

网络化摄像机监控软件的设计与实现

摘 要

视频监控系统是现代工业生产和生活中必不可少的部分，它可以广泛应用于银行、邮电、电力、水电、教育、交通、公安、监狱法庭、大型公共设施、大型仓库及军事基地等场所，其性能的优劣直接关系到工业生产的好坏和人民生命财产的安全，而摄像机是视频监控中的用于图像采集得主体部分。

传统的摄像机大多是通过 USB 端口或并口与个人电脑相连进行监控，这种方式的地域局限性、操作复杂性等弱点限制了其应用的可能性，因此我们利用计算机网络和 EI 技术，把体积很小、成本很低的嵌入式设备应用到现有的视频监控系统中，来达到降低成本、简化使用方法的的目的。因为嵌入式设备和普通 PC 机不同所以我们需要设计开发一套专门的监控软件来更好的达到视频监控得目的。

在论文中，首先对网络化摄像机进行了介绍，简单讲述了这种网络化摄像机的工作环境以及它的基本特性。然后在上述条件的基础上重点介绍了本论文欲实现的监控软件的设计过程，通过对整个系统欲实现的功能的分析，采用模块化方法对视频监控系统进行建模，划分出不同功能的子模块，然后详细叙述了各个模块的功能、模块之间的关系和协调工作的过程，并结合模块得程序流程图对每实现的方法进行了具体描述。

最后对实现的监控从功能和性能上进行测试和评估，其中包括其功能与实用性、图像清晰度和系统的稳定性。

关键词 Webit MJPEG 格式 EI 监控系统 IpCamera

The Design and Realization of Monitoring Software Based on Network Camera

Abstract

The monitoring system of video is a necessary part in both the modern industry and human life. It can be widely applied in many fields, such as bank, telecommunication, electricity, education, transport, police office, prison, court, public establishments, big depots, military base and so on, its performance is closely related to the industry manufacturing and the safety of people's possessions, the camera is the main part for collecting images in the monitoring system of video.

The monitoring procedure of the conventional camera is usually obtained by connecting the personal computer with either USB port or parallel port, however, the application of the system is limited by the regional limitation, the operating complexity and so on. In order to decrease the cost and simplify the operation, we embedded the device with small cheap and low cost into the existing monitoring system of video by using Internet and EI technology, thus we can reduce its cost and condense its method. Because of the differences between the embedded device and the common PC, we have to design special monitoring software to achieve the goal of better video monitoring.

In this thesis, we firstly briefly introduce the characterizations and working environment of the network camera. The thesis is concentrated on the design procedure of which the monitoring software is achieved, after analyzing the functions of entirely system, we regard the monitoring system of video as a model and divide it into several sub modules with different function by using the method of modularization. We describe the function of each sub module, its relation and the cooperation procedure between them in details. What more, we also integrate the flow chart into each module to describe the method of realization.

At last, we test and evaluate the functions and the performances of monitoring system, including the functions, practicability, image definition and the stability of the system.

Key words Webit, MJPEG Format, EI, Monitoring System, IpCamera

声 明

本人声明所呈交的学位论文是在导师的指导下完成的。论文中所取得的研究成果除加以标注和致谢的地方外,不包含其他人已经发表或撰写过的研究成果,也不包括本人为获得其他学位而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

本人签名:董 校

日 期: 2004.1.2

第一章 引言

1.1 课题提出的背景

随着中国加入 WTO, 企业管理的要求越来越高, 管理者非常希望远程、可视化的管理自己的单位, 这样在出差、度假或者出国访问时, 也能够远程监控看到自己的单位或企业的生产、工作情况。因此, 视频监控系统是现代工业生产和生活中必不可少的部分, 它集数据、语音、视频等功能于一身, 可以广泛应用于银行、邮电、电力、水电、教育、交通、公安、监狱法庭、大型公共设施、大型仓库及军事基地等场所。其性能的优劣直接关系到工业生产的好坏和人民生命财产的安全^[1], 然而随着时间的发展很多视频监控技术都已经过时, 存在着许多缺陷, 如传输距离近、布线复杂、操作繁琐、难以实现多中心控制、系统容量小、扩展困难和不能实现区域联网等缺点。少数视频监控系统虽然具有网络功能, 但其监控水平较低、成本昂贵, 联动连接方式上依然只是线缆的组合, 无法真正意义的实现计算机的智能化管理, 这些传统的视频监控系统显然不适合现代监控的需要。

Internet 已经成为社会重要的基础信息设施之一, 是信息流通的重要渠道。Internet 网络和无线骨干网技术的飞速发展, 使得基于分组交换技术的通信性能、质量和可靠性得以稳步提高, 网络应用的可靠性应是可以信赖的。过去大量的浩如烟海的 8/16 位单片机的嵌入式设备, 如仪器仪表、数据采集和显示、过程控制、工业自动化和家庭自动化等的实时应用, 已经到了享受网络方便的时候了。如果嵌入式系统能够连接到 Internet 上面, 则可以方便、低廉地将信息传送到几乎世界上的任何一个地方, 同时在任何设备中嵌入 WWW 服务器, 就可以借助网络来访问行为 (Action over IP)^[2]。有人预计, 将来每个家庭平均会拥有 50-100 个大规模集成电路微处理器, 它们嵌入在各种设备器件之中, 如手机、机顶盒、网络终端、个人数字助手、各种家电、汽车、安全防盗系统和烟火报警系统等。在工业控制、航天航空和军事的应用更是举不胜举。

嵌入式 Internet 技术产生的基础是嵌入式技术和 Internet 技术, 它的出现使得众多的工业仪器、设备和家用电器连入 Internet 成为可能。设备上网的概念不只是简单的指设备可以访问 Internet, 可以获取网上资源, 除此之外, 设备也应该能够被用户或网上其它的设备访问到。正是因为嵌入式技术的出现为视频监控的改革提供了前提条件, 再加上近年来计算机网络技术的应用和发展, 宽带网络的普及、视频压缩技术的发展, 为数字化网络视频监控的应用奠定了坚实的基础, 传统的视频监控模式必将被嵌入式数字化网络视频监控模式所替代。

用嵌入式系统作为监控的中心，并且把多媒体技术、网络技术、视频技术应用到监控上，已经完全超越出传统视频监控的范畴，所引入的思想理念带动了高新技术的发展和社会的进步，是一次里程碑似的飞越，很具现实意义。

1.2 课题的提出

目前，我国的现代化建设事业正在不断向前推进，工业和电力等部门为了能长期高效的工作都购买了大量的高档设备，支持着部门中所有的工作，任何一个故障和人为的失误都会给单位乃至更大范围内带来巨大的损失。面对这种情况，为了减少这种损失，国家相关部门已下令机房改造的规定，一些大型的机房已慢慢成为无人职守型。有关人员要实时的监控到设备的运行情况，机房环境的温湿度，电流电压的变化，即便是工作人员的日常维护，也要对其的工作环节进行记录，方便随时的工作检查，杜绝人为的误操作所带来的损失。为了保证上述机房的安全运行要求，该系统必须能够随时观察到机房的运行情况，并能得到相关的录像资料，还有多种设备和环境的各种参数进行遥测、遥信和遥控，实时监测其运行参数，诊断和处理故障，记录和分析相关数据，以及对告警信息的及时反应，在一定范围内联动的警示设备，通知有关人员做出反应，采取补救措施，从而实现机房的少人或无人值守的目的，并对相关设备进行集中监控、集中维护和集中管理，这其中视频监控是一大主要部分。另外，随着人们生活水平的提高，人们的安全意识也在不断的加强，家庭防盗、防火日益受到人们的重视，而这也要借助于视频监控系统，视频监控需求的不断增加与目前我国相对落后的视频监控系统现状构成了尖锐的矛盾。

嵌入式 Internet 技术的发展为提高视频监控系统的监控水平创造了条件。国内视频监控系统的势头虽然是好的，但监控水平仍然不高，提高视频监控系统的监控水平包括许多方面，例如系统能否具有网络功能即能否进行远程监控、能否实时地得到远方被监控环境图像等等，另外，系统硬件体积的大小、安装的方便性、成本的高低及用户界面的友好程度也是值得考虑的问题。

如何适应视频监控的需求，利用 Internet 网络和嵌入式 Internet 技术提高视频监控设备的监控水平是摆在我们面前的一大课题。

东北大学计算机体系结构研究所研究的网络化摄像机（为了和传统的摄像机区别，以后论文中用 IpCamera 来代替）监控系统，就是为了解决诸如此类远程可视化的管理的问题而研究、开发的。它是一种基于嵌入式 Internet 技术的视频监控系统，主要是为解决传统视频监控系统监控距离受限、不具备网络功能、成本高、不利于大范围推广应用等问题，使传统视频监控系统具有新的意义，真正能够广泛地应用到工业生产和家庭生活中去。用户身边只要有 Internet 即可在任何时间、任何地点远程可视化监控现场环境。

IpCamera 是集视频压缩技术、计算机技术、网络技术和嵌入式技术等众多先进的 IT 技术于一体的数字摄像设备。IpCamera 内置了 CCD 晶片、视频压缩卡、网络服务器和网卡等设备，采用嵌入式操作系统，无需计算机的协助便可独立工作。

IpCamera 有它的自己 IP 地址，可直接与以太网连接。它支持很多网络通信协议，如 TCP/IP 协议等，局域网上的用户以及 Internet 上的用户可以根据 IP 地址对 IpCamera 进行访问，观看通过网络传输的实时图像，还可通过对镜头、云台的控制对目标进行全方位的监控。

IpCamera 是专为面向 10/100M 以太网网络构架而设计的数字化视频压缩传输设备，其采用嵌入式 Internet 技术，具有强大的即时图像捕捉和压缩功能，并可将压缩的图像通过如 Internet、Intranet 或普通点对点的线路传送到你需要的任何地方。IpCamera 可广泛应用于安防监控领域，如军队、交通监控、银行安全监控系统、电力和无人值守系统等方面，用以取代传统的模拟视频监控系统，是一个先进、高效的网络视频监控解决方案。

因为嵌入式系统不同于普通的 PC 机，并且视频系统不仅仅是显示现场图像而已，它还包括很多控制功能。所以本文提出了该 IpCamera 的配套软件，使用户能更方便快捷的使用本 IpCamera 达到监控目的。

1.3 论文的组织

本论文的内容共分为六章。第一章是引言，简要介绍了本课题的提出及背景；第二章简要描述了视频监控系统的发展、目前的状况及其发展的趋势，同时介绍了软件设计的基本原则；第三章和第四章是本文的核心，在第三章中详细介绍了监控软件的设计过程，在第四章中详细介绍了软件的具体实现；第五章是对本系统的测试和分析；最后第六章是对全文的总结和对未来视频监控系统发展的展望。

第二章 视频监控系统的开发与软件设计原则

目前视频监控系统已经从本地模拟信号的视频监视、基于 PC 多媒体卡的视频监视这两个传统监控时期发展到了基于嵌入式的数字监控系统。随着计算技术的不断发展,未来的视频监控系统更将会朝着普适计算的方向发展。

2.1 视频监控系统的现实意义

用计算机作为监控的中心,同时把多媒体技术、网络技术、视频技术应用到监控上,已经完全超越出传统监控的范畴,这种思想理念带动了高新技术的发展和社会的进步,是一次里程碑似的飞越,很具现实意义。

1. 可观的经济效益

最先进的防范理论加上高新科技成果,结合功能强大的立体交叉防护网,将可破坏程度降低到零点,为监控现场提供最安全的保证,使人身和财产免受不必要的损失,是一个企业发展和社会进步的根本。由于采用计算机管理控制,只要事先设置好,就可以实现全自动化管理,从根本上实现无人职守,以便更好地分配人力资源,直接或间接带来的经济效益是不可估量的。

2. 强大的威慑力

安全可靠的监控系统通过计算机智能化管理,使报警布局更加趋于合理。计算机将安防工作做得有条不紊,避免了任何疏漏,并充分利用多媒体特性,结合实时捕捉、视频报警、硬盘录像等技术,使防范工作更安全,更周到,更方便,对罪犯起到了强大的威慑力,更能有效地防止犯罪。严密的计算机监控网络,跨越地域时空,实时跟踪每一个监控现场。迅速的反映能力,能更有效地打击犯罪。

3. 减轻保卫强度

保卫工作看似简单,但很繁琐,艰巨,责任重大。利用视频监控管理系统的高度智能化,进一步把保卫人员从枯燥无味的工作中解脱出来,在提高安全性的同时,减轻了工作强度;其独特的网络功能,免去了领导安全检查的辛劳。结合可视化编程技术,良好的操作界面,提高操作员的操作兴趣,更加有利于人机配合监视,提高安全性。

4. 进一步奠定发展的基础

二十一世纪是监控系统全面进入到计算机网络的年代,废弃传统系统,选择以计算机为核心的多媒体监控管理系统,为今后向更高、更远的发展奠定了基础。基于计算机,我们可以不断地将最新、最好的科技引进来,更有利于与时代接轨。

将目前监控仅用于安防的目的，向更高更远的方面发展，融合环境监测、消防安全、自动化控制、网络通讯、智能管理等多方面，使之成为一个整体，相互依托，势必会创造出强大的生产力，促进企业的不断发展。

2.2 传统的视频监控系统

传统的视频监控系统包括闭路电视监控系统和基于 PC 多媒体卡的视频监视的监控系统^[3-4]。

2.2.1 闭路电视监控系统

最早期的产品，多以摄像机与监视器（电视）一对一的监视系统为主，连接方式是靠视频电缆一对一直接相连，有多少个摄像机就有多少个监视器，没有任何技术含量，摄像机等监视设备种类比较单一，功能也有限。视频的传输模式基本采用模拟方式传输，一般采用一条电缆连接到中心控制室的方式。在控制方面大多采用模拟电路与中、小规模数字集成电路制成的控制器，通过手动方式对镜头和云台进行各种控制。

传统的闭路电视监控系统存在许多局限性：有线模拟视频信号的传输对距离十分敏感，当传输距离大于 1000 米时，信号容易产生衰减、畸变、群延时，并且易受干扰，使图像质量下降；其次，有线模拟视频监控只能以点对点的方式监视现场，这使得布线工程量极大；模拟视频信号数据的存储会耗费大量的存储介质（如录像带），而且录像资料的保存和查询工作十分烦琐；由于监控距离严格受限，因此监控管理员必须在专用的监控台前监视现场，对于工业生产关键部分的监控，必须安排多名监控管理员 24 小时轮流值班，一步也不能离开，这就耗费了大量人员的精力，闭路电视监控系统如图 2.1 所示。



图 2.1 闭路电视监控系统

Fig.2.1 The closed circuit television of monitor system

设备成本十分昂贵，无法大范围推广应用，个人家庭更是无力承担；技术含量较少，不具备网络功能，无法利用 Internet 网的巨大便利实现远程监控，用户

只能被动地接收监控报警信息，无法与现场设备进行交互，即无法主动地监控设备，无法根据实际的需要动态地配置设备的各项参数。这种系统的落后性显然与现代工业生产和生活中多媒体监控要求标准的不断增加构成了尖锐的矛盾，随着时代的进步和科技的发展，这种监控系统将逐渐退出历史舞台。

2.2.2 以 PC 机作为监控设备的前置设备的监控系统

在嵌入式 Internet 技术出现以前，有些传统的视频监控设备根据应用的需要，在内部嵌入了微控制器芯片，有些则没有。但是，即使是嵌入了微控制器芯片的设备也没有网络接口。为了扩充这些没有网络接口的视频监控设备的功能，使它们能够获得 IP 并接入到 Internet 上，研究人员常采用一台前置的 PC 机作为 Internet 网络接入服务器，由它来完成 Internet 网络中 TCP/IP 协议和现场监控设备远程监控协议之间的转换，此时这台前置 PC 机的作用相当于设备网关，所有接入到 Internet 的各种监控设备都是通过这台前置 PC 机来统一管理的。

这种基于通用 PC 机的视频监控系统可以为两种类型的设备提供 Internet 接口，一种是直接连接到前置 PC 机 I/O 端口上的设备；另一种是通过专用接口卡与前置 PC 机 I/O 端口连接的设备^[5]。

在第一种情况下，设备直接连接到前置 PC 机的 I/O 端口，如 RS485 接口、RS232 接口、RF 接口、并行口甚至是 USB 接口上，这时前置 PC 机不需要提供设备的专用驱动程序，而只要能够在通信过程中将客户端传送的数据帧中的命令转换成设备可以识别的格式并写入到相应的端口即可；另一方面能够将端口的状态以客户端可以识别的格式传送到相应的客户端即可，其基本结构如图 2.2 所示。



图 2.2 设备直接连接到前置计算机的 I/O 端口

Fig.2.2 Devices are directly connected to I/O ports of PC

但是，由于 PC 机的 I/O 端口资源是很有限的，对于复杂的设备，只利用 PC 机固有的 I/O 端口是远远不够的，而且这种做法是不能进行控制扩展的，有多少台设备上网就需要有多少台 PC 机作 Internet 接入服务器，这样会使成本增加到令人难以接受的程度。

在第二种情况下，通过专用接口卡将多个设备与前置 PC 机连接的方法能在一定程度上解决这一问题。与前一种情况相比，该技术相对复杂一些。与前置 PC

机连接的设备不仅需要设备网关，同时还需要相应的设备驱动程序，这样当用户的命令通过网关转换成设备能够识别的格式后，通过该驱动程序才能够正确的在设备中执行，其基本结构如图 2.3 所示。



图 2.3 设备通过专用接口卡与前置 PC 机连接

Fig.2.3 Devices are connected to I/O ports of PC by a special interface card

利用专用接口卡，可以把 Internet 技术嵌入到多个现场的设备中，只要写出嵌入设备在相应平台下的驱动程序，再根据该驱动程序完成相应的嵌入网关，这样就可以通过 Internet 远程监控现场设备了。

这种以 PC 机作为监控设备的前置设备的监控系统与闭路电视监控系统相比可以说是有了很大的进步，具备了网络功能，因此监控的距离得到了扩大，监控水平有了较大的提高，但在这种方式中，由于有前置 PC 机的存在而使设备成本仍然较高、体积较大，不利于大范围的推广，而且在某些场合，如现场中的干扰比较严重时 PC 机可能无法正常工作，毕竟 PC 机不是为适应视频监控任务而专门设计的，而此时监控系统的可靠性、可用性都将大打折扣。

2.3 嵌入式 Internet 技术及其在视频监控系统中的应用

嵌入式 Internet 技术是一种设备接入技术，借助于该技术，用户可以对接入到 Internet 上的各种非 Internet 标准的设备进行远程监控、管理及维护等操作，从而使各种非 Internet 标准的设备具有上网的能力。

从某种意义上说，嵌入式 Internet 技术放大了人们的感知，它将使得我们的地球变成了一个“地球村”，一个具有电子化皮肤的“地球生物”。

嵌入式系统的研究主要有三个方向：（1）嵌入式应用的研究；（2）嵌入式操作系统的研究；（3）嵌入式软件开发平台的研究。本文中的系统是嵌入式系统在视频监控领域中的一个应用实例，嵌入式 Internet 技术应用于视频监控系统，可以提高传统视频监控系统的监控水平，使其具备网络功能，即用户可以远程可视化监控现场中的各种情况。

基于前置 PC 机的视频监控系统之所以不能广泛的被应用，很大的原因在于它的控制模块（即 PC 机）的体积太大，在一些大型的设备监控上还可以发挥作用，但如果只是简单的监控一些小型的设备，就给人一种大炮打蚊子的感觉，不仅成本不能够接受，而且在有些场合下这种监控方式是根本应用不了的。

由于嵌入式 Internet 技术的应用, 控制模块的体积可以做得很小, 例如象火柴盒或一枚一元硬币那样的大小, 完全可以嵌入到现场的视频监控设备中去, 而且基于嵌入式 Internet 技术的嵌入式控制模块是针对具体应用场合的专用系统, 因此, 可以完全适应某些现场的恶劣环境条件, 不易受干扰的影响, 可以长期稳定的工作。可见, 嵌入式 Internet 技术在促进视频监控系统性能提高方面是功不可没的。

2.4 目前国内外嵌入式视频监控系统的现状

国外一些发达国家由于嵌入式 Internet 技术起步较早, 发展时间较长, 因此基于嵌入式 Internet 技术的视频监控系统监控水平相对较高。

典型的产品有瑞典 AXIS 公司的 AXIS 系列多媒体监控系统。作为网络周边设备发展的领导者, AXIS 公司把在瘦服务器 (Thin Server) 方面的领先技术应用到多媒体监控领域, 向客户提供了简单易用, 稳定可靠的 AXIS2100、AXIS2400 等基于 Web 服务器的网络监控产品。每个 AXIS 多媒体监控产品都是一个高性能的 Web 服务器。该系列多媒体监控产品为在网络上远程控制现场监控设备提供了一条简捷之路。与传统的视频监控不同 (需要专用的监视器和花费昂贵的同轴电缆), AXIS 可将视频监控设备直接连入以太网、Intranet, 甚至是 Internet。被授权的用户通过 Web 浏览器就可以在任何地方监控现场。AXIS 是一种既容易安装又容易使用的产品, 它相对低廉的投资花费使得它在加强用户的视频监控系统中成为一个全能的角色。

国内嵌入式 Internet 技术起步较晚, 发展时间较短, 但发展速度较快。

北京黄金眼科技有限公司研制的黄金眼系列视频监控系统它是集图像数据采集和控制系统于一体的软、硬件平台, 客户可以根据现场实际需求, 利用嵌入式组态软件和硬件板卡快速组建视频监控系统, 满足视频监控要求。它的硬件使用模块化、产品化结构设计, 软件使用嵌入式组态软件, 保障系统的安全性和可靠性, 真正达到以应用为中心, 软硬件可升级和扩展。

本文中的嵌入式视频监控系统是以部队项目为背景而研制开发的, 部队项目要求产品可靠性, 安全性等等, 因而具有一些独自的功能和特点, 这将在后文中作详细的介绍。

2.5 嵌入式视频监控系统优势分析

1. 实时远程监控

该系统毋需专用的监控主机或 PC 机, 直接接入 LAN 或通过 ADSL 接入即可实现实时的本地监控或远程监控; 网络的实时性使得监控的实时性成为可能。便捷的远程访问图像和视频流, 节省时间和专程往返的费用, 基于方便和/或安全的

考虑，图像可以进行自动远程存储。

2. 卓越的性价比

该系统具备传统监控系统的功能，同时可以通过网络远程监控和调控，而价格与传统监控系统相比具有很大的优势。到目前为止，普通网络图像解决方案通常都需要复杂的系统，涉及到 PC，附加软件和硬件，工作站，有时还有视频电缆系统。IpCamera 令人称道的是它的视频监控能力和其独特的高性能的 RISC 芯片，能够在 10/100M 网络上以每秒 30 帧（目前为 15 帧/秒）的速度传送高质量的动态图像。此系统所需设备极其简单，系统的控制全由后端的软件系统实现，省去了传统模拟监控系统中的大量设备，如昂贵的矩阵、画面分割器、切换器等。由于图像的传输通过综合布线网络，省去了大量的视频同轴电缆，降低了费用。

3. 简捷安装和管理方式

该系统配备网络馈电盒，直接通过网线为摄像头供电，无须布电源线，极大地简化了安装程序和施工难度。系统功能强大、使用灵活，使用计算机硬盘全数字化录像便于保存和检索。无需更换磁带。提供回放和搜索功能，优于模拟系统中相应功能。在网络中的每一台计算机，只要安装了客户端的软件，给予相应的权限就可成为监控工作站。

4. 优秀扩展能力

该系统可在网络环境许可下，极大限度的扩充摄像头的数量，而无须烦琐的配套工作。用户可以在任何时候方便快捷地将新的 IpCamera 添加到网络中。

只要有网络的地方增加监控点设备就可扩展新的监控点。

5. 先进性

该系统利用现有的综合布线网络传输图像，并进行实时监控。系统所需的前端设备少，连线简洁；后端仅需一套软件系统即可。

6. 可靠性

该系统的主要设备 IpCamera 和网络摄像服务器都采用了嵌入式实时操作系统，所需设备简单，而图像的传输是通过综合布线网络实现的，系统的可靠性是相当高的。

7. 安全性

该系统设置了不同等级的使用者权限，仅有最高级权限的用户才可对整个系统进行设置或更改。图像数据的存储是专有的格式。

8. 使用及维护性

该系统的安装极其简单，软件系统的安装及使用也非常易懂。在维护性方面，系统的接线十分简洁，而主要设备的可靠性很高，维护性能好，维护费用低，而且可实现远程维护，网络维护由网络提供商维护，前端设备是即插即用、免维护

系统。

2.6 未来视频监控系统的的发展方向

IBM 在 1999 年创造了一个名词——“普适计算”（又叫“普及计算”，Ubiquitous Computing）。“普适计算”指的就是，“无论何时何地，只要您需要，就可以通过某种设备访问到所需的信息”，即把计算机嵌入到环境或日常工具中去，让计算机本身从人们的视线中消失，让人们的注意力回归到要完成的任务本身。有人预言，在不远的将来普适计算必将得到广泛的应用，那时，各种具有计算和联网能力的设备将变得像现在的水、电、纸、笔一样，随手可得，人与计算机的关系将发生革命性的改变，变成一对多、一对数十甚至数百，同时，使用计算机的人既有具备一定计算机知识的专业人员，同时也包括普通的老百姓。那时的多媒体监控系统必将是基于普适计算的系统，其监控水平将更高，使用起来将更方便。

那么，什么是普适计算呢？通俗来讲，普适计算的含义十分广泛，所涉及到的技术包括移动通信技术、小型计算设备制造技术、小型计算设备上的操作系统技术及软件技术等。普适计算是指无所不在的、随时随地可以进行计算的一种方式；主要针对移动设备，比如信息家电或某种嵌入式设备，如掌上电脑、BP 机、车载智能设备、笔记本计算机、手表、智能卡、智能手机（具有掌上电脑的一部分功能）、机顶盒、POS 销售机、屏幕电话（除了普通话机的功能还可以浏览因特网）等新一代智能设备。普适计算设备可以一直或间断地连接着网络。与 Internet、Intranet 及 Extranet 连接，使用户能够随时随地获取相关的各种信息，并做出回应。用户可以通过使用各种各样的普适计算设备访问后台数据、应用和服务。无论使用何种普适计算设备，用户将能轻易地访问信息，得到服务。普适计算降低了设备使用的复杂性，帮助提高在外办公人员的效率和人们的日常生活水平^[6]。

普适计算是一种新兴的计算模式，与传统的计算模式相比有很大的不同，对设备硬件、操作系统、网络协议等的要求均有别于以往的计算模式。普适计算最显著的特点是计算的不可视性。以往的计算模式注重的是计算本身，人们的注意力往往集中在计算机上而不是任务本身上，这是不正确的方向。普适计算的不可视性把人们的注意力又重新拉回到任务本身上，计算是嵌入在任务里面的，对用户体现出不可视性，这样，用户可以更好地集中精力去完成任务，这才是正确的方向。

嵌入式 Internet 技术的发展使普适计算的深度和广度不断扩大，为提高视频监控系统的监控水平提供了巨大的可能性。

未来的基于普适计算的监控系统由于成本大幅度的下降而得到广泛的应用，

不仅在工业、电力、金融等部门中得到大量的使用，而且也将大范围的应用在智能楼宇、个人家庭中，使普通老百姓也能从中受益。

2.7 软件模型设计原则

开发软件是为了供别人使用，没有人使用的软件只是没有意义的数据的集合而已。许多在软件方面很有成就的行家在他们事业的初期却表现平平，因为在那个时候将主要精力都集中在技术上。显然，构件（components），EJB（Enterprise Java Beans）和代理（agent）是很有趣的东西。但是对于用户来说，如果设计的软件很难使用或者不能满足他们的需求，后台用再好的技术也于事无补。多花点时间到软件需求和设计一个使用户能很容易理解的界面上。好的软件设计人员把大多数时间花费在建立系统模型上，偶尔写一些源代码，但那只不过是为了验证设计过程中所遇到的问题。这将使设计方案更加可行，如果没有任何需求，就不要动手开发任何软件。成功的软件取决于时间（在用户要求的时间内完成）、预算和是否满足用户的需求。如果不能确切知道用户需要的是什么，或者软件的需求定义，那么工程注定会失败。Object ToolSmiths 公司的 Doug Smith 说：“分析是一门科学，设计是一门艺术”，意思是说在众多的“正确”分析模型中只存在一个最“正确”分析模型可以完全满足解决某个具体问题的需要。如果需求经常改动，很可能是没有作好需求分析，并不是需求真的改变了，需求真正改变的情况很少，但是没有做好需求分析工作的理由却很多。

高耦合度的系统是很难维护的。一处的修改引起另一处甚至更多处的变动。可以通过以下方法降低程序的耦合度：隐藏实现细节，强制构件接口定义，不使用公用数据结构，不让应用程序直接操作数据库。耦合度低的软件可以很容易被重用、维护和扩充。如果一个软件的模块只实现一个功能，那么该模块具有高内聚性。高内聚性的软件更容易维护和改进。判断一个模块是否有高的内聚性，看一看是否能够用一个简单的句子描述它的功能就行了。如果用了一段话或者需要使用类似“和”、“或”等连词，则说明需要将该模块细化。只有高内聚性的模块才可能被重用。

移植是软件开发中一项具体而又实际的工作，不要相信某些软件工具的广告宣传。即使仅仅对软件进行常规升级，也要把这看得和向另一个操作系统或数据库移植一样重要。记得从 16 位 Windows 移植到 32 位 windows 的“乐趣”吗？当使用了某个操作系统的特性，如它的进程间通信（IPC）策略，或用某数据库专有语言写了存储过程。软件和那个特定的产品结合度就已经很高了。好的软件设计者把那些特有的实现细节打包隐藏起来，所以当那些特性该变的时候，仅仅需要更新那个包就可以了。将所有系统将可能发生的变化以及潜在需求记录下来，以便将来能够实现通过在建模期间考虑这些假设的情况，就有可能开发出足够强

壮且容易维护的软件，设计强壮的软件是最基本的目标。

Internet 带给最大的教训是必须在软件开发的最初阶段就考虑软件规模的扩充性。今天只有 100 人的部门使用的应用程序，明天可能会被有好几万人的组织使用，下个月通过因特网可能会有几百万人使用它。在软件设计的初期，根据在用例模型中定义的必须支持的基本事务处理，确定软件的基本功能。然后，在建造系统的时候再逐步加入比较常用的功能。在设计开始考虑软件的规模需求，避免在用户群突然增大的情况下，重写软件。

关注软件设计中的一个重要因素——性能，这好像也是用户最关心的事情。一个性能不佳的软件将不可避免被重写，设计还必须具有可靠性，可用性，便携性和可扩展性。应该在工程开始就应该定义并区分好这些因素，以便在工作中恰当使用。性能可以是，也可以不是优先级最高的因素，给每个设计因素应有的考虑^[7]。

“UML User Guide”中指出，在开发阶段的早期就定义软件模块之间的接口。这有助于开发人员全面理解软件的设计结构并取得一致意见，让各模块开发小组相对独立的工作。一旦模块的接口确定之后，模块怎样实现就不是很重要了。从根本上说，如果不能定义模块“从外部看上去会是什么样子”，肯定也不清楚模块内要实现什么。

在软件开发中没有捷径可以走。缩短在需求分析上花的时间，结果只能是开发出来的软件不能满足用户的需求，必须被重写。在软件建模上每节省一周，在将来的编码阶段可能会多花几周时间，因为在全面思考之前就动手写程序。为了节省一天的测试时间而漏掉了一个 bug，在将来的维护阶段，可能需要花几周甚至几个月的时间去修复。与其如此，还不如重新安排一下项目计划。避免走捷径，只做一次做对。在设计的时候应当先建立一个技术原型，或者称为“端到端”原型，以证明设计是能够工作的。应该在开发工作的早期做这些事情，因为，如果软件的设计方案是不可行的，在编码实现阶段无论采取什么措施都于事无补。技术原型将证明设计的可行性，从而将更容易获得支持。

目前，有大量现成的分析和设计模式以及问题的解决方案可以使用。一般来说，好的模型设计和开发人员，都会避免重新设计已经成熟的并被广泛应用的东西。目前有很多种类的模型可以使用，用例捕获的是系统行为需求，数据模型则描述支持一个系统运行所需要的数据构成。可能会试图在用例中加入实际数据描述，但是，这对开发者不是非常有用。同样，数据模型对描述软件需求来说是无用的。每个模型在建模过程中有其相应的位置，但是需要明白在什么地方，什么时候使用它们。当收集需求的时候，考虑使用用例模型，用户界面模型和领域级的类模型。当设计软件的时候，应该考虑制作类模型，顺序图、状态图、协作图

和最终的软件实际物理模型。程序设计人员应该慢慢意识到，仅仅使用一个模型而实现的软件要么不能够很好地满足用户的需求，要么很难扩展。

如果测试对软件来说是无所谓的，那么软件多半也没什么必要被开发出来。建立一个技术原型供技术评审使用，以检验软件模型。在软件生命周期中，越晚发现的错误越难修改，修改成本越昂贵。尽可能早的做测试是很值得的。不值得归档的工作往往也不值得做。归档设想，以及根据设想做出的决定；归档软件模型中很重要但不很明显的部分。给每个模型一些概要描述以使别人很快明白模型所表达的内容。如果有人说“使用某种开发语言、某个工具或某某技术，就不需要再做需求分析，建模，编码或测试”。不要相信，这只能说明他还缺乏经验。抛开技术和人的因素，实际上软件开发的基本原理自 20 世纪 70 年代以来就没有改变过，必须还定义需求、建模、编码、测试、配置、面对风险、发布产品和管理工作人员等等。

第三章 网络化摄像机监控软件的设计

3.1 网络化视频监控系统总体体系结构的描述

系统的总体体系结构，如图 3.1 所示。从图 3.1 中可以看到：系统的总体体系结构主要包括两大部分：(1) 监控软件部分；(2) 现场设备部分。其中监控软件部分主要分两个程序：管理程序和监控程序，而现场设备部分即是本论文提到的产品 IpCamera 以及相应的云台。

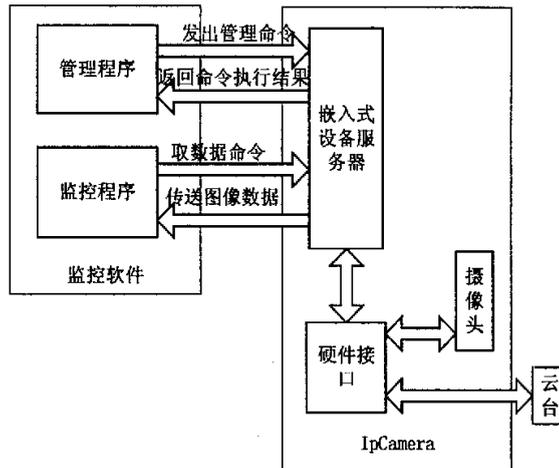


图 3.1 系统的总体体系结构

Fig.3.1 The architecture of system

IpCamera 分图像采集部分和嵌入式模块部分。图像采集部分是指用于采集图像的普通摄像头，它负责图像的采集，把光学信号变成电学信号，与此同时嵌入式模块利用压缩芯片把采集到的图像数据压缩成 MJPEG 格式以便达到减小网络传输量的目的，然后利用嵌入式模块的网络部分把处理好的图像传到网络上，来实现远程监控的目的。嵌入式模块是本系统得以实现得重要的部分，正因为有它的存在，才使得普通的摄像头具备网络功能，可以不通过 PC 机而直接把数据发到互联网上，有效地减少了用户的投资，降低了操作的复杂性，同时也减少了病毒的破坏作用。用户就可以在远程通过 Internet 利用本论文设计的监控软件对 IpCamera 进行访问、控制与管理等操作，使视频监控系统的水平得到了提高。该控制模块之所以叫“嵌入式”控制模块因为它是在嵌入式 Internet 技术基础上设计而成的，其可以实现服务器的功能，我们称它为 Webit，如图 3.2 所示。图 3.2 中，虽然为了看的清晰把它画得很大，但事实上它的体积很小，这一点可以通过

和旁边一元钱的对比看出来。它不过比一个普通的一元钱大那么一点点。但它的作用却是巨大的。使用时把 Webit 嵌入到普通得摄像头中，成为其中的一部分。

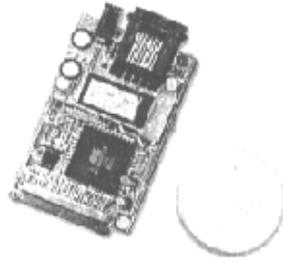


图 3.2 Webit 的外观

Fig.3.2. The appearance of Webit

虽然我们利用 Webit 实现了原来 PC 机实现的工作,但它和 PC 机是不一样得,速度和内存容量等等地方都有很大的不同,不能简单利用和 PC 机通信得方法来和它通信,并且还需要对摄像头及其所在云台进行一系列焦距、方向、环境和亮度等等调整,所以为了实现更好的监控,提高监控的保密性,我们需要设计一个专门的监控软件来实现视频监控的功能,方便用户的使用,一个视频监控系统客户端是一个非常重要的部分,这是本系统中唯一的和用户打交道得地方,用户通过它来进行远程监控,因此客户端部分设计的好坏,界面的美观与否将直接影响到用户的使用,所以本论文中精心设计一个功能完备、界面友好的监控软件。

网络化视频监控系统监控软件的设计要保证三点:

- 1) 设计要能满足客户的需求;
- 2) 设计要具有可持续发展的能力;
- 3) 设计的程序稳定性要好。

我们的设计就是以满足这三点为目的。在设计的过程中为了清晰的表达用户的需求,并使开发人员能够更好的理解设计思路,我们在系统建模时,采用了先进的 UML (Unified modeling language) 语言来表达整个模型及它们之间的相互关系。从需求分析到总体设计、功能实现直至系统测试都始终贯彻 RUP (Rational Unified Process) 的思想。

本系统预计实现如下的功能:

1. 登陆日志

功能描述: IpCamera 的主要用处就是用于视频监控,而视频监控的首要前提就是对安全性要求较高。为了防止内部人员渎职,利用登陆日志可以了解谁在什么时间进入本系统什么时间离开都保存在日志文件里,而日志文件只有权限最高的管理员可以进行删除操作,而其他用户则没有权限。

2. 监控功能

功能描述：这是视频监控中最重要的功能。合法的远程用户可以通过 Internet 利用本论文中设计的监控软件实时地获得远程 IpCamera 观测到的图像。本软件采用开辟录像缓冲区的方法来加强网络利用率，即使网络不是很稳定的环境下也可以使画面流畅。用户可以根据自己的需要进行调整以达到理想的观测画面。

3. 对 IpCamera 的参数进行配置

功能描述：合法的远程用户可以通过 Internet 利用本论文中设计的客户端远程配置 IpCamera 的各种参数，包括 IpCamera ID、IpCamera IP、环境、传输速度、图像格式、图像亮度和 IpCamera 位置描述等等。

4. 图像的录制与回放

功能描述：合法的远程用户可以通过 Internet 利用本论文中设计的客户端对目前观测到得图像进行录制。保存当前正在发生的事件，并可以利用配套设计的播放器来回放录制到的图像，每一帧图像都有与之相对应的时间，这可以作为追究责任的有效法律依据。本软件支持 16 路同时录制，可以抓取单帧，也可以录制连续帧。录制过程中还可以通过调整取图片速度来增加或者减少每秒录制的帧数。

5. 对 IpCamera 及相应云台调整

功能描述：在观测过程中可以通过云台随时调整 IpCamera 的观测角度，如上下、左右转动等动作。并可以针对观测物体距 IpCamera 的距离相应的拉伸镜头、控制镜头光圈的大小、聚焦的远近、焦距的长短等参数，以改变图像的大小。还可以调整 IpCamera 抓取图像的亮度，清晰度，每秒采集的帧数，可以自动控制摄像头各类设备电源的开关，如 IpCamera 电源的开关及照明电源的开关等。

6. 多画面显示

功能描述：监控软件可以同时在一台 PC 上进行多路 IpCamera 同时监控，可以显示 1、4、9 和 16 个画面或者全屏显示，也可以随意安排每个监控画面对应的 IpCamera 地点。可以单画面放大，也可任意切换。可以对 16 路图像同时录像，回放是单画面的，但可以实现录像和回放同时进行，回放的画面是实时的，检索按时间和事件进行，极其方便。

3.2 系统的拓扑结构

系统的拓扑结构，如图 3.3 所示。从图 3.3 中可以看出，本嵌入式视频监控系统的拓扑结构是由 IpCamera、本地网和广域网三部分组成。

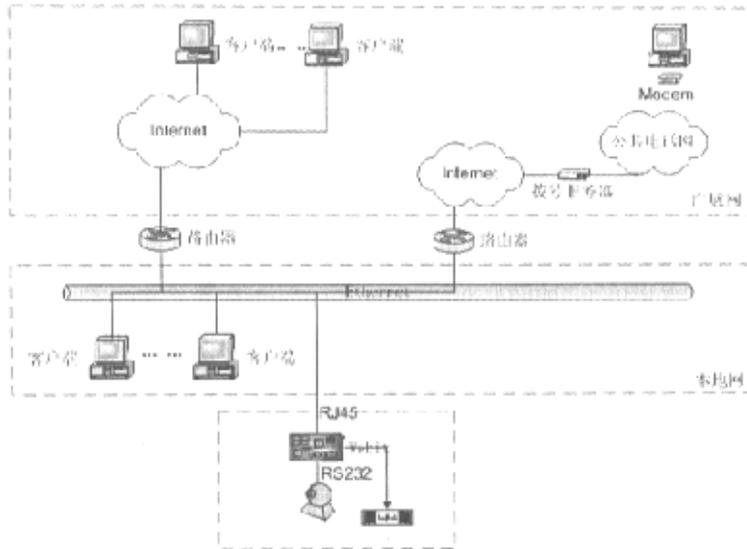


图 3.3 系统的拓扑结构

Fig.3.3 The topology of system

其中，系统的拓扑结构主要分为以下几个层面：

1. 现场设备：包括 Webit、云台和摄像头。摄像头采集的图像信息经 JPEG 压缩芯片压缩后通过 Webit 的网络接口上传到网络中。
2. 本地网：采用 Ethernet，通过 RJ45 接口与现场中的 Webit 相连，通过路由器与 Internet 相连，将现场监控设备发送过来的图像数据信息传给本网段的用户或通过路由器传给 Internet 上的远程用户；反之，也可将本网段或远程网段的用户发送过来的各种控制命令下载到现场设备中。
3. 广域网：包括 Internet 网，同时也包括公共电话网。

从系统的拓扑结构中可以看出，视频监控方式变得更加丰富了，既有本地视频监控，又有远程视频监控，大大方便了用户的使用。

3.3 监控软件总体体系结构的设计

根据监控系统的要求将整个功能分两部分程序来实现。一部分是管理程序，负责 IpCamera 和用户的管理，只有管理员才有权限使用它来控制所有的 IpCamera、配置 IpCamera 的各种参数、添加删除用户、修改用户权限，完成监控的前期工作。另一部分为监控主程序它具体负责操纵 IpCamera 进行监控工作。

整个系统工作过程如下：

1. 管理过程

客户端的管理程序发出搜索命令得到本网段中所有可操作的 IpCamera，分别为它们分配初始参数：IpCamera ID、IpCamera IP、环境、传输速度、图像格式、

和 IpCamera 位置等参数，这是监控的前期准备工作。

2. 监控过程

监控程序发送连接请求通过 Internet 传送到 IpCamera 嵌入式设备服务器以太网接口，嵌入式设备服务器响应这一请求，解析客户端发来得命令，开始进行图像采集并向客户端发出连接成功命令。客户端在得到响应后可以对 IpCamera 发出取数据命令，嵌入式设备服务器响应并解释这个命令后，向客户端发送一帧图像数据。

3. 控制过程

在监控过程中监控程序可以随时向 IpCamera 发出控制命令，IpCamera 接到命令后进行解析，如果符合实现规定的协议则按照客户的要求对摄像头或者云台进行相应的改变并回应客户端操作成功，否则回应失败信息。

根据上述工作过程将整个系统由几个模块组成，如图 3.4 所示。

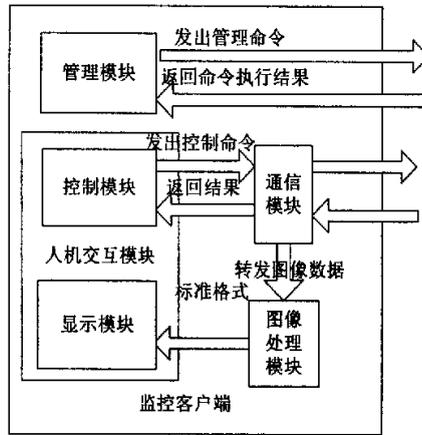


图 3.4 系统的软件体系结构

Fig.3.4 Software structure of the system

从图 3.4 中可以看出该软件体系结构是由人机交互模块、通信模块、图像处理模块和 IpCamera 管理模块组成，它们是该系统的核心结构，其中人机交互模块主要分显示模块和控制模块两部分。

模块间的协同工作过程如下：

1. 图像显示过程

图像的显示过程，如图 3.5 所示。用户启动图像模块的 play() 函数，图像模块向通信模块缓冲区中添加取数据命令，通信模块将这个命令发送到 IpCamera 端，IpCamera 根据这个请求开始工作，并根据收到的命令将存储区的图像数据发送到客户端的监控程序，监控程序的通信模块进行简单解析后将数据交给图像模

块，图像模块负责图像的解压工作将解压过的图片数据放在图像缓冲区中，通过单独子线程定时读取缓冲区中图像信息完成图像显示。图像模块根据用户的需要还可以进行图像保存工作，保存下来的数据可利用回放模块重新显示给用户。

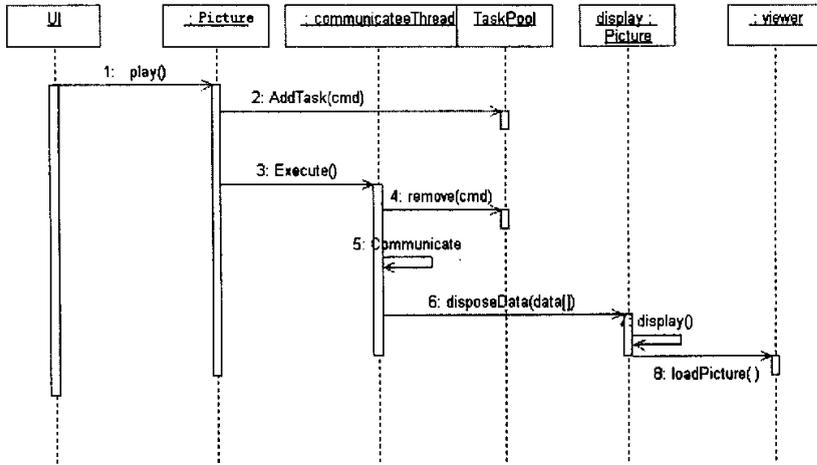


图 3.5 图像播放过程设计图

Fig.3.5. The UML chart of video playing process

2. 控制过程

控制过程如图 3.6 所示，控制模块向通信模块缓冲区中添加控制命令，通信模块将这个命令发送到 IpCamera 端，IpCamera 根据这个请求做出相应改变，并将操作的结果返回监控程序，通信模块进行简单解析后将数据交给控制模块，控制模块根据返回的信息判断是否改变控制按钮状态。

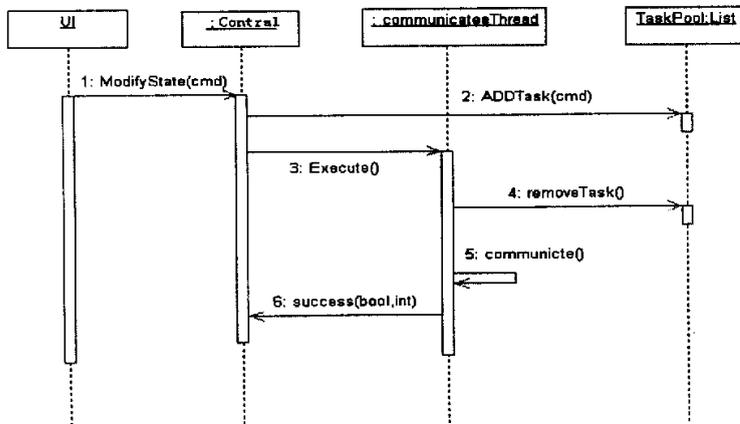


图 3.6 控制过程设计图

Fig.3.6. The UML chart of control process

3. 动态绑定过程

动态绑定过程如图 3.7 所示。因为我们的工作不是对一个 IpCamera 进行操作，需要同时监控几个 IpCamera，所以要让每个 IpCamera 都能正常工作，我们必须为每个 IpCamera 配置显视模块、控制模块、图像处理模块和通信模块。而它们之间的关系不是固定不变的，所以我们这里采用动态绑定的方法，通过我们的模块管理临时确定和取消它们的对应关系。

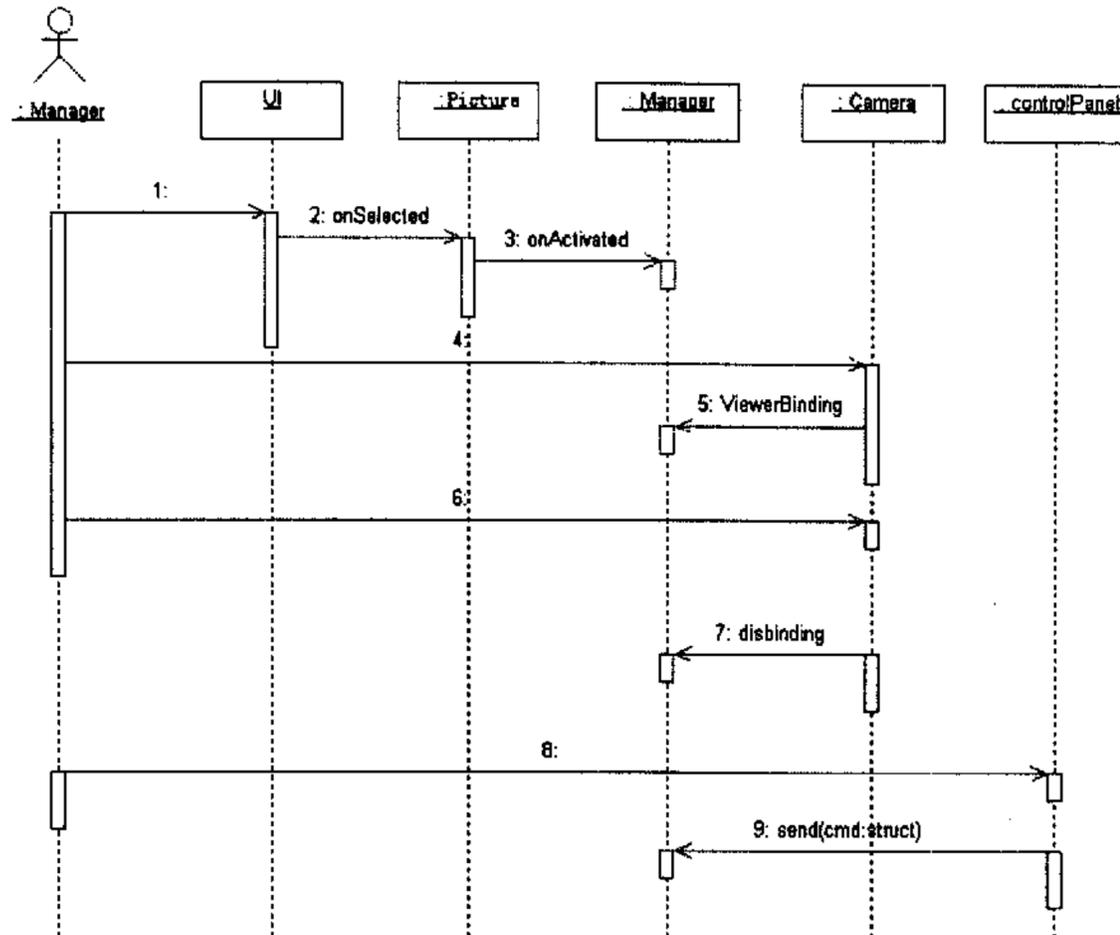


图 3.7 控制过程设计图

Fig.3.7. The UML chart of control process

3.3.1 人机交互模块的设计

人机交互模块界面的设计，如图 3.8 所示。从图 3.8 中可以看出，其由人机交互模块的界面由用户登陆界面、监控界面和控制界面三部分组成。

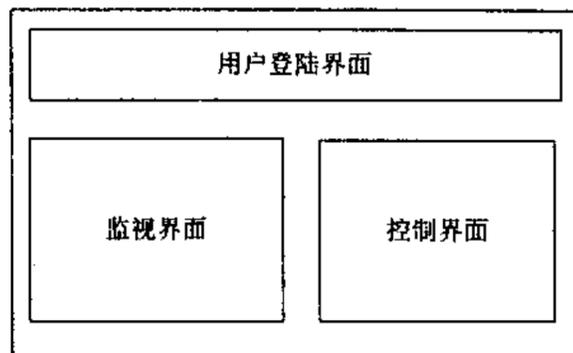


图 3.8 视频监控系统客户端界面

Fig.3.8 The client interface of visual monitor system

用户登陆界面是用户首先进入的界面，用于用户的登陆操作，用户在这个界面中输入用户名、密码这两个安全认证信息，系统根据安全认证信息正确与否决

定用户是否能够进入到下一个界面——视频监控界面；用户对现场 IpCamera 的监控操作是在视频监控界面中完成的，这些监控操作包括通过 Internet 对 IpCamera 参数进行的改变、调整摄像头焦距，调整云台角度，调整每秒采集的帧数，改变画面格式。通过这些改变可以使用户的观测角度和图像清晰度达到最佳，有效的实现视频监控。

监控软件可以同时在一台 PC 上实时监视 16 个点的现场图像，可以实现单画面、4 画面、9 画面、和 16 画面显示，满足了不同用户的需求。并且可以对每一监控画面进行放大，每一监控画面和与之对应的 IpCamera 之间是动态绑定的关系。因此可迅速的任意切换画面对应的 IpCamera。可以对 16 路图像同时录像。回放是单画面的，但可以实现录像和回放同时进行。回放的画面是实时的，可以判断每一帧图片发生的时间，检索按时间和事件进行，极其方便。

对 IpCamera 的控制采用异步方式，即控制面板直接把命令放入通信模块缓冲区，然后继续执行其它操作，等到该命令运行结果返回后再对面板相应属性加以改变。

3.3.2 通信模块的设计

因为整个监控过程中频繁的发生数据交互所以我设计了一个通信模块。通信模块是整个客户端与外界交互的桥梁。所有的图像数据获取，控制命令发送等等都通过这个模块进行。它包括发送缓冲区和接受缓冲区，所有需要发送给 IpCamera 的命令都放在发送缓冲区中。接收到得数据放在接受缓冲区中进行简单判断然后转发给对应的模块。通信模块是整个视频监控系统的基礎，无论是观测、记录、控制、还是超时重传，高质量的通信构架是必不可少的保障。这个通信构架不仅要完成基本的数据传送的功能，而且还要有好的结构以支撑各种功能需求，保证良好的扩展性。

在设计客户端通信模块时，我们考虑了以下几点：

1. 由于 IpCamera 功能繁多，如果把通信和其他模块合在一起开发，容易顾此失彼。可能会因为开始的考虑不周全，造成在增加某项新功能时不得不修改通信底层和原来的实现代码，导致重复开发。所以我们采用单独的通信模块的方式把通信和具体功能分开，这样在增加新功能时就只需少量修改通信模块的设置，不必再改动其它模块的程序代码了。
2. 采用多线程并行处理的方式编程，多线程能够通过并行计算和共享内存提高代码效率和资源利用率。
3. 图像处理模块与通信模块之间可以使用队列进行通信，队列采用先进先出。另外，通过队列通信，也可以为今后增加的功能提供良好的扩展性。
4. 采用 UDP 协议来实现这个系统的数据传输，原因下文会提到，这里就不

再讲解。

对 IpCamera 和云台的控制我们使用自定义的协议，如表 3.1 所示。

表 3.1 命令格式

Table 3.1 The format of command

命令名称	命令内容	成功回应命令	失败回应命令	命令 ID
取图像	00 03 *	00 03 **	无	0
连接命令	00 00 27 11	00 00 27 11	无	1
云台 auto 关闭	00 05 00 12 00	00 05	00 06	2
QVGA	00 05 09 00 00	00 05	00 06	3
Indoor	00 05 0B 00 00	00 05	00 06	4
结束命令	00 04	无	无	5
Outdoor 命令	00 05 0B 01 00	00 05	00 06	6
Indoor 命令	00 05 0B 00 00	00 05	00 06	7
QCIF	00 05 09 01 00	00 05	00 06	8
VGA	00 05 09 02 00	00 05	00 06	9
亮度	00 05 02 ** 00	00 05	00 06	10
对比度	00 05 03 ** 00	00 05	00 06	11
饱和度	00 05 05 A5 00	00 05	00 06	12
雨刷 on	00 05 00 11 04	00 05	00 06	13
雨刷 off	00 05 00 12 04	00 05	00 06	14
I/O on	00 05 08 01 00	00 05	00 06	15
I/O off	00 05 08 00 00	00 05	00 06	16
云台自动	00 05 00 11 00	00 05	00 06	17
Iris +	00 05 00 0B 00	00 05	00 06	18
	00 05 00 07 00	00 05	00 06	
Iris -	00 05 00 0C 00	00 05	00 06	19
	00 05 00 07 00	00 05	00 06	
Focus +	00 05 00 0D 00	00 05	00 06	20
	00 05 00 07 00	00 05	00 06	
Focus -	00 05 00 0E 00	00 05	00 06	21
	00 05 00 07 00	00 05	00 06	
Zoom +	00 05 00 0F 00	00 05	00 06	22
	00 05 00 07 00	00 05	00 06	

表 3.1 命令格式 (续)

Table 3.1 The format of command (continued)

命令名称	命令内容	成功回应命令	失败回应命令	命令 ID
Zoom -	00 05 00 10 00	00 05	00 06	23
	00 05 00 07 00	00 05	00 06	
云台 上	00 05 00 01 00	00 05	00 06	24
	00 05 00 03 00	00 05	00 06	
云台 下	00 05 00 02 00	00 05	00 06	25
	00 05 00 03 00	00 05	00 06	
云台 左	00 05 00 04 00	00 05	00 06	26
	00 05 00 03 00	00 05	00 06	
云台 右	00 05 00 08 00	00 05	00 06	27
	00 05 00 03 00	00 05	00 06	
Auto1 开	00 05 00 11 01	00 05	00 06	28
Auto1 关	00 05 00 12 01	00 05	00 06	29

通过表 3.1 的协议我们可以发出相应的命令来实现 IpCamera 的某种改变, 如果操作成功则返回相应的成功命令给客户端。其中取图像的命令与其他的不同, 其它的都是固定不变的, 这个命令是顺序加一来实现的, 到 00, 03, ff 后变为 00, 03, 00 这样做主要是为了数据重传, 客户端收到的数据包应该是序号连续的, 如果不连续则说明有错误发生。或者当网络不好发生重传时客户端只需要发正确接收帧得序号加一即可。可以说基本实现了 TCP 得丢包重传机制, 这里之所以不用 TCP 而采用相对简单的 UDP 主要是考虑服务器端是嵌入式系统。它的存储空间较小, 而我们需要花费很大的空间来存放控制摄像头, 云台的程序, 采用 UDP 协议可以节省大量的程序空间, 使 IpCamera 工作更稳定。

3.3.3 图像处理模块的设计

1. 图像传送的设计

因为视频监控系统的功能就是图像显示, 所以这是整个系统中最重要也最难实现的一部分。在这一部分需要完成图像数据的解析, 以及图像的保存。时间的嵌入等等一系列问题。

如今在 Internet 上, 传统基于字符界面的应用逐渐被能够浏览图像信息的 WWW (World Wide Web) 方式所取代。WWW 尽管漂亮, 但是也带来了一个问题: 图像信息的数据量太大了, 本来就已经非常紧张的网络带宽变得更加不堪重负, 使得 World Wide Web 变成了 World Wide Wait。所以在网络上传输图片信息

的首要前提的就是对图像进行压缩。

那么为什么要压缩？答案是很显然的，你也能想得到。因为图像信息的数据量实在是太惊人了。举一个例子就明白：

传统的视频图像是模拟的讯号，而且各国的视频标准各有不同，例如台湾地区是使用美国标准的 NTSC，大陆及欧洲地区则是使用 PAL 系统，我们就 NTSC 系统而言，它每秒有 60 个 FIELD，其中单数的 FIELD 与双数的 FIELD 合成一个 Frame，每个 Frame 有 525 条扫描线，诚如大家熟知的，网际网络传输的资料均为数字式，所以，如果我们要在网际网络上传输视频图像则必须先将图像数字化。图像数字化分为两个步骤：一为取样 (Sampling)，二为量化 (Quantization)。图像的取样是二维的，通常是每个 Frame 的长与宽，例如 640×480 ， 800×600 等等，量化则是将已取样的 Pixel，设定适当的值，例如 0-255，0-65535 等等，假设我们以 640×480 来取样，以 16bits 来加以量化，则一秒钟视频图像的资料长度为：

$$640 \times 480 \times 16\text{bits}/8\text{bits} \times 30\text{frames}/\text{seconds} = 18432\text{k bytes}/\text{second} = 18\text{MB}。$$

$$\text{一分钟视频图像的资料长度为：} 18\text{MB} \times 60 = 1080\text{MB}。$$

$$\text{一小时视频图像的资料长度为：} 1080\text{MB} \times 60 = 64800\text{MB}。$$

想象一下这个资料量的巨大，我们一般拨接上网的频宽 33.6kbps，大约每秒只能传 3KB 的资料，如果以这样的频宽来传送一分钟的图像资料大约需要 4.2 天才能完成，计算如下： $1080\text{MB} \times 1024/3/60/24 = 4.2666$ 。

因此，不将数据压缩而直接传送，确定是不可行的，于是，图像压缩的技术在图像传输的应用上，便更突显其重要性。

总之，大数据量的图像信息会给存储器的存储容量，通信干线信道的带宽，以及计算机的处理速度增加极大的压力。单纯靠增加存储器容量，提高信道带宽以及计算机的处理速度等方法来解决这个问题是不现实的，这时就要考虑压缩。

压缩可分为两大类：第一类压缩过程是可逆的，也就是说，从压缩后的图像能够完全恢复出原来的图像，信息没有任何丢失，称为无损压缩；第二类压缩过程是不可逆的，无法完全恢复出原图像，信息有一定的丢失，称为有损压缩。选择哪一类压缩，要折衷考虑，尽管我们希望能够无损压缩，但是通常有损压缩的压缩比（即原图像占的字节数与压缩后图像占的字节数之比，压缩比越大，说明压缩效率越高）比无损压缩的高。

我们这里采用 JPEG 得压缩格式来对图像进行压缩，JPEG 是联合图像专家组 (Joint Picture Expert Group) 的英文缩写，是国际标准化组织 (ISO) 和 CCITT 联合制定的静态图像的压缩编码标准。和相同图像质量的其它常用文件格式（如 GIF，TIFF，PCX）相比，JPEG 是目前静态图像中压缩比最高的。我们给出具体的数据来对比一下。例如采用 Windows98 目录下的 Clouds.BMP，原图大小为 640

×480, 256 色。用工具 SEA (version1.3) 将其分别转成 24 位色 BMP、24 位色 JPEG、GIF (只能转成 256 色) 压缩格式、24 位色 TIFF 压缩格式、24 位色 TGA 压缩格式。得到的文件大小 (以字节为单位) 分别为: 921,654, 17,707, 177,152, 923,044, 768,136。可见 JPEG 比其它几种压缩比要高得多, 而图像质量都差不多 (JPEG 处理的颜色只有真彩和灰度图)。正是由于 JPEG 的高压缩比, 使得它广泛地应用于多媒体和网络程序中, 例如 HTML 语法中选用的图像格式之一就是 JPEG (另一种是 GIF)。这是显然的, 因为网络的带宽非常宝贵, 选用一种高压缩比的文件格式是十分必要的。

JPEG 文件大体上可以分成以下两个部分: 标记码 (Tag) 加压缩数据。先介绍标记码部分。JPEG 的每个标记都是由 2 个字节组成, 其前一个字节是固定值 0xFF, 每个标记之前还可以添加数目不限的 0xFF 填充字节 (fill byte)。

标记码部分给出了 JPEG 图像的所有信息 (有点类似于 BMP 中的头信息, 但要复杂的多), 如图像的宽、高、Huffman 表、量化表等等。标记码有很多, 但绝大多数的 JPEG 文件只包含几种。标记码的结构, 如表 3.2^[8-11]:

表 3.2 标记码结构
Table.3.2 Jpeg format

标记码结构	含义
0xFF 0xD8	图像开始 SOI
0xFF 0xE0	JFIF 应用数据块 APP0
0xFF 0xE1 - 0xEF	其他的应用数据块 (n, 1~15) APPn
0xFF 0xDB	量化表 DQT
0xFF 0xC0	帧开始 SOF0
0xFF 0xC4	霍夫曼 (Huffman) 表 DHT
0xFF 0xDA	扫描线开始 SOS
0xFF 0xD9	图像结束 EOI

其中有一些是必需的, 如图像高度宽度、量化表和霍夫曼 (Huffman) 表等等考虑到网络流速得不同, 为了使本产品的适用范围更广, 我们采用了两种图像格式, 一种是 VGA 格式, 一种是 CIF 格式。VGA 是 640×480 每帧图片大小约为 10k, CIF 是 320*240 每帧图片大小约为 3k, 这样在网络环境较好的情况下可以采用 VGA 格式, 图像比较清晰, 网络环境较差的情况下可以采用 CIF 格式, 图像较小, 放大后图像相对较差, 但是可以达到画面流畅。

对 JPEG 的图像格式进行分析后, 我们发现可以把图像高度宽度, 量化表, 霍夫曼 (Huffman) 表等等参数固定, 采用固定的参数来对采集到得图像进行压缩。这样这些参数就可以作为 IpCamera 与客户端双方共知的参数不在网络上进行

传输，只在网络上传输压缩过的图片数据部分，这部分参数数据有 0.6K，而 CIF 格式的图片大小为 3K，这样每帧图像数据量减小了 20%，很显然在条件相对恶劣的网络环境下这对保持图像连续性的作用是相当巨大的。

为了保持图像连续性，我们设计了一个图像缓冲区，这里并不是收到一帧图像的数据等待显示后再去取第二帧数据，而是开辟了一个图像缓冲区，只要是缓冲区没满，就不停的取 IpCamera 端数据放在缓冲区中。然后按照客户的设定，每隔一定的时间取缓冲区中一帧图像。这样做的好处是最大限度得利用网络带宽，因为现实情况中的网速不是恒定不变的，它会有堵塞等等情况发生，如果是一帧一取显然网络不好时或发生图像冻结，但采用缓冲区技术后就很好的杜绝了这个现象。因为缓冲区不停的发生数据储存读取是一个生产者消费者问题，所以我们采用缓冲环技术来实现。所谓缓冲环就是程序初始化时申请的一个大的静态数组，我们把它的首尾逻辑上连续起来，看起来像个环型结构。将缓冲环的起始地址，终止地址，当前页的地址和界限地址分别保存在四个变量中。利用一定的操作可以实现图像数据的保存储存读取。缓冲环的工作方式如图 3.9:

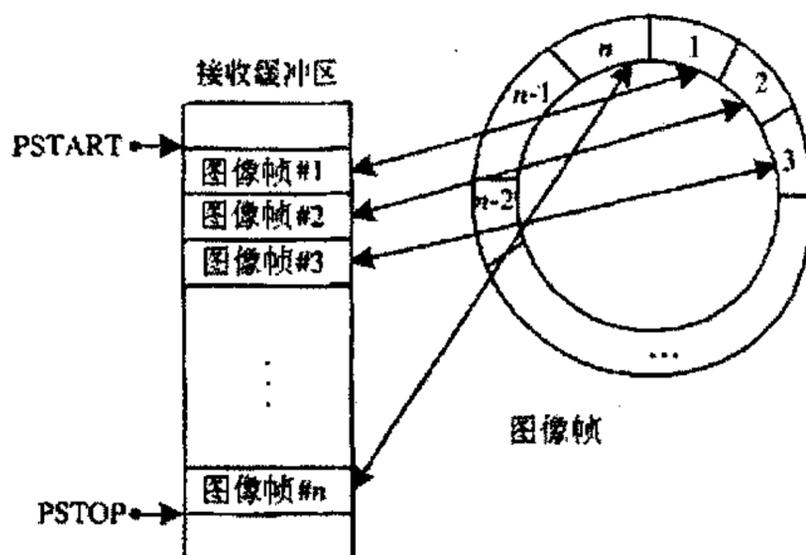


图 3.9 图像缓冲环

Fig.3.9 The buffer loop of image

当前页的地址指针保存用于接受图像数据的缓冲区地址，界限寄存器用于保存未被读取得第一帧图像。接收缓冲区分成 5 个 12K 字节的缓冲区，这样的缓冲区通过指针控制链接成一条逻辑上的缓冲环。缓冲环的开始页面地址存入变量 PSTART，环页面结束地址存入变量 PSTOP。PSTART 和 PSTOP 确定了接收缓冲环的大小和边界。为便于缓冲环读写操作，还需要 2 个指针：当前页面指针 CURR 和边界指针 BNRV。CURR 确定下一帧图像数据放在何处，起着缓冲环写页面指针作用；BNRV 指向未经显示模块取走处理最早到达的图像帧起始页面，新接收的图像数据不可将其覆盖，起着缓冲环读页面指针的作用。也就是说，CURR 可以告诉用户新接受的图像数据分组当前放到了什么位置，而 BNRV 则用于确定显示模块读缓冲环到了什么地方。由于接收缓冲区为环形结构，BNRV 和 CURR 相

等时，环缓冲区可能满也可能空。为了能辨别这两种状态，我们申请了一个变量 ImageNum 来保存缓冲区中当前图像数据个数。

2. 图像保存的设计

因为本系统是以视频监控为主要目的，单独的保存一帧图片或一段录像并没有太大的意义，所以要求把时间和对应的图像关联起来，办法可以有：

1) 不破坏原有图像数据以标准 JPEG 格式保存单帧图像，以标准 AVI 格式保存连续帧图像并以文件名等其他方式记录一段录像的开始时间和结束时间；

2) 不采用标准格式把每一帧图像采集得时间嵌到该帧图像的数据中。

这两种办法都可以实现时间与图片的一一对应但各有优缺点：

1) 方法一的优点主要是兼容性好，以这种方式保存的单帧和连续帧图像可以使用普通的 ACDSEE, Media Player 等等软件来观看，而方法二采用在 JPEG 压缩标准基础上自定义的格式，所以不能使用除本论文设计的播放器外其它软件来观看。

2) 方法一因为是利用开始时间和帧序号计算每一帧图像采集得时间，所以录制过程中必须采用匀速录制，如果在录制过程中调整录制速度比如由原来的每秒 15 帧变为每秒 10 帧那么就无法计算出每一帧图像采集的具体时间，而方法二因为是每一帧图片与其采集时间都是一一对应的所以可以随时在录制过程中改变采集速度。

3) 方法一如果保存得连续帧发生部分损坏，想还原其它部分数据比较麻烦。而方法二因为采用 MJPEG 格式，帧与帧之间没有联系，就等于是一张张 JPEG 图片堆砌在一起，其中一帧或几帧图片损坏完全不影响其他的帧，可以提高监控安全性。

3. 图像回放的设计

因为采集图像的保存是采用自定义的格式，所以需要设计一个专门的播放器来对保存好的图像进行回放。播放器采用不规则界面，圆形按钮，美观大方，方便实用。播放器的主要功能有：播放、暂停、停止、前进、后退、播放速度选择和按时间顺序查找等。

3.3.4 管理模块的设计

因为本产品主要是用来做视频监控的，所以安全性是排在第一位的。在这里把 IpCamera 管理和 IpCamera 监视分开，做成两个软件。只有高级用户才能有权限对 IpCamera 进行管理，而普通用户只能使用监视软件，这样可以防止内部员工的人为破坏，杜绝了内外勾结进行破坏活动的可能。

管理模块主要包括对 IpCamera 的管理和用户的管理。可以对 IpCamera 进行增加、删除、修改属性等操作。对用户进行添加、删除、改变用户级别等操作。

采用进程间同步通信技术来实现 IpCamera 的动态添加。可以在不重启监控程序的基础上动态添加或删除 IpCamera。

第四章 网络化摄像机监控软件的实现

网络摄像机监控软件使用 C++ Builder 语言为开发工具^[12-13]，客户端的开发/运行平台是 Windows 2000 操作系统，实行可视化开发。

4.1 用户界面的实现

这一部分分两个子模块：显示模块和控制模块。他们都是和用户打交道得部分，技术含量相对较低，实现时主要考虑的是界面的友好性，合理性。

如图 4.1 所示，这部分主要是通过编写函数实现当监控窗口数目发生改变时重新计算每个监控窗口（图 4.1 中所示每一个部分为一个监控窗口）的位置或者当前窗口发生大小改变时计算新窗口的位置。

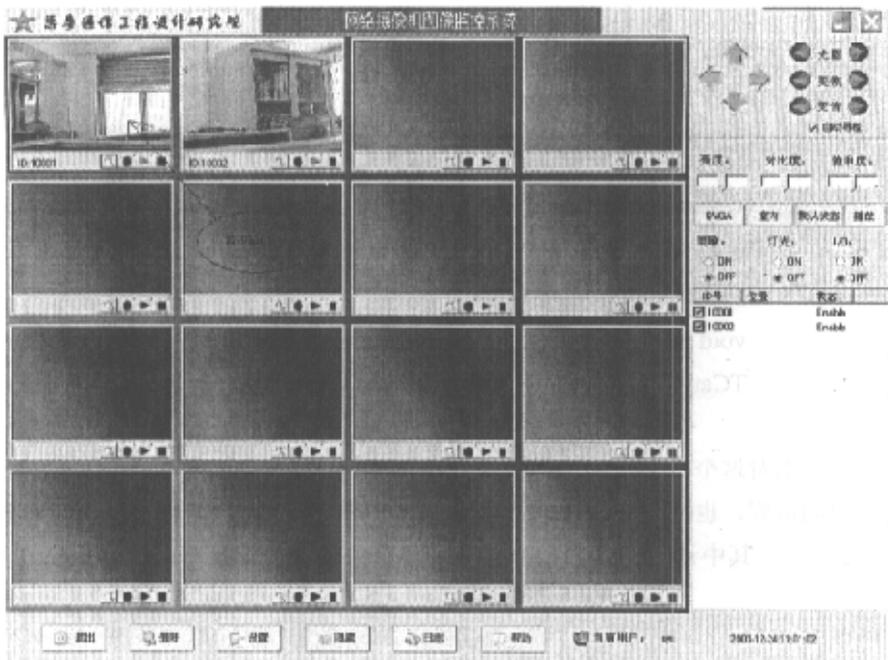


图 4.1 监控窗口

Fig.4.1 The monitoring window

将监控窗口做成一个可视化类，这样当采用多窗口监控时，所要做的就是循环实例化这个类，把它们放在了 TList 容器物件中，然后指定每个实例观测窗口的位置。这个类的定义为（只列出了较为重要的变量及成员函数）

```
class TViewer : public TFrame
{
```

```

__published: // IDE-managed Components
    void __fastcall bbtnPlayClick(TObject *Sender);
                                                //发送连结命令, 开始播放
    void __fastcall PlayStopClick (TObject *Sender);
                                                //发送结束命令, 停止播放
    void __fastcall RecordSeqClick (TObject *Sender);
                                                //连续帧录制函数
    void __fastcall RecordSingleClick (TObject *Sender);
                                                //单帧采集函数

private:// User declarations
    TPoint points; //监控窗口位置
    int width,height; //监控窗口大小
    bool bindingStatus; //绑定状态变量

public: // User declarations
    __fastcall TViewer(TComponent* Owner);
    void __fastcall LoadPicture(Graphics::TBitmap *Picture);
                                                //显示图像函数
    void __fastcall Bind(TCamera *aCamera); //绑定 IpCamera 函数
    bool __fastcall IsBinding(); //绑定状态判断
    void __fastcall Disbind(); //解除绑定函数
    TCamera *thisCamera; //需要绑定类指针
};

```

在这里对这个类的主要操作就是随着监控窗口的数目和大小来相应的改变监控窗口的的位置, 也即修改类 TViewer 中得窗口位置, 大小属性。实现的流程图如图 4.2 所示, 其中计算下一监控窗口位置的函数为

```

if((i+1)%4!=0)
    Position_Left+=Position_Width;
else{
    Position_Left=0;
    Position_Top+=Position_Height;
}

```

其中变量 Position_Left、Position_Top、Position_Width 和变量 Position_Height 确定了每一个监控窗口的位置

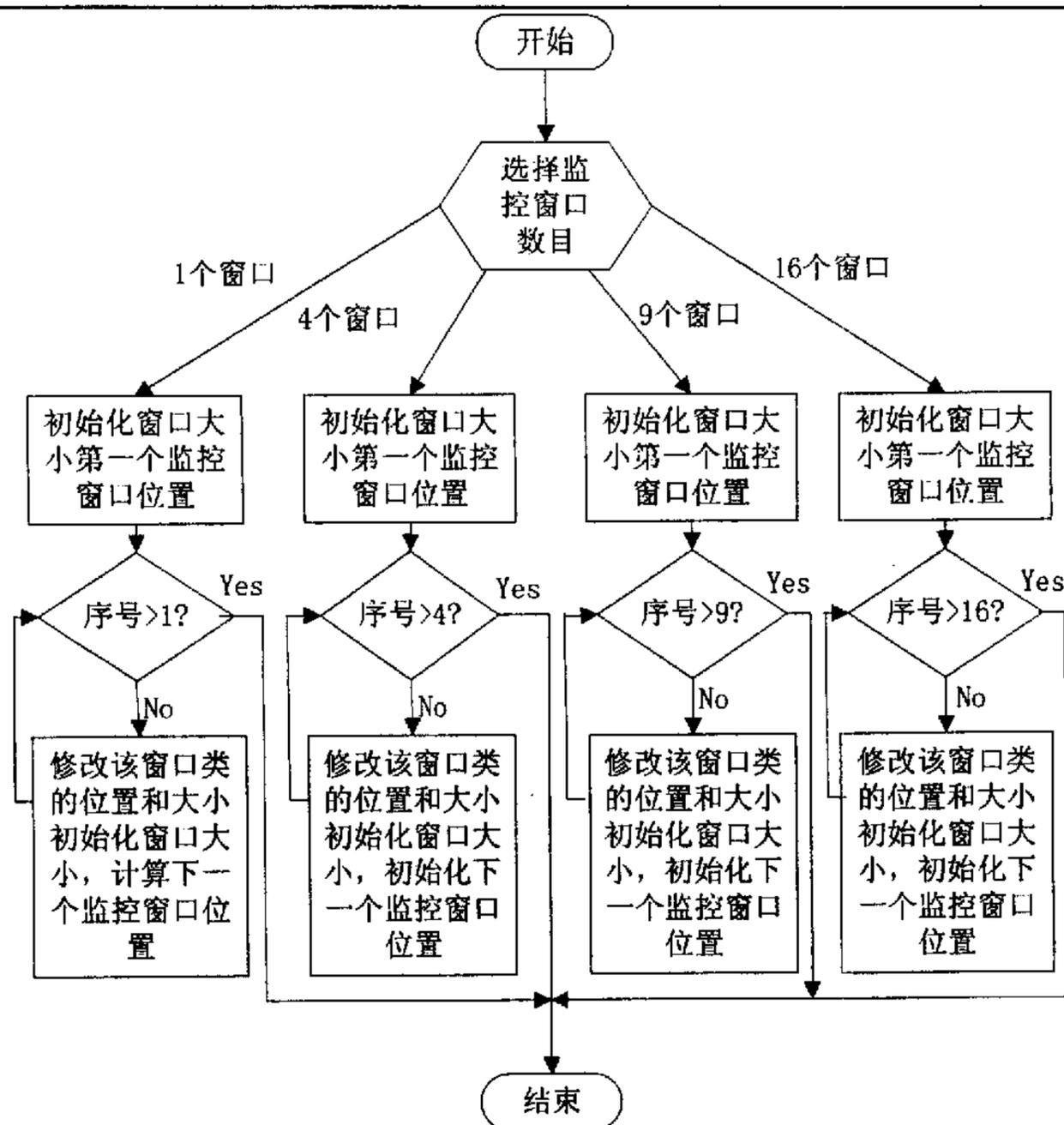


图 4.2 窗口位置计算程序流程

Fig.4.2 The flowchart of window's position

4.2 通信模块的实现

通信模块负责整个客户端与服务器端的通信工作，是整个程序中至关重要的一部分，如图 4.3 所示。它负责把发送缓冲区中的数据发送出去和简单解析接收到数据，并按照解析的结果把数据传给相应的上层模块。

从图 4.3 中可以看出这里采用一帧一应答的方式，即客户端每向服务器端发送一帧数据都要等到服务器应答后再进行下一帧数据的发送，如果收不到应答则重传该帧数据。这样图像显示的速度由客户端来控制。客户端取数据命令发送的快慢可以通过图像显示速度来控制。

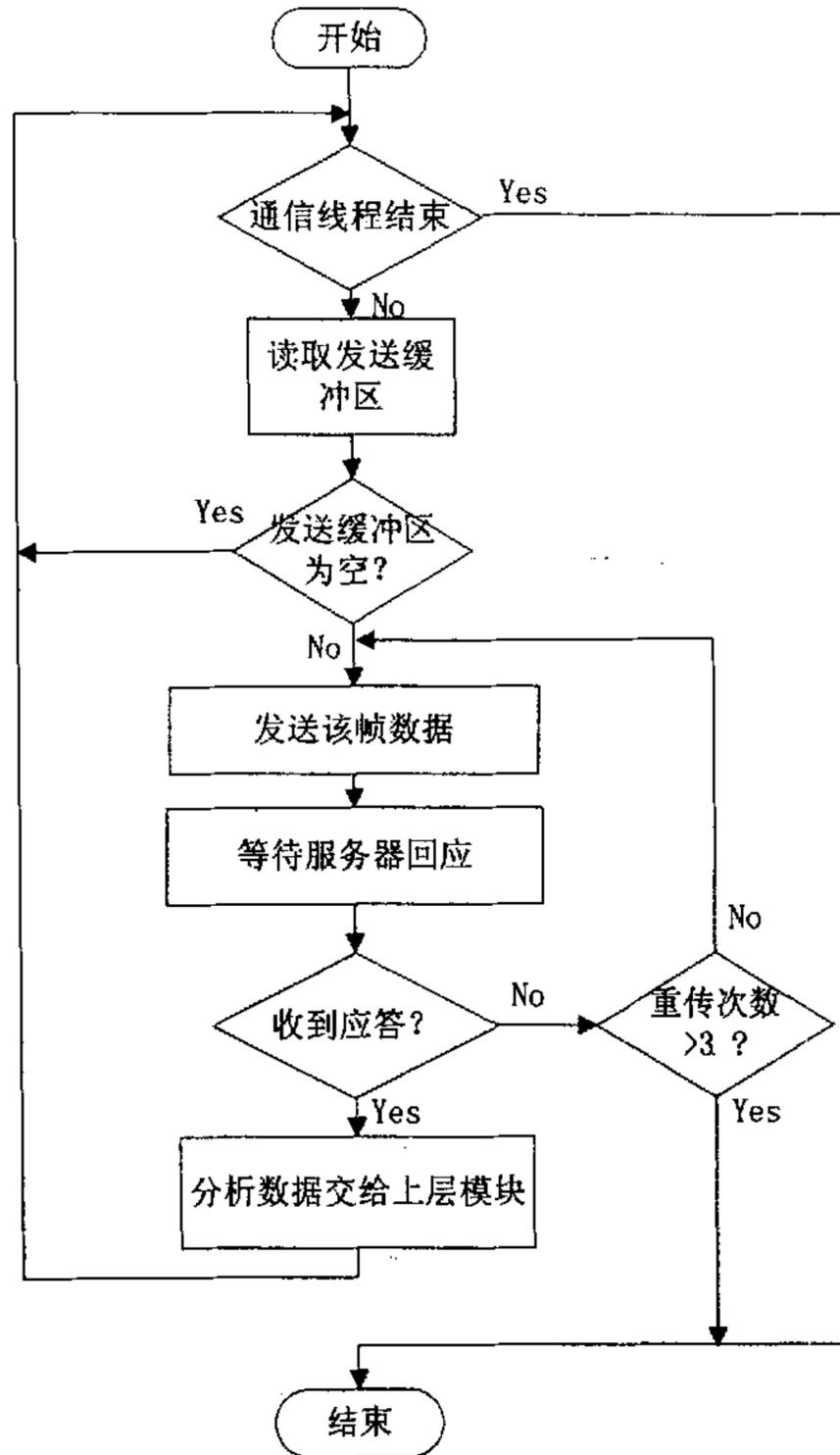


图 4.3 发送并接收 UDP 监控报文程序流程

Fig.4.3 The flowchart of sending and receiving UDP frame function

从上一章提到的控制协议格式中我们可以看出各个控制帧得正确应答都是 0x00, 0x05, 而且我们采用异步方式来处理控制命令。那么数据模块发送缓冲区中可能同时有很多数据, 那么怎么区分当前处理的是哪一个控制命令呢? 因此我们设计了一个如下结构作为缓冲区的基本结构

```

typedef struct _Cmd{
    unsigned char content[5];           //命令字符串
    int lengthOfContent;               //命令长度
    int ID;                             //命令 ID
    AnsiString cmdOfMeanings;         //该命令执行的操作解释
}Cmd;
    
```

可见我们只要根据 ID 这项参数参照表 3.1 就可以判断出控制模块发出的是什么控制命令。

通信类的定义

```
class TCommunicatee : public TThread
{
private:
    TIdUDPClient *UDPClient;           //通信用 SOCKET
    bool GetResult;                    //读通信缓冲区命令数据成功?
    AnsiString ip;                     //保存目前通信的 IpCamera IP
    int port;                           //保存目前通信的 IpCamera PORT
    Cmd *acmd,*ptr_cmd;                //命令缓冲池指针
    TCamera *aCamera;                  //IpCamera 类指针 (绑定用)
    bool bindingCondition;             //当前绑定状态

protected:
    void __fastcall Execute();

public:
    TList *TaskPool;                   //容器
    __fastcall TCommunicatee(bool CreateSuspended,TList *ATList);
                                     //构造函数
    void __fastcall Bind(TCamera *thecamera); //绑定函数
    bool __fastcall IsBinding();           //绑定状态判定函数
    void __fastcall Disbind();            //解除绑定函数
    void __fastcall Get();                //读控制命令缓冲区
    void __fastcall Set();                //写控制命令缓冲区
    __fastcall ~TCommunicatee();         //析构函数
};
```

4.3 图像处理模块的实现

这一部分是整个监控软件实现的主要部分，因为我们的客户端支持 16 路 IpCamera 同时监控，如果再几路同时录制的情况下，系统消耗将很大，所以这一部分实现的方法，采用的算法的效率等等将直接影响最后用户的观测效果。这一部分需要实现的功能也较多，实现时需要进一步细化。这一模块主要的功能是图片格式的调整，因为我们的图片并不是标准的 JPEG 格式，这一点模块设计中已经提及，这里不再赘述，所以需要在显示前对通信模块传送来的数据进行调整使它变为标准 JPEG 格式，这样才能用标准的 JPEG 解压函数。

4.3.1 图像数据重组的实现

IpCamera 端的工作过程就是简单的把图像压缩成一帧帧 JPEG 图片，然后去掉图像压缩参数部分发给监控客户端。因为每一帧压缩好的 JPEG 图像的大小都是几 K，如 VGA 为 11K，CIF 为 3K,可见每帧图像数据都需要分几包 UDP 包传送，所以图像模块需要处理得是把接收到的数据包合并起来按照固定的顺序放在一个缓冲区中，并加上正确得 JPEG 格式图片压缩参数部分重新恢复完整图片格式，如果保存标志为真，就把该帧图片信息写入磁盘，最后将该图像信息加入图像缓冲区，最后利用标准控件显示 JPEG 图像。图像在网络中传输的格式，如图 4.4 所示。

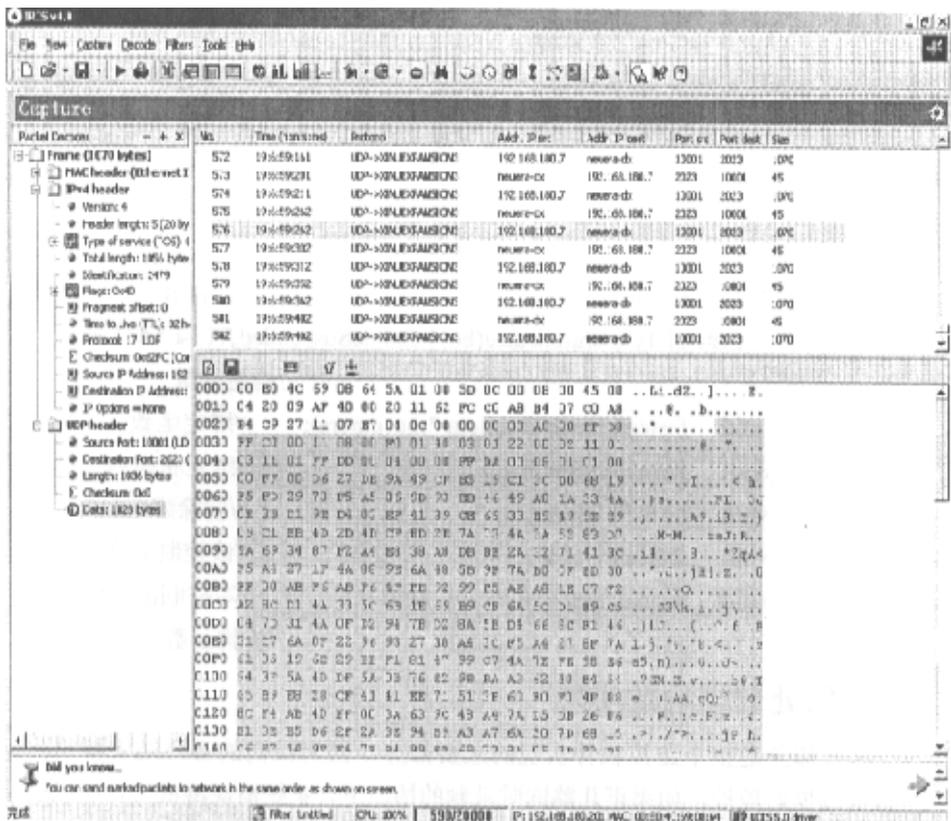


图 4.4 图像数据包

Fig.4.4 The data packet of frame

从图 4.4 中，可以看出 JPEG 图像有效数据从标志 FF, D8 开始，而前面还有 4 个字节的数据，这就是前面所讲到的保证不丢包而且包顺序正确的标志。图像显示的流程，如图 4.5 所示。

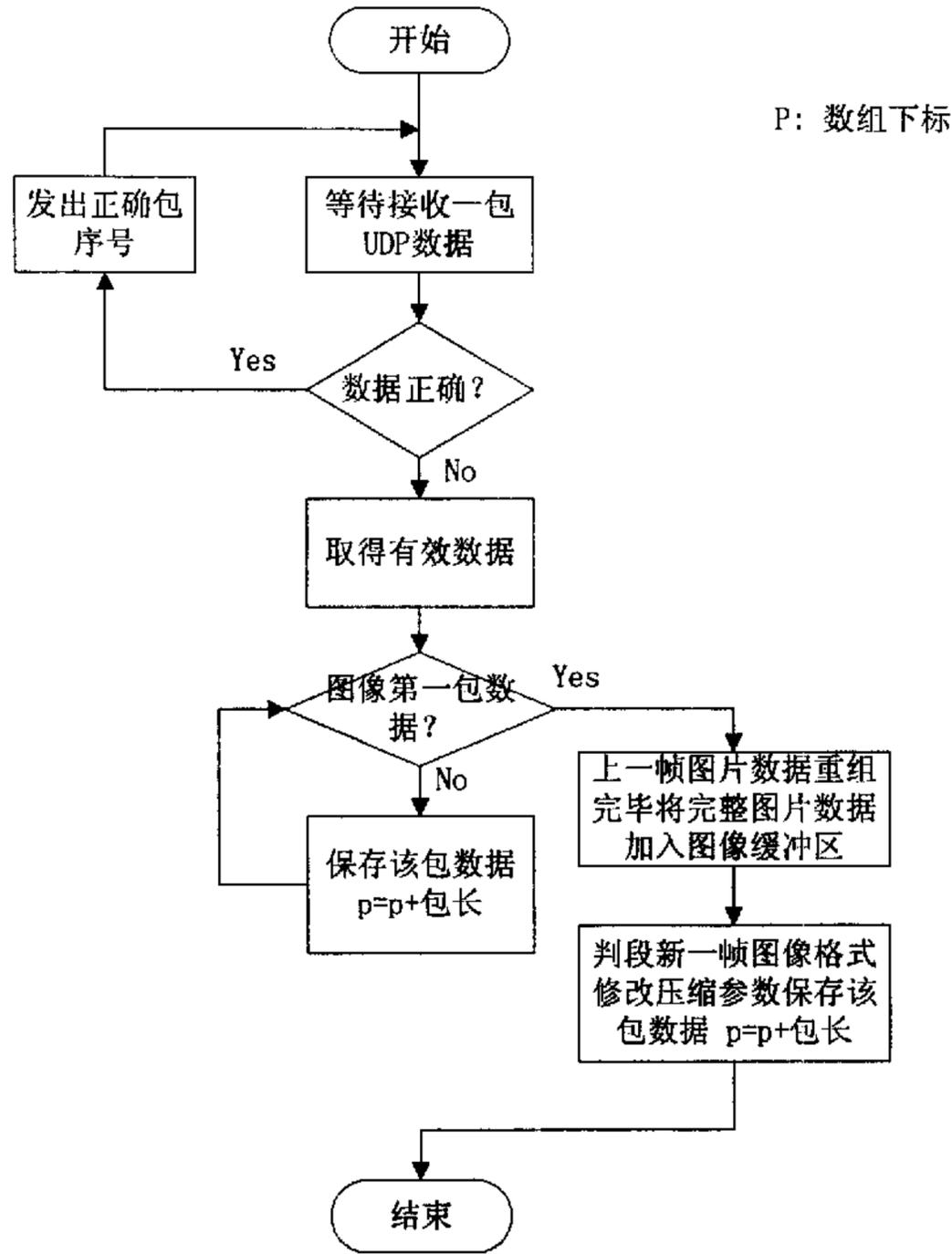


图 4.5 图像数据重组

Fig.4.5 Regrouping image data

4.3.2 图像缓冲区的实现

为了更好的存储和读取缓冲区的图像数据，我们采用如下的结构存储一帧图像数据

```

struct _ImageData{
    int Data_Type;           //图像数据类型 (QVGA,VGA,CIF)
    int Data_length;       //图像数据长度
    int Data_Buf[15000];   //JPEG 图像数据
};
    
```

这里缓冲区规定最大可以有 5 个上述节点，即定义 struct _ImageData Image_Data[5]。

对缓冲区得操作主要有缓冲区的初始化，缓冲区的读取与缓冲区的接收。

1. 缓冲区的初始化

数组缓冲环的初始化很简单，就是对基本的指针进行初始化。

PSTART=缓冲区首指针= Image_Data[0];

PSTOP=缓冲区尾指针= Image_Data[4];

BNRY=PSTART;

CURR= PSTART;

Image_Num=0;

2. 缓冲区数据读取的实现

读取部分采用单读的线程来实现，如图 4.6 所示。它的任务就是在符合条件的情况下读取图像缓冲区，如果读取成功，则将读到的数据交给 TViewer 类进行显示操作，读取过程比较简单，就是判断一下变量 Image_Num，如果不为 0，就表示缓冲区非空可以进行读取。其中需要注意的地方是读取得时候需要注意临界区的问题。

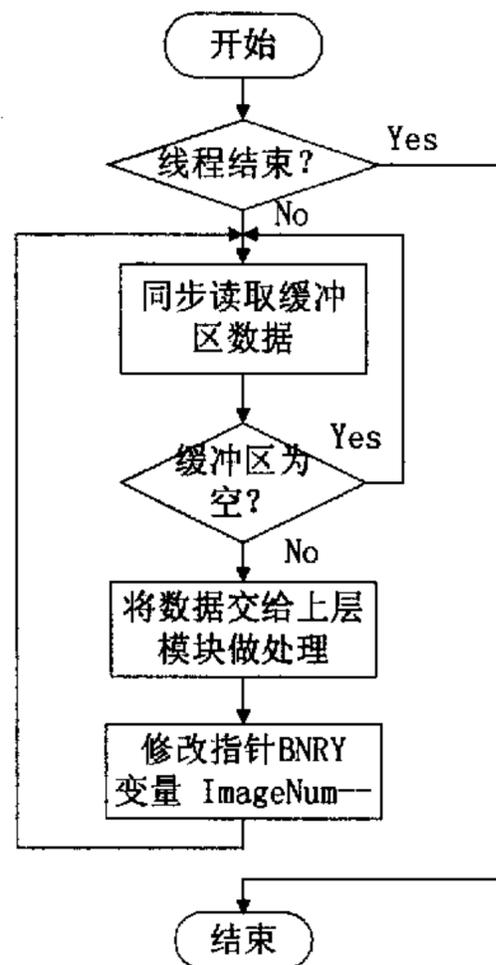


图 4.6 图像数据读取流程

Fig.4.6 The flowchart of getting image

3. 图像缓冲区数据写入的实现

每当图像数据重组子模块完成一帧 JPEG 图像的重组并简单判断该帧图像的正确性后，就将该帧图像数据和图像基本信息写入节点 struct Image_Data，然后判断缓冲区是否已满，不满则发出命令取下一帧数据。每次写入之前首先判断该帧数据是否需要保存，如需要则将其写入硬盘。当然写入时也要注意临界区的

问题，其工作流程如图 4.7 所示。

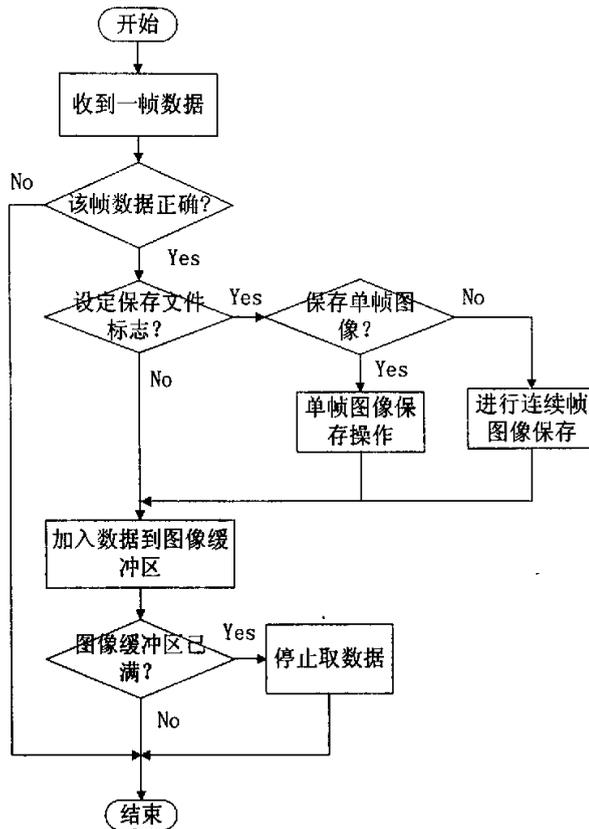


图 4.7 图像数据写入流程

Fig.4.7 The flowchart of putting image

其中保存数据的操作就是把图像数据写入硬盘，这是多监控窗口同时保存图像数据的瓶颈所在，为了提高该项性能，我们应该采用读写速度较快的硬盘。

4.3.3 回放模块的实现

回放模块是用来播放保存过的图像数据，将监控过程重现。界面较为流行的采用不规则的显示界面，如图 4.8 所示。

其中查询主要是通过拉动滑动条办法来实现，和普通的播放器使用方法一样，简单易用。我们采用 MJPEG 图像格式来保存采集到连续的图像，也就是一帧得 JPEG 图片依次存入磁盘文件，播放时也是简单地一帧 JPEG 图片交替显示。播放器解析图像文件的方法就是严格按照图 4.9、图 4.10 的文件保存格式读取文件信息，进行显示。



图 4.8 播放器界面

Fig.4.8 The player interface

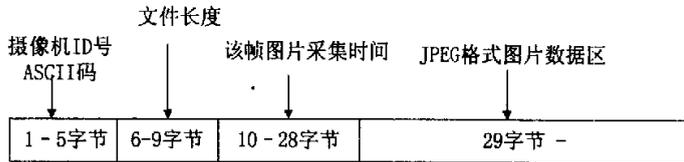


图 4.9 单帧文件格式

Fig.4.9 The format of single frame

如图 4.9 所示，前 28 个字节保存图像的附带属性，这 28 个字节是回放的主要依据。摄像机 ID 号是摄像机搜索时判断摄像机位置的依据，文件长度标识本帧图像的数据长度，采集时间和图片一一对应可以具体指明采集该帧的时间。

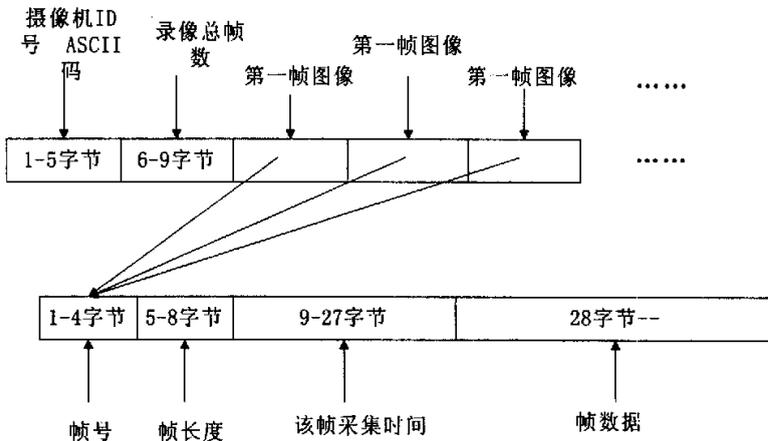


图 4.10 连续帧文件格式

Fig.4.10 The format of sequence frame

如图 4.10 所示，其中录像总帧数是用来初始化查找滑动条，帧号字段是用来做查找工作的。当拉动滑动条实行查找工作时我们需要进行计算把文件指针移到对应的位置上，算法的流程图，如图 4.11 所示。

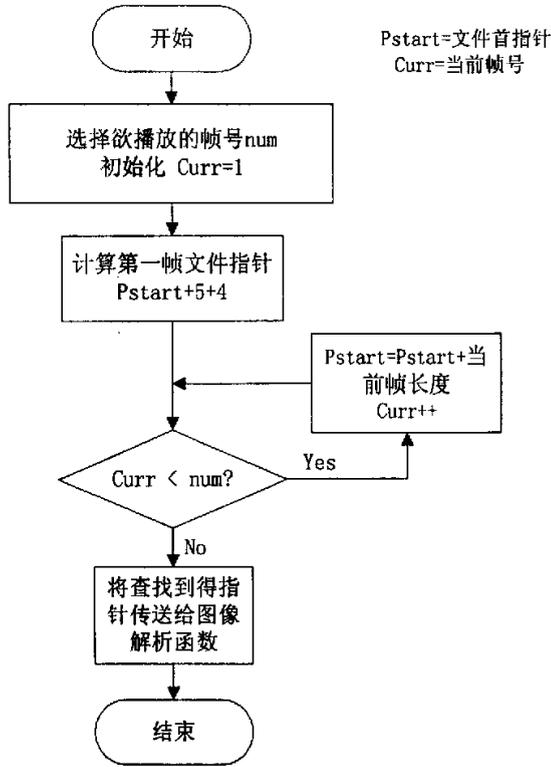


图 4.11 图片播放流程

Fig.4.11 The flowchart of playing

4.4 管理模块的实现

这一模块负责 IpCamera 和用户的管理，所以很显然应该单独做成一个程序，不然就起不到管理用户的作用，它的界面实现如图 4.12 所示。从图 4.12 中可以看出，我们对 IpCamera 实行分组管理，功能相同相近或者地域相同的 IpCamera 可以放在一组以便于管理。对用户也实行按职责划分操作权限。程序支持右键菜单功能，在需要操作的地方点击右键可以弹出操作菜单进行选择，操作直观方便。

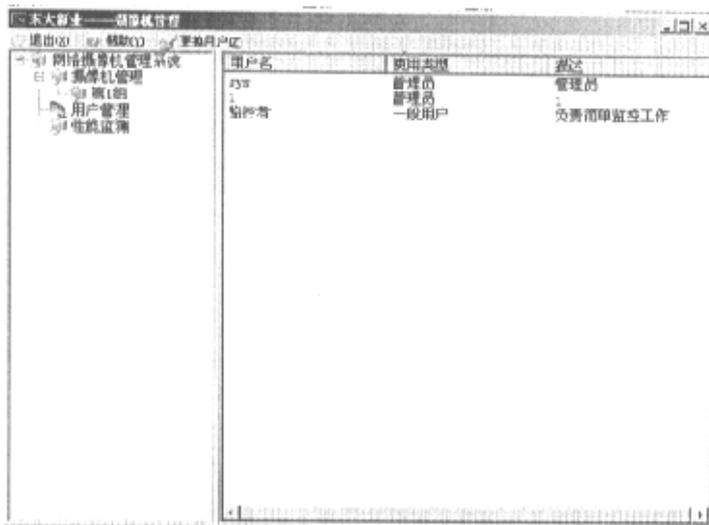


图 4.12 管理界面

Fig.4.12 Manage interface

从图 4.12 中可以看出这部分主要分为 IpCamera 管理、用户管理，性能监测三个部分。

其中，IpCamera 管理主要是搜索，修改 IpCamera 属性，可方便的通过点击右键或快捷键的方式调出界面，如图 4.13 所示。

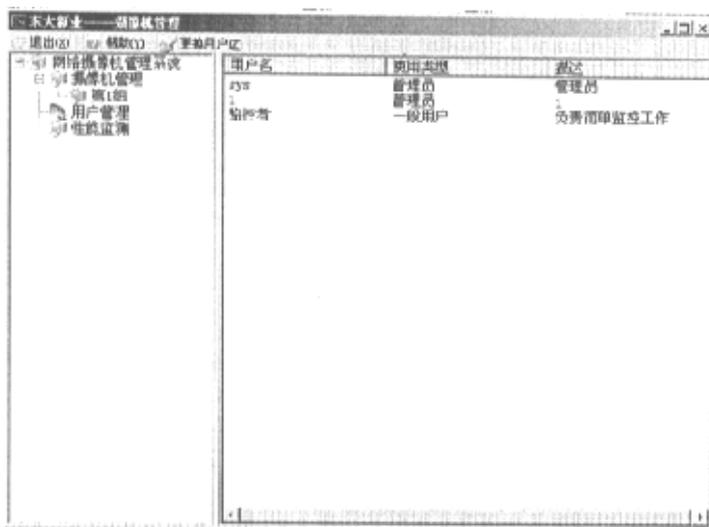


图 4.13 IpCamera 管理界面

Fig.4.13 The manage interface for IpCamera

从图 4.13 中可以看出可以修改 IpCamera 的各种属性。IpCamera 的 IP 地址是可以动态改变的。正因为如此所以我们采用在局域网内发固定格式广播包的方法来搜索 IpCamera, 发送数据包的格式为 0x00, 0x09, 0x65, 0x79, 0x65, 收到这个广播包的 IpCamera, 如果处在正常工作模式下就会做出反应, 向客户端回应它的 MAC 地址和 ID 号, 这里 MAC 是不可以改变的唯一标识 IpCamera 的参数。回应格式如下 0x00, 0x09+6 个字节 MAC 地址+2 个字节 IpCamera ID 号, 搜索到得 IpCamera 会在图 4.8 中箭头所示的部分以列表方式出现, 用户在列表框中做出选择然后修改图中所示各个参数得默认值点击确定按钮即可成功添加该 IpCamera。搜索 IpCamera 的工作流程, 如图 4.14 所示。

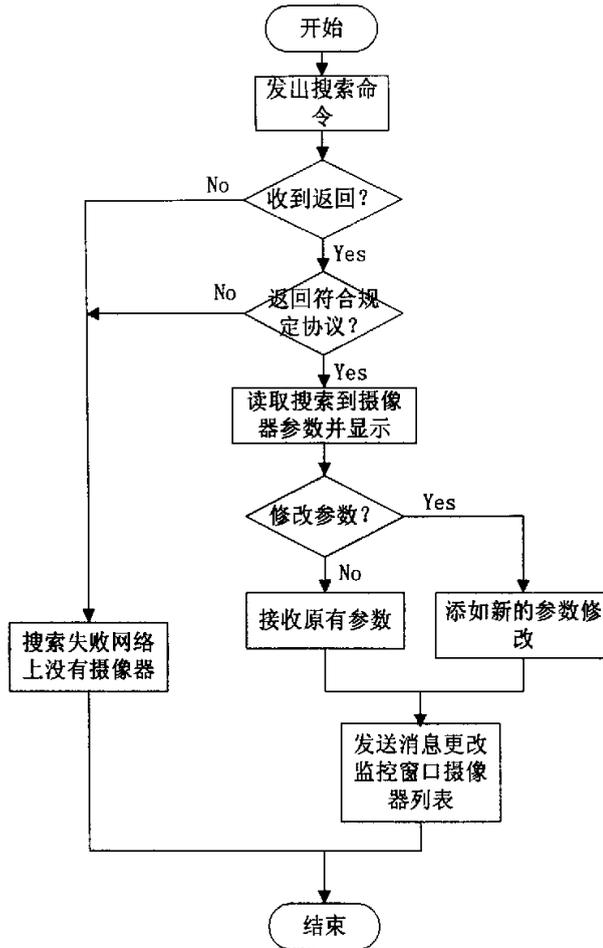


图 4.14 搜索 IpCamera 流程

Fig.4.14 The flowchart of search IpCamera

我们对用户的管理主要分管理员和一般用户两大部份。管理员拥有较高权限, 它可以进入图 4.12 这个界面进行用户的添加, 删除、属性的修改等等操作, 也可

以进入图 4.1 的监控界面进行监控操作。而普通用户的权限较低，他们只能进入图 4.1 的监控界面进行图像监控操作，并不能对用户进行操作。添加用户操作的界面，如图 4.15 所示。



图 4.15 用户添加管理界面

Fig.4.15 The manage interface for user added

第五章 性能评价

网络视频监控系统是基于 C/S 架构的专门适用于大型网络监控系统的企业级专业软件。它的前端采用的是 M-JPEG 压缩标准的网络视频服务器硬件，清晰度高。完全满足上百个监控点的监控，适用于大型企业和分布广泛的企业集团使用。

论文中提出的基于 IpCamera 监控系统的实现具有很大的优势。不同的用户可根据职务不同分配不同的控制权限。由于采用模块化设计方案，模块与模块之间关系很小，这样当系统扩充或局部做出修改时，只需扩充相应模块或升级相应模块即可，不用对整个程序的很多部分做出更改，也利于后来的开发者很快的熟悉本程序代码。比如说要修改通信模块现在采用的 UDP 协议为 TCP 协议，我们只需要把通信模块的相应部分修改即可，其它部分一点都不需要改动。可提高软件的更新换代得速度

5.1 测试环境

1. 硬件环境: 16 个 IpCamera, 一台 PC 机(IP 地址为 192.168.180.195), 10M 的以太网;
2. 软件环境: Windows2000 professional, 本论文设计的客户端软件。

5.2 测试内容

包括以下测试内容:

1. 几种格式下图像显示效果测试;
2. 监控系统稳定运行时间测试。

5.3 系统的功能评价

利用一台 PC 机, 在局域网中同时对 16 个 IpCamera 进行操作, 通过对这 16 个 IpCamera 的各种功能的操作, 我们可以肯定监控软件完全实现了初期设计的功能, 具体如下

1. 实时视频监控
 - 远程视频传输: 系统采用标准的 TCP/IP 协议, 可应用在局域网上, 授权用户监控远程现场;
 - 多画面监视: 1/4/9/16 彩色活动画面同屏显示, 单画面达 15 帧/秒;
 - IpCamera 分组功能: 强大的分组功能可将 IpCamera 根据监控环境不同编成不同工作组, 每个工作组可分配数个 IpCamera;
 - 用户管理功能: 提供安全完善的用户账号密码管理功能。

2. 录像

- 手动录像：监控终端用户可根据需要随时选择系统各个监控前端进行录像控制，图像数据保存在客户端本地的硬盘上；
- 录像的检索及回放：录像记录可以在所有监控终端上以具有检索权限的用户名登录后进行检索，录像检索可以根据不同的查询条件如日期、监控地点和报警类型检索录像记录；
- 录像管理：对录像硬盘空间、文件备份、文件删除、硬盘容量等进行统一管理；
- 手动抓图功能：可实现监控同时，将一些重点情况保存成文件，并通过工具观看。

5.4 系统的性能评价

2. 图像质量

图像质量是视频监视系统性能评价的核心问题，支持格式的对比，如表 5.1 所示。视频图像信号经压缩、存储、再经过解压缩，其转换的优劣及转换的速度，可以从图像的清晰度、灰度、色彩还原和实时性能等多个方面加以评价，其中清晰度是评价图像质量的最重要的指标。清晰度可以分为显示清晰度和回放清晰度，回放清晰度能表达 IpCamera 的录像和回放的总效果，清晰度评价的度量标准主要由图像的分辨率来决定，图像的分辨率是指一幅图像能分解成多少个像素所组成，国际标准是按其水平和垂直的像素点的乘积来表征的。

表 5.1 支持格式的对比

Table 5.1 The comparison of supported format

常见的图像格式	176*144(QCIF)	352*288 (CIF)	704*576(FCIF)	640*480 (VGA)
可视通 ^[14]	不支持	支持	支持	不支持
黄金眼 ^[15]	支持	支持	支持	支持
IpCamera	支持	支持	不支持	支持

从表 5.1 的同类产品对比中可以发现，我们的系统在图像分辨率上的水平较高，它支持从低到高大部分的分辨率格式，而且支持的高分辨 640×480 (VGA) 使我们的系统完全可以应用在高质量图像要求的环境中。

3. 压缩方式

压缩方式是视频监控的核心技术。压缩方式很大程度上决定着图像的质量、压缩比、传输效率和传输速度等性能，它是评价视频监控系统性能优劣的重要一环。随着多媒体技术的发展，相继推出了许多压缩编码标准，目前主要有 JPEG/M-JPEG、H.261/H.263 和 MPEG 等标准^[16-19]。

表 5.2 图像压缩的对比

Table 5.2 The comparison of image compression

常见的图像压缩方式	JPEG/M-JPEG	H.261/H.263	MPEG 1/2	MPEG 4
画面质量	最好	较差	一般	好
算法复杂性	简单	一般	一般	复杂
压缩率	较低	一般	较低	较高

我们选用的是表 5.2 中的 MJPEG 压缩格式来实现图像的压缩,它的优点可以从上面的对比中看出。我们采用这种压缩方式的考虑是这种压缩算法比较简单,相对容易在低端嵌入式模块中实现。并且现在互联网的发展趋势是带宽越来越大,网络环境变得更好。比较有发展前景。

4. 每秒处理图像的总帧数

IpCamera 监控客户端每秒处理图像的总帧数取决于 JPEG 图像解压速度和网络带宽,当要求视频监控点比较多时,必然会降低每路的图像分辨率或减少实时录像的路数,所以,监控客户端每秒处理图像的总帧数也是评价本系统性能优劣的一个方面。在基于 PC 机环境下运行的客户端中,图像处理速度又决定于 CPU 的处理速度、总线的传输速度和内存容量。在奔腾 2.4G CPU、192M 内存、VGA 图像格式、10M 局域网环境条件下测试参数如下(这个参数的获得是在程序中加一个定时器累计得来)。

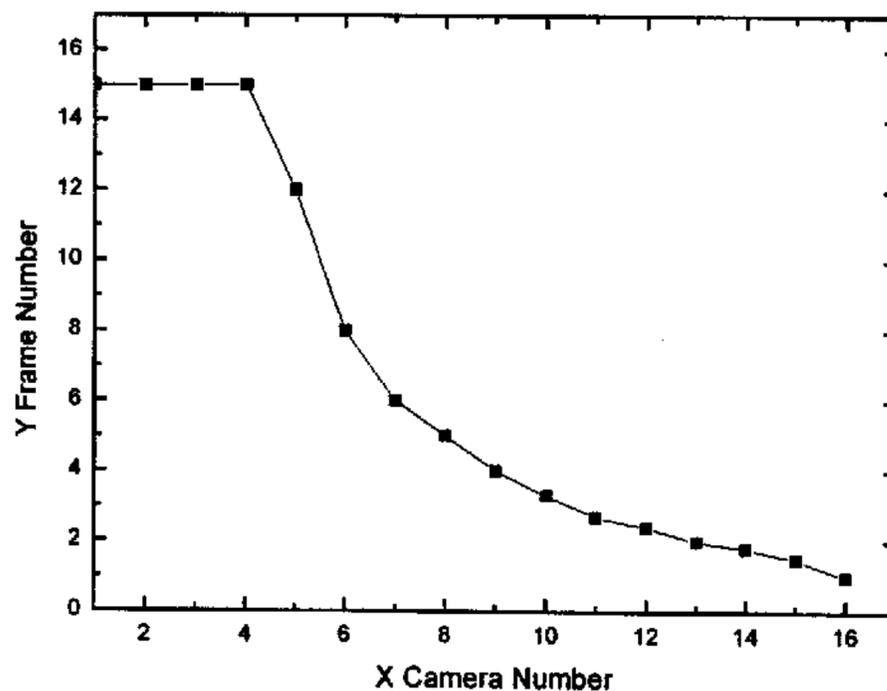


图 5.1 VGA 格式下播放个数与最大播放帧数的关系

Fig.5.1 Validations of the playing number in VGA format with maximum playing frame

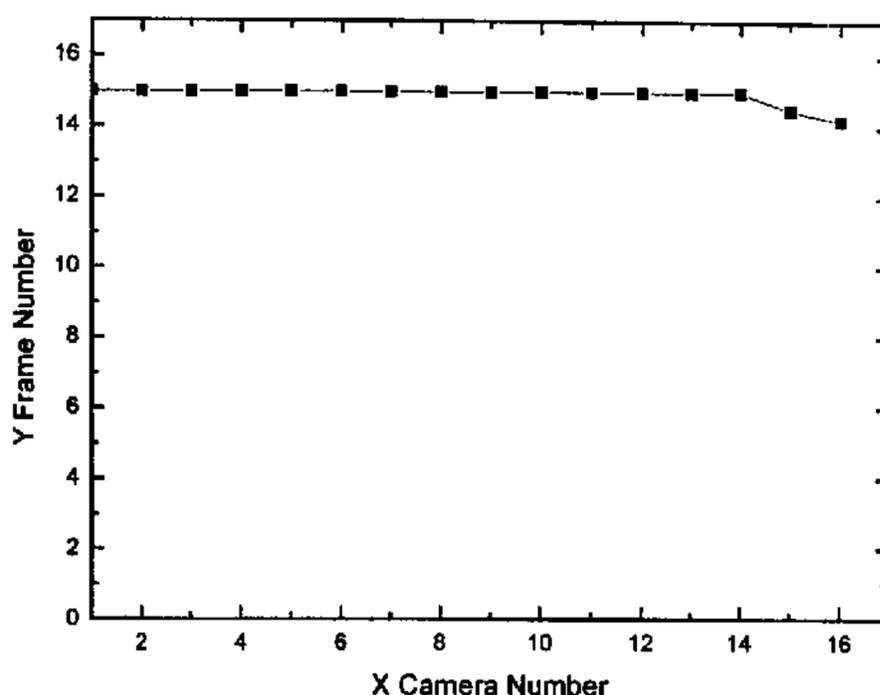


图 5.2 QCIF 格式下播放个数与最大播放帧数的关系

Fig.5.2 Validations of the playing number in QCIF format with maximum playing frame

图 5.1 中监控系统性能之所以下降这么大的主要原因是图像分辨率较大，导致图像数据过大，达到网络流量瓶颈。解决这个问题的办法有三个，第一个是增加网络带宽，采用 100M 网或 1000M 网；第二个是修改图像压缩算法；第三个是降低图像分辨率，减少每帧图像数据量。对比三个办法，我们采用了可以立即实现的第三种方法，即增加了分辨率相对较低的 QCIF 格式，在这种格式，其他测试环境不变的情况下，重新测试了该项参数如图 5.2 所示。可见监控窗口增多时每秒可以播放的帧数提高了很多。针对以上两种分辨率的对比分析，我们在较好的网络环境中或者同时监控路数较少的情况下可以采用 VGA 格式，否则可以采用 QCIF 格式来达到流畅的监控画面。

5. 稳定性

稳定性的测试是指在上面描述的环境中使系统不间断的运行，测试系统崩溃或出错的概率，并分析其出现的原因。对此进行了为期两周的测试，系统基本稳定，一周中只出现了一次较大 bug，对此已经做出改善。稳定工作对视频监控是至关重要的，也是当前视频监控最主要的问题。通过一段时间的测试以及通过对测试中出现的问题的分析解决，经过分析认为对于基于 PC 机的视频监控客户端，影响其稳定运行的因素主要有以下四种。

- 1) PC 机：兼容的 PC 机用于 24 小时以上不间断工作的性能不是很稳定，工控机相对兼容 PC 机的稳定性是一种档次上的提高，适用于较复杂的工作环境，对外部电/磁场的干扰有一定的抑制功能。
- 2) 操作平台：Windows98 操作系统的稳定性不高，容易死机。Windows 2000，

Linux 等操作系统较 Windows98 有更稳定的性能。

- 3) 软件设计：监控软件的设计也很重要，其能力上应支持多任务并发处理，如监控、录像、回放、备份、报警、控制、远程接入等的多工处理能力。
- 4) 图像解压缩：软件解压占用 CPU 资源非常大，使其它功能在运行时，CPU 来不及处理造成程序出错。

当然，仅仅两周的测试时间是不能说明问题的。目前对这项的测试还很不够，这主要是很难以提供长时间运行的环境，不过这一点我们可以在产品的应用中通过长时间得运行来继续发现、修改问题来提高它的稳定性。

第六章 总结与展望

本文针对网络化摄像机设计并实现了一个相应的监控软件，该软件不但具有远程图像的显示功能，而且还可以对其进行相应的控制。

首先，介绍了视频监控的发展历史，总结了早期的视频监控的缺点，有针对性的引入了嵌入式监控的概念，提出了设计该软件的必要性。

其次，结合面向对象编程的思想，将其功能进行了模块化的分析，分成功能相对独立的模块，并详细地介绍了每个模块的功能。

之后，在论文中详细描述了每个模块的实现过程，实现中出现的问题以及所采取的解决措施。

最后，对实现的系统进行了测试，并对测试结果进行了分析。

经过大量的测试和现场实践，证明该系统实现了以下的功能：

1. 多画面监视，1/4/9/16 彩色活动画面的同屏显示，单画面达 15 帧/秒；
2. 摄像机分组管理，可将摄像机根据监控环境的不同进行分组，从而进行分类的管理；
3. 用户管理功能；
4. 单帧或连续帧录制功能。

本系统是以解放军总参研究院视频监控项目为背景而设计和实现的，在功能上主要考虑到实际需求。由于该系统是配合安防监控系统一起联合工作的，只是在发生报警时才启动图像录制功能，所以在该系统采用了压缩率相对较小的 MJPEG 压缩格式，这种算法的优点是实现简单、图像清晰，但并不利于大量图像存储。

随着普适计算、嵌入式 Internet 技术和新的高效的压缩算法的不断进步，在不久的将来，我们可以在任何时间、任何地点利用身边存在的小巧的数字交互设备通过无线的方式视频监控现场，使视频监控走向简单化，大众化。

参考文献

1. 盛凌志, 柴乔林, 丁长青. 基于 Internet 的城市楼宇安全监控系统设计与实现[J], 计算机应用研究, 2002, 19 (9): 100-103.
2. 赵海. 嵌入式 Internet-21 世纪的一场信息技术革命[M], 北京: 清华大学出版社, 2001.9, 160-165.
3. 蔡敬猷. 监控系统的发展[EB/OL], <http://www.shmen.com/jiankzx/jiankzx.htm>, 2003.
4. 慧聪安防商务网, 视频监控系统的视频发展趋势 [EB/OL]<http://www.security.sinobnet.com/zl/shipinjk/>, 2002.
5. 金欢. 基于嵌入式 Internet 技术的 Webit 体系结构研究与实现[D], 沈阳: 东北大学, 2002.
6. 熊江. 嵌入式系统与普适计算[EB/OL], <http://bbs.neuera.com/viewtopic.php?t=32>, 2003.
7. Grady Booch, James Rumbaugh, Ivar Jacobson 著. UML 用户指南[M], 北京: 机械工业出版社[M], 2001, 3-30.
8. 朱志刚. 数字图像处理[EB/OL], <http://www-scf.usc.edu/~flv/ipbook/chap09.htm>, 2001.
9. 林福宗. 多媒体技术基础及应用[M], 北京: 清华大学出版社, 2000, 100-150.
10. L. Berc, W. Fenner, R. Frederick, S. McCanne, P. Stewart. RTP Payload Format for JPEG-compressed Video, RFC2035, Network Working Group, October 1996.
11. L. Berc, W. Fenner, R. Frederick, S. McCanne, P. Stewart. RTP Payload Format for JPEG-compressed Video, RFC2435, Network Working Group, October 1998.
12. 钱能. C++ 程序设计教程[M], 北京: 清华大学出版社, 1999, 290-311.
13. 徐新华. C++ Builder 编程指南[M], 北京: 人民邮电出版社, 163-228.
14. 北京可视通电子技术有限责任公司, 公司产品介绍 [EB/OL], <http://www.videocomm.com.cn/product/product2.htm>, 2003.
15. 黄金眼科技有限公司, 公司产品介绍[EB/OL] <http://www.goldeneye.net.cn/production/>, 2003.
16. T. Turlitti, C. Huitema. RTP Payload Format for H.261 Video Streams, RFC2032, Network Working Group, October 1996.
17. M. Civanlar, G. Cash, B. Haskell. RTP Payload Format for Bundled MPEG, RFC 2343, Network Working Group, May 1998.
18. D. Hoffman, G. Fernando, V. Goyal, M. Civanlar. RTP Payload Format for MPEG1/MPEG2 Video, RFC 2250, Network Working Group, January 1998.
19. Y. Kikuchi, T. Nomura, S. Fukunaga, Y. Matsui, H. Kimata. RTP Payload Format for MPEG-4

-
- Audio/Visual Streams, RFC 3016, Network Working Group, November 2000.
20. 徐光佑, 史元春, 谢伟凯. 普适计算[J], 计算机学报, 2003, 26 (9): 1042-1045.
21. Weiser M. The computer for the twenty-first century[J]. Scientific American, 1991, 265 (3): 94-104.
22. Dertouzos M. The future of computing[J]. Scientific American, 1999, 282 (3): 52-63.

致 谢

首先，非常感谢我的导师赵海教授。在论文的选题、材料组织和课题研究等方面赵老师一直给予我细致耐心的指导。在跟随赵老师攻读硕士学位的两年中，他渊博的学术知识，深奥的哲学思想，丰富的思维方式使我在专业知识方面受益匪浅，而他那独有的人格魅力，稳重潇洒的处事作风，平易近人的性格，使我对人生有了更深一层的理解。我很高兴能够在赵老师的指导下完成硕士学业。

其次，我要感谢韩光洁博士、王济勇博士、林涛博士、王金东博士，张文波博士、关沫博士、王晓英博士的悉心指导。这其中尤其要感谢韩光洁博士，在论文的写作过程中他给了我很多指导性的意见，并仔细阅读了论文的全文，对于我的帮助尤多。另外所有的博士一起利用大量的时间探讨我的论文，给出了许多建设性的意见，使我对于论文的思路和结构有更深刻的认识。

在论文的整理、资料的收集、图文的处理等方面，得到了王鑫硕士、罗玎玎硕士、尹震宇硕士、阮冠春硕士、梁琼硕士、谭柳萍硕士、刘琦硕士和黄丽雯硕士等人的积极帮助，占用了他们许多宝贵时间，在此也向他们表示由衷的感谢。

在论文即将付印之时，对以上导师、同学给予我的真诚的帮助再次表示衷心的感谢。

论文相关工作

在攻读硕士课题期间，围绕课题的研究和开发工作本人完成了如下的相关工作：

1. Webit 网络部分 TCP 协议改进；
2. 辽阳市电业局安防系统 SNMP Trap 包解析，与中兴电源通信，上层软件后台函数编写；
3. 嵌入式实时操作系统内核的研究，参研人员；
4. Webit 拨号上网方式的设计与实现，参研人员。