnt # ping www.baidu.com
PING www.baidu.com (163.177.151.109): 56 data bytes
64 bytes from 163.177.151.109: seq=0 ttl=54 time=43.788 ms
64 bytes from 163.177.151.109: seq=1 ttl=54 time=39.003 ms
64 bytes from 163.177.151.109: seq=2 ttl=54 time=36.852 ms
64 bytes from 163.177.151.109: seq=3 ttl=54 time=39.106 ms
64 bytes from 163.177.151.109: seq=4 ttl=54 time=37.085 ms
64 bytes from 163.177.151.109: seq=5 ttl=54 time=36.871 ms
64 bytes from 163.177.151.109: seq=6 ttl=54 time=37.466 ms
64 bytes from 163.177.151.109: seq=7 ttl=54 time=37.228 ms
^C
--- www.baidu.com ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 36.852/38.424/43.788 ms
```

图 6.4 通过 4G 信号连接网络

6.2.3 对录像功能以及播放录像进行测试

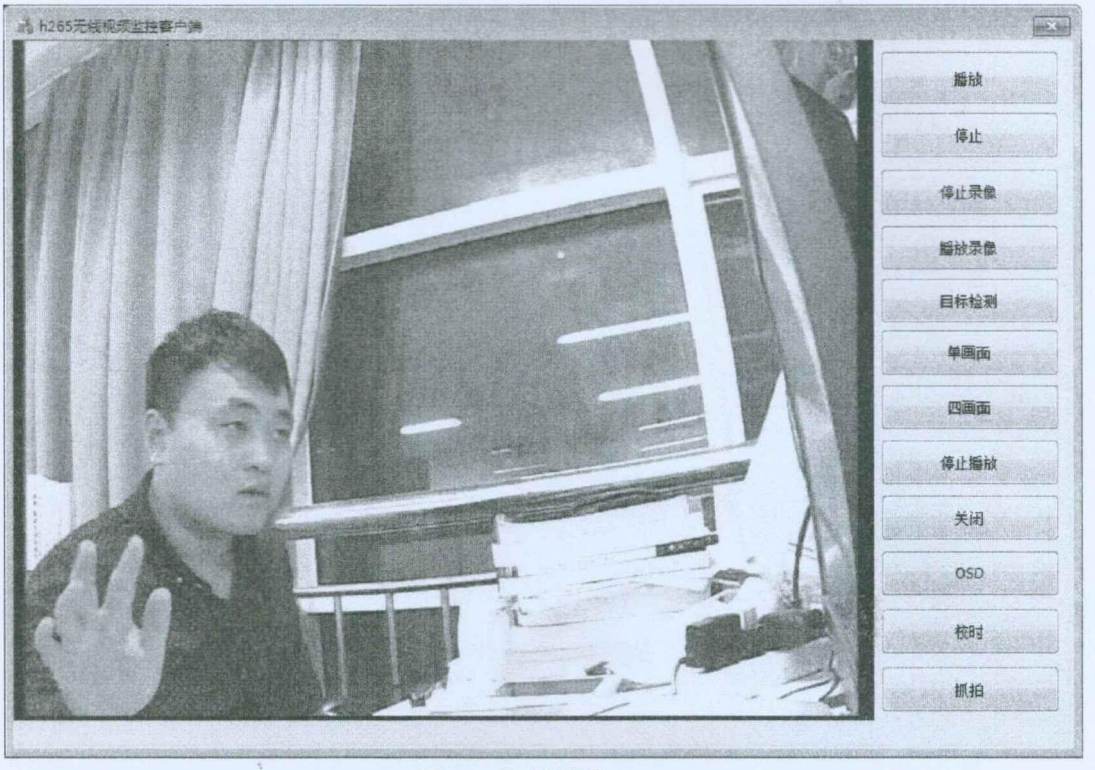

图 6.5 录制视频

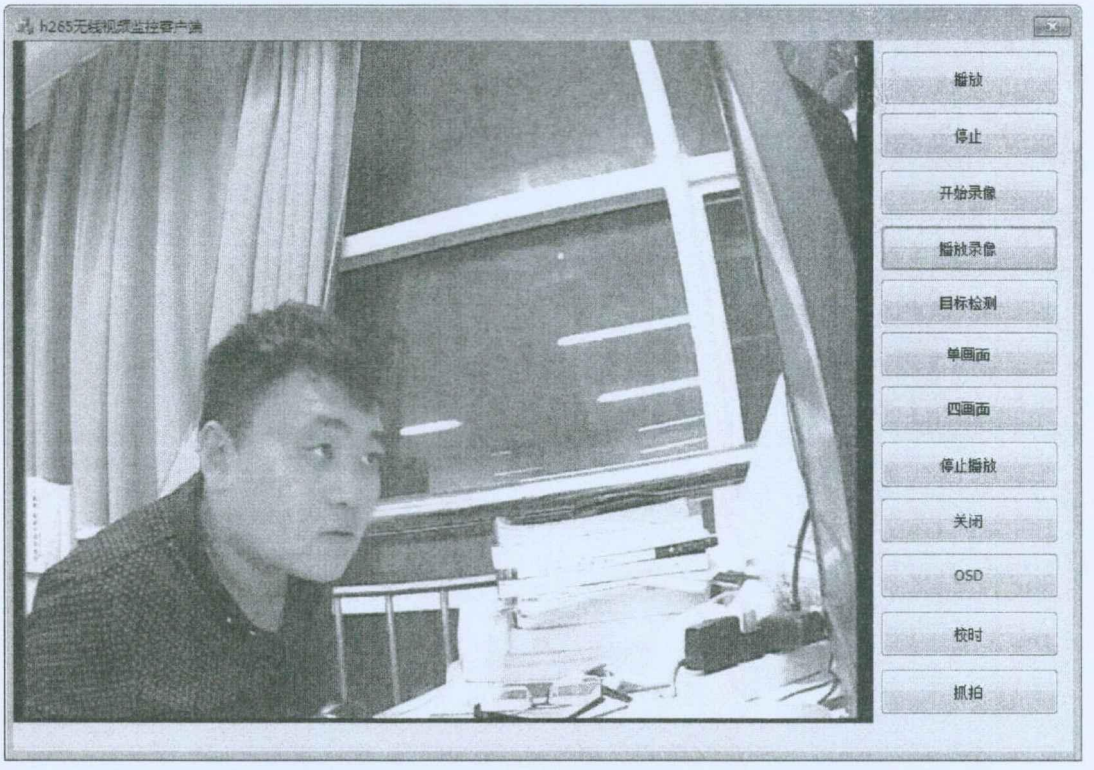

图 6.6 播放录像

6.2.4 对监控视频进行目标检测

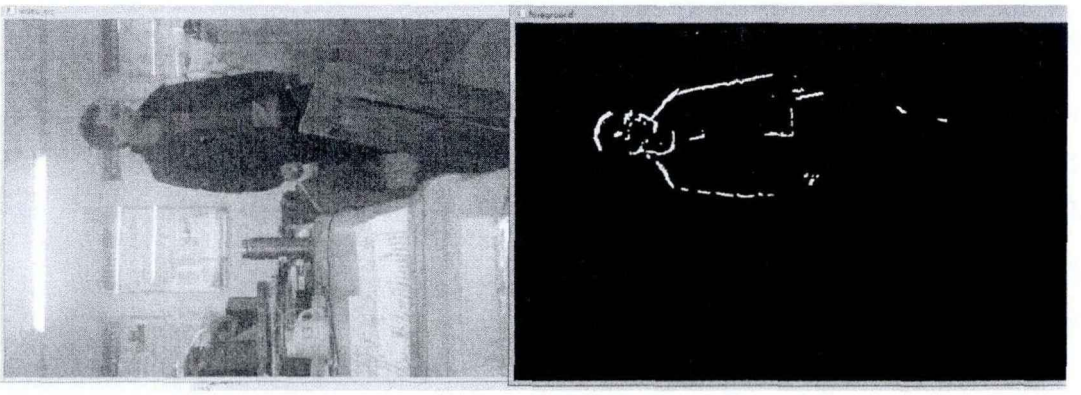

图 6.7 对运动物体进行目标检测

6.3 结果分析

从上一节的测试结果可知，采集的视频信号进行 H.265 编码后的视频通过 4G 模块进行传输，并且在客户端接收图像信号显示。由图 6.2 所示，显示的图像清晰流畅。解决了 H.264 编码造成的卡顿影响。通过对编码标准的比较测试，表 6.1 表明通过 H.265 进行编码大大的降低了传输时的数据量，最大压缩比可以达到 47.47:1。相对 H.264 编码的视频数据可以降低 40%以上的数据量，测试结果表明 H.265 必将是未来几年编码领域的深度研究方向，可以大大降低编码率。由图 6.4 的测试结果表明，图像在通过 4G 传输的丢包率较低。图 6.5 和 6.6 是对客户端进行录像以及录像的播放的测试，通过测试表明该功能基本实现。图 6.7 是对监控图像下有运动物体进行目标检测，并显示出来。通过师弟在监控画面下运动，在目标检测端可以基本显示师弟的轮廓，功能基本完成。同时在客户端可以正常显示说明对视频的解码开发成功。通过对模块逐一测试后，各项指标基本完成。整个系统运行流畅，基本达到设计要求。

第 7 章 总结与展望

7.1 总结

近年来，随着中国社会与经济的飞速发展，安防系统在中国乃至世界显的尤为重要。在数字化和网络化成熟的今天，安防中的视频监控系统是由嵌入式和视频编码两者应用技术的结合体。在高清晰化高智能化的今天，只能通过高压缩的无线传输来满足我们的需求。本系统的实现具有实时性、高压缩性、无线传输等特点。主要完成的任务如下：

（1）对现阶段视频监控系统技术做了大量研究，对比在视频监控系统中压缩算法的研究、无线传输介质的研究、处理器方案的研究等。根据研究对比设计了一套视频监控系统。

（2）根据本论文对监控系统的要求，对系统进行整体设计。对 Hi3516A 芯片进行学习开发，以 Hi3516A 为中心服务器设计外围电路。实现系统的硬件电路。

（3）对 Hi3516A 进行嵌入式开发环境的搭建，包括交叉环境的搭建、内核的移植以及跟文件系统的制作等。

（4）对 Hi3516A 的媒体处理软件的开发，其中主要包括对视频图像的采集、处理、编码以及对 MPP 进行移植实现。视频编码对 H.265 进行开发使采集到的视频通过 H.265 编码。

（5）对本系统中传输模块进行开发，主要包括 RTSP 的开发、对 4G 模块进行开发使能够进行在客户端与服务器间进行 4G 无线通信。

（6）对 FFmpeg 进行开发实现对 H.265 编码视频的解码，并通过对 DirectDraw 进行开发显示出来并实现视频信号的录像、录像回放等功能；对 OpenCV 进行开发实现对监控视频的目标检测。

7.2 展望

本系统实现了一套视频监控系统的整体搭建。并完成了对采集视频通过 H.265 压缩后进行 4G 信号传输到客户端显示出来。同时还完成了在客户端对监控视频进行录像、录像回放以及目标检测的功能。整个监控系统实现了高压缩传输、4G 无线传输、实时传输等性能。但在某些方面还应需要改进，具体如下：

（1）4G 已经比较普遍，5G 时代马上来临，将会有更好的无线传输介质用于无线视频监控系统。

（2）运用差分法实现目标检测还存在不是很准确的情况。并且没有在检测的同时加入报警功能。

（3）现在手机监控 APP 很方面，没有延伸出手机客户端来实现视频监控。

参考文献

[1] 章卓君. 海康威视:受 G20 峰会带动的企业价值评估分析[J]. 企业导报, 2016(18):194-194.

[2] 何海东, 董全武, 陈宏. HEVC & VP9 在网络直播点播中的应用探索(上)[J]. 广播与电视技术, 2014, 41(8):171-172.

[3] 刘娟. 基于高性能视频编码(HEVC)算法的改进[D]. 东华理工大学, 2014.

[4] 蔡晓霞, 崔岩松, 邓中亮,等. 下一代视频编码标准关键技术[J]. 电视技术, 2012, 36(2):80-84.

[5] 张亮. 视频检测技术及在移动监控中的应用[J]. 中国公共安全, 2006(1):110-113.

[6] 简伟, 于海滨, 盛庆华. 基于同轴电缆的多路视频混合传输技术的研究[J]. 电子器件, 2012, 35(4):412-416.

[7] 耿建军. 智能建筑中视频监控系统设计与应用[D]. 山东大学, 2007.

[8] 侯云东. 基于嵌入式系统的视频监控系统的设计与实现[D]. 电子科技大学, 2010.

[9] 王彤. 基于 FFmpeg 的 H.264 解码器实现[D]. 大连理工大学, 2011.

[10] 艾孟奇, 刘钊. 基于 H.264 标准的视频解码优化及 DSP 实现[J]. 计算机应用, 2007, 27(s2):299-301.

[11] 李海涛. 视频编码标准 H.264/AVC 的编码算法研究[D]. 杭州电子科技大学, 2014.

[12] 刘定佳. H.264 视频编码算法研究及 DSP 实现[D]. 西安电子科技大学, 2010.

[13] 钱亚冠. 视频编码理论与 MPEG-4 的 DSP 实现[D]. 浙江大学, 2005.

[14] 赵亚楠. 新一代视频编码标准 H EVC 的并行化研究[D]. 上海交通大学, 2013.

[15] You S D, Chen W H. Comparative study of methods for reducing dimensionality of MPEG-7 audio signature descriptors[J]. Multimedia Tools and Applications, 2015, 74(10):3579-3598.

[16] Duta S, Mitrea M. Capacity evaluation for MPEG-4 AVC watermarking[J]. Proceedings of SPIE - The International Society for Optical Engineering, 2016:70000V-70000V-10.

[17] 赵爽. 基于 H.265 的高清网络视频处理技术的研究与实现[D]. 中国舰船研究院, 2016.

[18] 刘景松. H.265 自适应插值滤波器的研究与 CUDA 优化[D]. 北京邮电大学, 2011.

[19] 刘明健. 基于嵌入式 linux 的视频监控系统设计[D]. 江西理工大学, 2015.

[20] 郭铮言. 多标准视频解码芯片高效运动补偿模块设计[D]. 上海交通大学, 2012.

[21] 张云龙. 3G 和 4G 网络的对比[J]. 电子测试, 2013(10):72-73.

[22] 郎为民, 杨德鹏, 李虎生. LTE-Advanced 标准化进展[J]. 数据通信, 2012(1):11-15.

[23] Lu Y, Han B, Cheng J, et al. Design of H265 real-time stream transmission system based on Hi3516A[J]. Microcomputer & Its Applications, 2015.

[24] 闫光. 基于 H.264 的实时视频监控系统的设计与实现[D]. 北京邮电大学, 2012.

[25] Liu Y P, Zhen G Y, Liu D H. Design of Ethernet Interface Based on MSP430 and DM9000[J]. Automation & Instrumentation, 2010.

[26] 鲁云, 韩宾, 程锦发,等. 基于 Hi3516A 的 H265 码流实时传输系统设计[J]. 微型机与应用, 2015(20):42-44.

[27] 鲁云. 基于 Hi3516A 的高压缩比网络图像编码设备的设计[D]. 西南科技大学, 2016.

[28] 王景存, 高峰. 基于 ARM9 的 Bootloader 的分析及设计[J]. 现代电子技术, 2010, 33(2):44-46.

[29] 吴国伟, 姚琳, 毕成龙. 深入理解 Linux 驱动程序设计[M]. 清华大学出版社, 2015.

[30] 刘少华. 基于 ARM9+LINUX 的无线视频监控系统的设计[D]. 西北师范大学, 2015.

[31] 史巧硕, 范东月, 柴欣,等. 嵌入式 Linux 根文件系统的构建与分析[J]. 计算机测量与控制, 2015, 23(2):656-659.

[32] 李飞, 武金虎, 石颖博. 基于 busybox 的根文件系统制作[J]. 电脑知识与技术, 2010, 06(17):4655-4656.

[33] 李乃翠. 基于 Android 与 Wi-Fi 的实时视频监控系统的研究[D]. 山东大学, 2014.

[34] Arcuri A, Fraser G, Galeotti J P. Generating TCP/UDP network data for automated unit test generation[C]// Joint Meeting. 2015:155-165.

[35] Comer D E. Internetworking with TCP/IP.[M]. 人民邮电出版社, 2002.

[36] Montagud M. Design, development and evaluation of an adaptive and standardized RTP/RTCP-based IDMS solution[C]// ACM International Conference on Multimedia. ACM, 2015:1071-1074.

[37] Khan S Q, Gaglianello R, Luna M. Experiences with blending HTTP, RTSP, and IMS [IP Multimedia Systems (IMS) Infrastructure and Services][J]. Communications Magazine IEEE, 2007, 45(3):122 - 128.

[38] Santos-González I, Rivero-García A, González-Barroso T, et al. Real-Time Streaming: A Comparative Study Between RTSP and WebRTC[M]// Ubiquitous Computing and Ambient Intelligence. 2016.

[39] 朱恩亮, 赵腊才, 茹伟,等. Linux 环境下 USB 设备驱动程序设计[J]. 电子科技, 2016, 29(1):108-110.

[40] 邹龙, 王德志, 刘忠诚,等. Linux 系统下 4G 终端模块驱动的实现[J]. 电脑知识与技术:学术交流, 2015, 11(27):206-209.

[41] Liu J, Liu Y F. Design and implementation of the AVS video player based on FFmpeg[J]. Journal of Zhengzhou University of Light Industry, 2015.

[42] 彭旭康. 基于 ARMv8 架构的 H.265/HEVC 视频解码优化[D]. 华中科技大学, 2015.

[43] Online H. Videokodierung: HEVC/H.265-Encoder hält Einzug bei FFmpeg und LibAV[J]. 2014.

[44] Newmarch J. FFmpeg/Libav[M]// Linux Sound Programming. Apress, 2017.

[45] 王彦朝. DirectDraw 技术及其应用[J]. 信息系统工程, 2010(5):84-84.

[46] 张华琳, 高升, 马茜,等. DirectDraw 技术在快视显示系统中的应用[J]. 无线电工程, 2010, 40(8):56-58.

[47] Bradski G R, Kaehler A. Learning OpenCV - computer vision with the OpenCV library: software that sees[M]. DBLP, 2008.

[48] 贾天宇, 谢滨, 闫磊. 基于 OpenCV 的运动目标检测与跟踪[J]. 科技风, 2015(21):14-14.

[49] 丁磊, 宫宁生. 基于改进的三帧差分法运动目标检测[J]. 电视技术, 2013, 37(1):151-153.

[50] 李永忠. 用 Visual C++的多线程技术设计应用程序界面[J]. 网络新媒体技术, 2002, 23(3):187-190.

[51] 段利国. 网络编程实用教程[M]. 人民邮电出版社, 2016.

攻读硕士学位期间的科研成果

[1] 参与一项桂林市科学研究与技术开发课题“基于 4G 车载视频监控和语音集群调度系统研究”，项目编号：20150103-4。

[2] 钟俊文，宋树祥，一种无线视频监控系统[P]. 中国实用新型专利，（申请号：201621097313.4）。

[3] 宋树祥，钟俊文. 一种无线视频监控服务端、客户端、系统和处理方法[P]. 中国发明专利，实质审查中（申请号：201610870051.9）。

致　谢

时光飞逝，在广西师范大学近三年的研究生学习生涯马上就要结束了，我在这两年多的时间里，无论在学习、工作还是生活上都有不少收获。同时我也结识了很多良师益友，从他们那里，我也增长了我的见识。在此，我向在近三年来关心、支持和帮助过我的人表示由衷的感谢。

首先我真诚的感谢我的导师宋树祥老师。在选题和研究过程中，我多次得到宋老师的悉心指导。宋老师还多次询问我的课题研究过程，并为我指点迷津，帮助我开拓思路。宋老师渊博的专业知识，严谨的治学态度，精益求精的工作作风，诲人不倦的高尚师德，严以律己、宽以待人的崇高风范，朴实无华、平易近人的人格魅力对我影响深远。不仅是我树立了远大的学术目标、掌握了基本的研究方法，还使我明白了许多待人接物为人处世的道理。正是宋老师那无微不至的关怀，给我的研究工作创造了优越的条件，让我顺利的完成了论文的撰写和修改。在此我特意向导师表达我诚挚的谢意和崇高的敬意！

我还要感谢我的同门、师弟、师妹，他们为我的研究工作提出了宝贵的意见，对我的毕业产生了很大的影响；感谢我的室友给我创造了一个非常好的学习环境；感谢三年中每一个陪伴在我身边的同学，感谢他们为我提出的各种有益的建议。正是有了他们的支持、鼓励和帮助，我才能充实的度过了三年的学习生活。

我还要感谢我的父母，他们总是在我背后默默的支持我，不管是经济上还是精神上，都给了我支持和鼓励，令我安心的去学习。顺利的完成三年的研究生学业。

我还要感谢培养教育我的广西师范大学，感谢教授我的各位老师，舒适的学习环境和各位老师传授给我的深厚的专业基础知识，必将是我人生中的宝贵财富，让我终生难忘。

最后，由衷的感谢能在百忙之中来审阅本论文的各位老师与专家们。由于本人的知识与能力水平有限，文中避免不了会存在不妥的地方，还敬请各位老师与专家予以指正和批评！

论文独创性声明

本人郑重声明：所提交的学位论文是本人在导师的指导下进行的研究工作及取得的成果。除文中已经注明引用的内容外，本论文不含其他个人或其他机构已经发表或撰写过的研究成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。

研究生签名：钟俊文 日期：2017.6.16

论文使用授权声明

本人完全了解广西师范大学有关保留、使用学位论文的规定。本人授权广西师范大学拥有学位论文的部分使用权，即：学校有权保留本人所送交学位论文的复印件和电子文档，可以采用影印、缩印或其他复制手段保存论文；学校有权向国家有关部门或机构送交学位论文的复印件和电子版。本人电子文档的内容和纸质论文的内容相一致。除在保密期内的保密论文外，允许论文被查阅和借阅，可以公布（包括刊登）论文的全部或部分内容。论文的公布（包括刊登）授权广西师范大学学位办办理。

本学位论文属于：

□保密，在 年解密后适用授权。

☑不保密。

论文作者签名：钟俊文 日期：2017.6.16

指导教师签名：[illegible] 日期：2017.6.15

作者联系电话：_____ 电子邮箱：_____