** **

**Southwest university of science and technology**

本科毕业设计（论文）

门禁监控视频软件设计与实现

|  |  |
| --- | --- |
| 学院名称 | 计算机科学与技术学院 |
| 专业名称 | 软件工程 |
| 学生姓名 | 张浩 |
| 学号 | 20111689 |
| 指导教师 | 刘畅 |

二〇-五年六月

门禁监控视频软件设计与实现

摘要：随着社会的发展和技术的不断进步，基于视频的监控和处理技术已经从模拟化、数字化、网络化发展到了智能化，监控方式也从纯人工发展到了半人工以及部分应用的智能化，如人头、人脸等目标的自动检测。人头检测作为人工智能和模式识别领域的一个重要内容。相关技术和理论越来越受到人们的关注，成为该领域的一个研究热点[7]。

就目前实际情况而言,智能视频监控不论是在理论方面还是在实际应用领域都存在诸多难题,国内外众多学者和有志之士致力于该领域的研究,并取得了不少卓有成效的突破。本文在分析和研究这些研究成果的基础上,对视频监控系统中存在的一些关键技术进行分析并实现。其中主要涉及对运动目标检测、提取和对运动目标跟踪等。

本文首先论述视频监控系统开发涉及的功能需求分析、总体技术方案、系统实现等,然后针对固定摄像机的视频监控系统,利用一种改进的快速模板匹配算法,实现运动目标检测,实现运动目标跟踪。

关键词：视频监控； 人头检测； 模板匹配

**Access control monitoring of video software Design and Implementation**

**Abstract**：With the continuous development of society and technology, video-based monitoring and processing technology has evolved from analog, digital, networked to the intelligence, surveillance mode from purely human development to the semi-artificial and intelligent part of the application, such as head, face, such as automatic detection of targets. Head detection as artificial intelligence and pattern recognition is an important content. Related technology and theory more and more people's attention and become a hot topic in the field.

On the current situation, there are many problems in theory and in practical applications with the intelligent video surveillance, many scholars and ambitious people working on the field, and made a lot of fruitful breakthroughs. Based on the analysis and research on the results of these studies, some of the key technologies of video surveillance systems that exist in the analysis and implementation. Which primarily relates to moving target detection, extraction and moving target tracking.

This paper discusses the development of a video surveillance system involving functional requirements analysis, technical solutions, system implementation, and fixed cameras for video surveillance systems, the use of a modified fast template matching algorithm, moving target detection, tracking moving objects.

**Key words:** Video Monitor, Head Detection, Template Matching

目 录

[第1章 绪 论 1](#_Toc420251495)

[1.1 选题背景和意义 1](#_Toc420251496)

[1.2 视频监控研究现状 1](#_Toc420251497)

[1.3 本文研究内容和章节安排 2](#_Toc420251498)

[第2章 视频监控系统中的关键技术 4](#_Toc420251499)

[2.1图像处理技术 4](#_Toc420251500)

[2.1.1 OpenCV的特征 4](#_Toc420251501)

[2.1.2 OpenCV的功能 4](#_Toc420251502)

[2.1.3 OpenCV 模块 5](#_Toc420251503)

[2.2 视频检测技术-模板匹配 5](#_Toc420251504)

[2.2.1 实现原理 5](#_Toc420251505)

[2.2.2 OpenCV中支持的匹配算法 7](#_Toc420251506)

[2.2.2 人头检测算法 8](#_Toc420251507)

[2.3 MFC 10](#_Toc420251508)

[2.4 TinyXML 12](#_Toc420251509)

[2.5 本章小结 13](#_Toc420251510)

[第3章 视频监控系统整体功能介绍 14](#_Toc420251511)

[3.1 视频监控系统的整体结构 14](#_Toc420251512)

[3.1.1 系统框架 14](#_Toc420251513)

[3.1.2 系统组成 15](#_Toc420251514)

[3.2 系统的功能及要求 16](#_Toc420251515)

[3.3 系统开发环境介绍 16](#_Toc420251516)

[3.3.1 操作平台 16](#_Toc420251517)

[3.3.2 开发环境 17](#_Toc420251518)

[3.4本章小结 17](#_Toc420251519)

[第4章 视频监控系统软件实现 19](#_Toc420251520)

[4.1 系统设计原则 19](#_Toc420251521)

[4.2 功能需求 19](#_Toc420251522)

[4.3 基于MFC的用户界面实现 21](#_Toc420251523)

[4.4 基于OpenCV的图像处理实现 30](#_Toc420251524)

[4.4.1 人头检测的实现 30](#_Toc420251525)

[4.4.2 目标跟踪 31](#_Toc420251526)

[4.5 本章小结 32](#_Toc420251527)

[第5章 测试、总结与展望 33](#_Toc420251528)

[5.1 测试 33](#_Toc420251529)

[5.2 总结 34](#_Toc420251530)

[5.3 展望 34](#_Toc420251531)

[参考文献 36](#_Toc420251532)

[致谢 38](#_Toc420251533)

第1章 绪 论

1.1 选题背景和意义

随着视频监控技术的迅速发展与设备成本的不断下降，视频监控已经遍布于城市的各个角落成为了一种重要的视频信息来源。目前，视频监控系统在银行的安全防范、道路的交通管理、公安刑事侦查等特殊领域已经发挥了及其重要的重用。由于上述领域对于实用的监控视频采集、处理与分析技术和应用表现了十分迫切的需求，各类监控视频信息处理相关的基础理论研究与应用系统的设计开发也越来越受到政府、科研院以及公司的重视。

目前使用的监控系统系统大多只具有录像功能，因此只能提供事后查询，并不能对当时情况作出及时判断并提出警告，例如门禁监控，传统的门禁系统往往将门禁前的视频图像发到视频墙上，由人工监视什么时间人员可以进出，进出是否刷卡，是否有异常情况发生。所以有很多不足产生：

1. 工作强度大，持续时间长，容易对负责浏览视频的工作人员的视觉会造成很大的影响，危害身体健康。
2. 监控成本上，人力成本的持续投入使得成本不断增加。
3. 长时间高强度、高压力的工作状态下，负责视频浏览的工作人员注意力难以集中，容易出现一些重要信息的遗漏和误判现象。
4. 门禁系统刷卡与视频无关联，在突发情况下，监控人员很难判断出发生的原因。

为了解决现阶段门禁的不足。本文设计出基于门禁的视频监控软件系统，在本系统中实现了门禁硬件与视频监控的交互。减轻监控人员的负担，记录异常情况的发生和截图。

1.2 视频监控研究现状

目前美国以及欧洲地区等西方先进国家仍然掌握着大部分智能视频监控系统的核心算法技术,在国际市场上的智能视频监控系统中占据很大的优势。

美国国防部高级研究项目署（Defense advanced research projects agency , DARPA）于1997年建立了视觉监控重大项目VSAM(Visual surveillance and monitoring), 该项目对用于民用场景和战场中视频监控的自动理解技术进行了重点研究。马里兰大学研发成功的实时视频监控系统W4主要任务是是被监控场景中的行人，将形状分析和目标跟踪技术相结合，对人的外表进行模型构建。高级研究计划VACE(Video Analysis and Content Extraction)于2003年由美国ARDA机构开始主持。在对车辆和行人的跟踪及其交互作用识别的相关研究方面上，英国雷丁大学（University of Reading）取得了一系列的研究成果。以色列也在智能视频监控领域取得了很大的研究成果，他们的产品主要用在边界不含以及商业调查分析领域[10-11]。

在对智能视觉监控领域的研究方面，国内起步比较晚，目前，中国科学院自动化研究所模式识别国家重点实验室在国内的研究机构中处于领先地位，取得了一系列研究成果。除此之外，国内还有一些高校也在这一领域投入了相当的研究精力，如上海交通大学、清华大学、华中科技大学、北京航空航天大学、北京理工大学等[3]。

1.3 本文研究内容和章节安排

本文将监控视频作为研究对象，基于监控视频本身的特点，对监控视频摘要技术进行了研究。在门禁视频监控中，分析人进出门禁的情况，并进行统计某一时刻进出人员数量。

基于以上工作，将本文安排为五章，具体如下:

1. 绪论

简单介绍视频监控系统的研究背景，意义以及视频监控系统的现状和发展趋势，总结前人的基础上，结合项目实际应用阐述本论文的主要工作。

1. 视频监控中的关键技术

本章主要介绍了视频监控系统中应用到的一些关键技术，例如Opencv图像处理、模块识别技术、目标跟踪[12-13]等。

1. 视频监控系统软件总体分析与设计

针对现有视频监控的缺点，根据项目需求，本章介绍了系统的总体设计方案，确定了系统的组成结构以及系统体系结构。最后针对该项目的具体情况，简单介绍了软件开发环境。

1. 视频监控系统软件实现

本章介绍了系统软件设计。由于本项目中所有功能都是通过软件界面操作来实现了，故从总体设计到详细的功能模块介绍，最终实现设计要求。

1. 测试、总结与展望

对系统进行测试，对本项目的设计进行了总结，并提出了不足之处以及进一步的研究工作。

论文各章节的主要内容及相互关系如图1-1所示。



图1-1

第2章 视频监控系统中的关键技术

2.1图像处理技术

OpenCV[4]是由Intel公司资助的开源计算机视觉库，它由一系列C函数和少年C++类构成，实现图像处理和计算机视觉方面的很多通用算法。

OpenCV包括300多个C/C++函数的跨平台的中、高层API。它不依赖于其他的外部库，但是也可以使用某些外部库。

OpenCV对商业应用和非商业应用都是免费的。另外，OpenCV也为Intel公司的Integrated Perfirmance Promitives(IPP)提供了透明接口，这意味着如果有特定处理器(当然是Intel的处理器)优化的IPP，OpenCV将在运行时自动加载这些库。

2.1.1 OpenCV的特征

Opencv具有以下特征：

1. 开源计算机视觉库采用C/C++编写
2. 使用目的是开发实时应用程序
3. 独立于操作系统，硬件结图形管理器
4. 具有通用的图像/视频载入、保存和获取模块
5. 具有底层和高层的应用开发包。

2.1.2 OpenCV的功能

应用OpenCV能够实现以下功能：

1. 对图像数据的操作，包括分配、释放、复制、设置和转化数据。
2. 对图像和视频的输入输出，指文件和摄像头作为输入，图像和视频文件作为输出。
3. 具有对矩阵和向量的操作以及线性代数的算法程序，包括矩阵积、解方程、特征值以及奇异值等。
4. 可对各种动态数据结构，如列表、队列、集合、树和图进行操作。
5. 具有基本的数字图像处理能力，如可进行滤波、边缘检测、角点检测、采样与差值、色彩转换、形态操作、直方图和图像金字塔等操作。
6. 可对各种结构进行分析，包括连接部件分析、轮廓处理、距离变换、各种矩的计算、模板匹配、Hough变换、多变形逼近、直线拟合、椭圆拟合和Delaunay三角划分等。
7. 对摄像头的定标，包括发现与跟踪定标模式、定标、基本矩阵估计、齐次矩阵估计和立体对应。
8. 对运动的分析，如对光流、运动份额和跟踪的分析。
9. 对目标的识别，可采用特征法和隐马尔可夫模型（HMM）法。
10. 具有基本的GUI功能，包括图像与视频显示、键盘和鼠标事件处理及滚动条等。
11. 可对图像进行标注，如对线、二次曲线和多表型进行标注，还可以书写文字。

2.1.3 OpenCV 模块

OpenCV包括以下几个模块，其具体功能是:

1. CV 主要的OpenCV函数
2. CVAUX 辅助的OpenCV函数
3. CXCORE 数据结构与线性代数支持
4. HIGHGUI 图像界面函数
5. ML 机器学习，包括模式分类和回归分析等
6. CVCAM 负责读取摄像头数据的模块。

2.2 视频检测技术-模板匹配

模板匹配[6]是一项在一幅图像中寻找与另一幅模板图像最匹配(相似)部分的技术.

2.2.1 实现原理

我们需要2幅图像

a 原图像(I)：在这幅图像里面，能找到一块和模板匹配的区域

b 模板(T)：将和原图像比照的图像块

检测最匹配的区域:

 ＋  ＝ 

为了确定匹配区域，我们不得不滑动模块图像进行比较:



通过滑动，即将图像一次一个像素(从左往右,从上往下). 在每一个位置, 都进行一次度量计算来表明它是 “好” 或 “坏” 地与那个位置匹配 (或者说块图像和原图像的特定区域有多么相似).

对于T覆盖在**I**上的每个位置,你把度量值保存到结果图像矩阵**(R)**中。在**R**中的每个位置（x,y）都包含匹配度量值:



上图就是**TM\_CCORR\_NORMED**方法处理后的结果图像**R**。最白的位置代表最高的匹配。红色椭圆框住的位置很可能是结果图像矩阵中的最大数值, 所以这个区域 (以这个点为顶点,长宽和模板图像一样大小的矩阵) 被认为是匹配的。

2.2.2 OpenCV中支持的匹配算法

1. 平方差匹配 method=CV\_TM\_SQDIFF

这类方法利用平方差来进行匹配,最好匹配为0.匹配越差,匹配值越大.

R(x,y)= \sum _{x',y'} (T(x',y')-I(x+x',y+y'))^2

1. 标准平方差匹配 method=CV\_TM\_SQDIFF\_NORMED

R(x,y)= \frac{\sum_{x',y'} (T(x',y')-I(x+x',y+y'))^2}{\sqrt{\sum_{x',y'}T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}}

1. 相关匹配 method=CV\_TM\_CCORR

这类方法采用模板和图像间的乘法操作,所以较大的数表示匹配程度较高,0标识最坏的匹配效果.

R(x,y)= \sum _{x',y'} (T(x',y')  \cdot I(x+x',y+y'))

1. 标准相关匹配 method=CV\_TM\_CCORR\_NORMED

R(x,y)= \frac{\sum_{x',y'} (T(x',y') \cdot I'(x+x',y+y'))}{\sqrt{\sum_{x',y'}T(x',y')^2 \cdot \sum_{x',y'} I(x+x',y+y')^2}}

1. 相关匹配 method=CV\_TM\_CCOEFF

这类方法将模版对其均值的相对值与图像对其均值的相关值进行匹配,1表示完美匹配,-1表示糟糕的匹配,0表示没有任何相关性(随机序列).

R(x,y)= \sum _{x',y'} (T'(x',y')  \cdot I(x+x',y+y'))

在这里

\begin{array}{l} T'(x',y')=T(x',y') - 1/(w  \cdot h)  \cdot \sum _{x'',y''} T(x'',y'') \\ I'(x+x',y+y')=I(x+x',y+y') - 1/(w  \cdot h)  \cdot \sum _{x'',y''} I(x+x'',y+y'') \end{array}

1. 标准相关匹配 method=CV\_TM\_CCOEFF\_NORMED

R(x,y)= \frac{ \sum_{x',y'} (T'(x',y') \cdot I'(x+x',y+y')) }{ \sqrt{\sum_{x',y'}T'(x',y')^2 \cdot \sum_{x',y'} I'(x+x',y+y')^2} }

2.2.2 人头检测算法

人头检测是一个具有较高的理论和实际应用价值的研究课题，随着研究的不断深入，各种新的检测方法也不断出现。但总体来讲主要分为两大类，即如前所述的基于知识的方法和基于统计学习的方法。基于知识的人头检测方法通过一定的先验知识规则(如颜色、肤色、形状等)来描述人头，并根据这些规则判断一个区域内是否包含人头。模板匹配是一种典型的基于知识的人头检测方法，该方法通过模板对人头进行描述，人头模板包括颜色、形状及其组合等多种形式。

AdaBoost(Adaptive Boost)算法由 Freund 等人于 1996 年提出，算法的最初思想是基于弱学习模型，该模型假设存在一种弱学习算法，其获得的分类器的分类能力只比随机分类稍好(对于两类分类问题，其分类能力略高于 50%)。利用该弱学习算法经过训练获得若干个弱分类器，通过对弱分类器进行组合可以获得任意高精度的分类器。

人头检测可视为一个两类分类问题，其中一类为人头(正样本)，另一类为非人头(负样本)，假设 AdaBoost 学习算法的输入为训练样本集

LS = {(xi,yi),i=1,2,3, ⋅⋅⋅,N }

其中，xi表示第i 个训练样本，yi 表示样本类别，且 yi∈(+1, − 1)， +1 表示正样本， −1 表示负样本。因此，对于每个样本，在训练过程中都事先知道其样本类型，以便对算法做出的预测正确与否进行判断。

在算法训练前，需要建立用以描述待检测目标物的特征库，训练过程就是从这些特征库中挑选最能区分正负样本的特征。

在算法训练阶段，首先对每个样本赋予相同的初始权重W ,之后进行K 轮训练，并且在每轮训练结束后根据当前分类结果调整训练样本的权重，减小正确分类样本的权重，同时增加错误分类样本的权重，从而使每个弱分类器主要围绕难以正确分类的样本进行学习。经过K 轮训练得到一个由K 个弱分类器(每个弱分类器对应一个特征)组成的弱分类器序列

h ={(h1, α1), (h2, α2), (h3, α4), ⋅⋅⋅ , (hk, αk)}

其中，hi 表示第 i 个弱分类器，αi 为弱分类器的加权系数。最后对K 个弱分类器进行组合，得到最终的强分类器H 。虽然每个弱分类器hi的分类预测能力只比随机稍好，但通过对K 个弱分类器进行组合，能够使强分类器H 获得任意高的精度。

AdaBoost 算法流程如图 2.2.1 所示。算法主要由离线训练和在线检测两部分组成，在训练阶段首先对正负样本进行训练，从特征库中选取最优特征并组合成强分类器，在检测阶段，用训练好的分类器对各个待检测窗口进行检测。

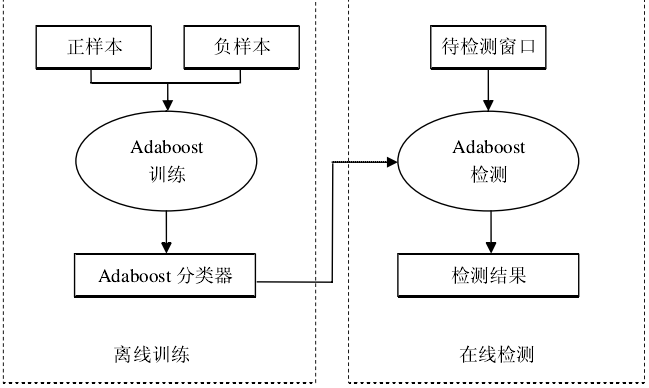


图2.2.1

一般而言，AdaBoost 算法对以下两种情况能获得较好的效果：一是样本具有多样性，即各训练样本的分类难度互不相同，算法通过训练不断对样本权重的分布进行调整，使学习过程更加关注分类难度较大的样本；二是算法对训练样本集的改变较敏感，由不同训练样本集产生的分类器存在较大差异，通过在训练过程中不断更新训练样本，能够使最终的分类器更趋近于真实分类面，减少算法的分类误差。

2.3 MFC

MFC,[微软](http://baike.baidu.com/view/2353.htm)基础类(Microsoft Foundation Classes)[5]， 同VCL类似，是一种Application Framework，随微软[Visual C++](http://baike.baidu.com/view/100377.htm)开发工具发布。目前最新版本为9.0（截止2008年11月）。该类库提供一组通用的可重用的类库供开发人员使用。大部分类均从CObject 直接或间接派生，只有少部分类例外。

MFC应用程序的总体结构通常由开发人员从MFC类派生的几个类和一个CWinApp类[对象](http://baike.baidu.com/view/2387.htm)（应用程序对象）组成。MFC 提供了MFC AppWizard 自动生成框架。 　[Windows](http://baike.baidu.com/view/4821.htm)应用程序中，MFC 的主包含文件为"Afxwin.h"。此外MFC的部分类为MFC/ATL 通用，可以在Win32 应用程序中单独包含并使用这些类。

作为Application Framework，MFC的使用只能提高某些情况下的开发效率，只起到辅助作用，而不能替代整个Win32 程序设计。

在本系统开发中，我们将选用MFC中的对话框进行开发。然而对话框又分为两种模式(模式对话框和非模式对话框)

1. 模式对话框

一个模式对话框是一个有系统菜单、标题栏、边线等的弹出式窗口。在创建对话框时制定WS\_POPUP,WS\_SYSMENU,WS\_CAPTION和DS\_MODALFRAME风格。及时没有制定WS\_VISIBLE风格，模式对话框也会被显示。创建对话框窗口时，将发送WM\_INITDIALOG消息给对话框过程。

对话框窗口被创建之后，Windows使得它成为一个激活的窗口，它保持激活直到对话框过程调用::EndDialog函数结束对话框的运行或者Windows激活另一个应用程序为止，在激活时，用户或者应用程序不可以激活它的所属窗口（Owner window）。从某个窗口创建一个模式对话框时，Windows自动地禁止使用（Disable）这个窗口和它的所有子窗口，直到该模式对话框被关闭和销毁。虽然对话框过程可以Enable所属窗口，但是这样做就失去了模式对话框的作用，所以不鼓励这样做。

Windows创建模式对话框时，给当前捕获鼠标输入的窗口（如果有的话）发送消息WM\_CANCLEMODE。收到该消息后，应用程序应该终止鼠标捕获(Release the mouse capture)以便于用户能把鼠标移到模式对话框；否则由于Owner窗口被禁止，程序将失去鼠标输入。

为了处理模式对话框的消息，Windows开始对话框自身的消息循环，暂时控制整个应用程序的消息队列。如果Windows收到一个非对话框消息时，则它把消息派发给适当的窗口处理；如果收到了WM\_QUIT消息，则把该消息放回应用程序的消息队列里，这样应用程序的主消息循环最终能处理这个消息。

当应用程序的消息队列为空时，Windows发送WM\_ENTERIDLE消息给Owner窗口。在对话框运行时，程序可以使用这个消息进行后台处理，当然应该注意经常让出控制给模式对话框，以便它能接收用户输入。如果不希望模式对话框发送 WM\_ENTERIDlE消息，则在创建模式对话框时指定DS\_NOIDLEMSG风格。

一个应用程序通过调用::EndDialog函数来销毁一个模式对话框。一般情况下，当用户从系统菜单里选择了关闭（Close）命令或者按下了确认（OK）或取消（CANCLE）按钮，::EndDialog被对话框过程所调用。调用::EndDialog时，指定其参数nResult的值，Windows将在销毁对话框窗口后返回这个值，一般，程序通过返回值判断对话框窗口是否完成了任务或者被用户取消。

1. 非模式对话框

一个无模式对话框是一个有系统菜单、标题栏、边线等的弹出式窗口。在创建对话框模板时指定WS\_POPUP、WS\_CAPTION、WS\_BORDER和WS\_SYSMENU风格。如果没有指定WS\_VISIBLE风格，无模式对话框不会自动地显示出来。一个无模式对话框既不会禁止所属窗口，也不会给它发送消息（WM\_ENTERIDlE）。当创建一个无模式对话框时，Windows使它成为活动窗口，但用户或者程序可以随时改变和设置活动窗口。如果对话框失去激活，那么即使所属窗口是活动的，在Z轴顺序上，它仍然在所属窗口之上。

应用程序负责获取和派发输入消息给对话框。大部分应用程序使用主消息循环来处理，但是为了用户可以使用键盘在控制窗口之间移动或者选择控制窗口，应用程序应该调用::IsDialogMessage函数。这里，顺便解释::IsDialogMessage函数。虽然该函数是为无模式对话框设计的，但是任何包含了控制子窗口的窗口都可以调用它，用来实现类似于对话框的键盘选择操作。当::IsDialogMessage 处理一个消息时，它检查键盘消息并把它们转换成相应对话框的选择命令。例如，当Tab 键被压下时，下一个或下一组控制被选中，当Down Arrow键按下后，一组控制中的下一个控制被选择。::IsDialogMessage完成了所有必要的消息转换和消息派发，所以该函数处理的消息一定不要传递给TranslateMessage和DispatchMessage处理。一个无模式对话框不能像模式对话框那样返回一个值给应用程序。但是对话框过程可以使用::SendMessage给所属窗口传递信息。

在应用程序结束之前，它必须销毁所有的无模式对话框。使用DestroyWindow销毁一个无模式对话框，不是使用::EndDiaLog。一般来说，对话框过程响应用户输入，如用户选择了“取消”按钮，则调用::DestroyWindow；如果用户没有有关动作，则应用程序必须调用::DestroyWindow。

在本系统开发过程中，首先使用非模式创建窗体，即系统的交互界面，然后使用模式对话框实现视频源的添加、删除以及一些消息的提醒。采用三层架构及显示层、业务层、数据层完成系统的架构。

2.4 TinyXML

现在越来越多的数据和配置采用了xml格式来存放和进行传输解析了，在本系统中需要使用xml文件来存放视频源信息。然而在C++方面，没有本地支持的库，所以需要寻找技术实现。微软的msxml这种重量级的xml解析器显得有些大材小用，尤其使用msxml解析xml设计到的复杂的com类型转换，让人感觉到繁琐。对于简单的xml文件解析，选择开源的TinyXml比较适合。

TinyXML是目前非常流行的一款基于DOM模型的XML解析器，简单易用且小巧玲珑，非常适合存储简单数据，配置文件，对象序列化等数据量不是很大的操作，支持对XML的读取和修改，能够用于c++,能够在Windows或Linux中编译。

DOM模型及文档对象模型，是将整个文档分成多个元素(如书、章、节、段等)，并利用树型结构表示这些元素之间的顺序关系以及嵌套包含关系。如下是一个XML片段:

 <Persons>  
None    <Person ID="1">  
None       <name>周星星</name>  
None       <age>20</age>  
None    </Person>  
None    <Person ID="2">  
None       <name>白晶晶</name>  
None       <age>18</age>  
None    </Person>  
None  </Persons>

2.5 本章小结

门禁视频监控的实现是建立在许多技术基础之上，本章主要介绍一些关键技术，例如模块匹配，OpenCV，MFC等等。

第3章 视频监控系统整体功能介绍

随着数字化、网络化的步伐逐步加快，现在社会在监控领域的需求也日益增强。视频监控系统本身也由传统的安防监控逐步深入到各行各业。在功能、性能、可靠性、结构方式等方面也发生了很大变化，这种方式具有低成本、结果简单的优点。但其中也有很多的缺点，比如：

1. 对现场场景无法进行分析判断，现在的视频监控系统基本上无法和“智能”联系起来，通常只是靠人力来实现‘监’与‘控’，也只是简单的将监控信息记录下来，以供事后查验。这样的话数据存储量庞大，如发生异常情况时，工作人员需要对大量的监控信息进行检索，从而花费巨大的时间和精力
2. 工作强度大，持续时间长，容易对负责浏览视频的工作人员的视觉会造成很大的影响，危害身体健康。
3. 监控成本上，人力成本的持续投入使得成本不断增加。
4. 长时间高强度、高压力的工作状态下，负责视频浏览的工作人员注意力难以集中，容易出现一些重要信息的遗漏和误判现象。
5. 门禁系统刷卡与视频无关联，在突发情况下，监控人员很难判断出发生的原因。

我们了解了目前视频监控系统所处的环境及面临的问题，也分析了本系统中采用的一些关键技术后，进入设计阶段。本章是本项目监控系统的整体介绍，将从系统框架，系统组成、系统体系结构以及系统的功能要求进行阐述。最后，介绍了实现系统的开发环境，为后续系统的开发做好铺垫。

3.1 视频监控系统的整体结构

3.1.1 系统框架

本系统利用最新的计算机处理技术，将门禁视频监控前端采集到的视频信息显示在监控系统中，经过系统一系列复杂的算法处理后进行门禁实时监控。协调各个模块进行智能分析监控画面，达到能够事先预警,对不安全因素采取及时报警,能够有效的遏制异常及恶性事件的发生系统的结构拓补图如图所示。



图3.1.1

3.1.2 系统组成

系统是由一个可执行的应用程序组成。系统由以下功能模块组成：(1)视频播放模块、(2)视频显示模块、(3)视频检测模块、(4)视频信息输出模块完成系统的播放，识别，跟踪，处理，存储，报警等功能，构成一个完成的视频监控系统，系统框架图如图3.1.2所示。



图3.1.2

1. 视频播放模块

负责存储选择过的视频源，以及播放选择的视频源

1. 视频显示模块

将选择视频源中的视频图像进行显示，也可以显示检测的视频图像

1. 视频检测模块

负责对所播放的视频进行人头检测，以及进行跟踪检测，并判断检测出的人是否进出门禁。

本模块主要利用视频分析技术，打破以往完全依靠人力来完成的监控工作，实现自动分析，判断时间的发生，对异常行为检测，并将识别出的行为时间存入数据库并作报警。

1. 视频信息输出模块

负责输出视频的视频名称，视频路径、监控人员对系统的一些操作信息显示以及开启检测功能后显示检测图像的检测数据。供监控人员进行查阅以及验证。

3.2 系统的功能及要求

本系统总的设计思想是充分考虑现场实际情况下，兼顾先进性、可靠性、安全性、经济型、可扩充性、可维护性和规范性做到一切应从实际出发，使监控系统具有较高的实用效能。这也是安防产品，智能化建筑在当今之所以能迅速兴起并发展的关键所在。

该系统具有如下功能及要求：

1. 视频文件需要有较高的清晰度，人工可分辨出视频中的人和物体。从而提高检测的准确度。
2. 将智能分析技术应用于视频监控系统中,借助计算机强大的数据处理能力过滤掉视频画面无用的或干扰信息、充分挖掘抽取视频资源中的关键信息,当出现异常情况或者发现可疑人物游荡在安全管理的区域以及其他的异常行为,系统会根据事件的可疑程度做出智能化的判断,并以最快和最佳的方式发出警报或触发其它动作,从而有效的抑制了人作为主体所引发的一系列问题,从而真正意义上实现了事前预警,事中处理,事后及时取证的全自动、全天候、实时监控的智能系统。
3. 在行人进出门禁时，智能检测出行人，并进行跟踪检测。智能判断该行人是进门还是出门，若是走出门禁系统，则应给出该行人出门禁的提示信息。

3.3 系统开发环境介绍

3.3.1 操作平台

目前，Windows系统是最为流行的一种操作系统，在全球桌面市场中占有90%左右的份额，同时在中低端服务器市场也有广泛的应用，如Web服务器和数据库服务器[1]。其中，Windows 7系统兼容性好,对新技术、新产品的支持良好,是设计开发人员常用的操作系统。另外,Windows 7系统集成了微软的防火墙技术,保障了用户计算机使用安全;

Windows 7 具有超强的兼容性，目前，总共有来自10,000家不同公司的32,000个人参与到围绕Windows 7的测试计划当中，其中包括5,000个硬件合作伙伴和5,716个软件合作伙伴。全球知名的厂商比如Sony、ATI、NVIDIA等等都表示将能够确保各自产品对Windows 7正式版的兼容性能。 据统计，目前适用于Windows Vista SP1的驱动程序中有超过99%已经能够运用于Windows 7。这对于一款仍然处于测试开发阶段的操作系统来说，是一项卓越的成就。

现在仍然有许多用户坚守着Windows XP的阵地，为的就是它强大的兼容性，游戏、办公，甚至企业级应用全不耽误。同时也有许多企业仍然在使用XP。为了让用户，尤其是中小企业用户过渡到Windows 7平台时减少程序兼容性顾虑，微软在Windows 7中新增了一项Windows XP模式，它能够使Windows 7用户由Windows 7桌面启动，运行诸多Windows XP应用程序。

综合上述特点,Windows 7满足了本项目对操作系统的需求,故选之为视频监控系统的开发平台。

3.3.2 开发环境

Windows平台下选择Microsoft Visual Studio 2013开发工具,它是目前最流行windows平台应用程序开发环境[2]。Visual Studio 2013是一个基本完善的开发工具集它包括了整个软件生命周期中所需要的大部分工具，如UML工具、代码管控工具、集成开发环境(IDE)等等。所写的目标代码适用于微软支持的所有平台，包括[Microsoft Windows](http://baike.baidu.com/view/182749.htm)、[Windows Mobile](http://baike.baidu.com/view/40733.htm)、[Windows CE](http://baike.baidu.com/view/41539.htm)、[.NET Framework](http://baike.baidu.com/view/18370.htm)、[.NET Compact Framework](http://baike.baidu.com/view/1546355.htm)和Microsoft [Silverlight](http://baike.baidu.com/view/942429.htm) 及[Windows Phone](http://baike.baidu.com/view/2708518.htm)。

[Visual Studio](http://baike.baidu.com/view/28727.htm)是目前最流行的[Windows](http://baike.baidu.com/view/4821.htm)平台应用程序的[集成开发环境](http://baike.baidu.com/view/14867.htm)。Visual studio 2013具有功能全面,灵活性好,效率高,深入底层的优点。另外,Visual Studio 2013具有MFC AppWizard/Class Wizard等功能强大的编程向导工具，在很大程度上简化了应用程序的开发,也提高了编程效率。Visual Studio 2013自然是本项目开发的首选。

3.4本章小结

本章从系统整体结构出发,介绍了本项目中监控系统的系统框架以及组成结构,由界面显示系统和检测系统组成。并且对系统的功能及要求进行了阐述,最后,简单介绍客户端软件的开发环境,为第四章中的模块开发做铺垫。

第4章 视频监控系统软件实现

门禁监控系统是软硬件结合的两层结构体系，常言说硬件是基础，软件是灵魂，软件设计陈红与否是系统成功的关键。本章主要介绍了门禁视频监控系统的软件实现。在本系统中采用MFC实现人机交互，系统的大多数功能是通过人机交互的操作来实现的。因此界面设计是本章的重点。

4.1 系统设计原则

视频监控系统都是安装在安全性比较高的，这就对监控系统的设计提出了较高的要求，系统不仅要时刻保持工作状态，而且性能稳定，同时也不能因为监控换件变化而使监控质量降低。另外，监控系统还要有升级的能力，这些都应该在系统设计的时候需要考虑。本系统设计遵循的原则主要有以下几条：

实时性：实时性原则要求设计出的监控系统能够实时播放监控画面。实时采集现场数据，而且系统在确认可疑目标后立即发出报警，起到实时监控现场的作用。

可靠性：可靠性是系统的重要性能，一旦系统瘫痪，其后果是难以想象的。设计系统应严格要求，保证系统能够稳定的运行，减少维护频率和周期，提高监控质量。

可升级性：本系统采用的技术和设备应该能够适应以后的发展，一旦更换设备或者采用更为先进的技术，能够将系统快速升级。

智能性：智能性要求监控系统能够根据预定的原则，自主判断事件的发生，是否发生预警，在确定可疑目标时，还需要对目标进行跟踪，并实时的保存录像图片。

安全性：系统具有安全防范和保密措施。确保数据在传输过程中的安全性，进而防止不法分子的窃取和篡改。同时防止用户的非法操作。

实用性原则：易于操作，模块化设计便于维护和管理，具有良好的人机界面。

4.2 功能需求

因为本系统的运行需要进行人机交互：操作人员选择要监控的视频，同时选择是否开启检测功能，系统自动显示出不同功能选择下的监控画面。界面上的模块设计决定了该系统是否具有良好的人机交互，是否能够及时有效的操作系统，根据需求，设计如下模块：

1. 视频源选择模块

主要完成视频源的选择，可以选择本地已经存储好的视频文件，也可选择正在拍摄的监控视频。

1. 视频显示模块

根据选择的视频源，播放对应的视频。如果开启视频检测，则显示检测后有检测标记的视频。

1. 视频信息显示模块

显示选择的视频源的文件信息。包括文件名，文件路径以及播放的时间。

1. 监控信息显示模块

若开启检测功能，则显示对所播放的视频进行检测后的信息，进出门的人员数（进入、离开）以及发生异常情况(未刷卡就进入)的视频截图。若未开启，则显示“请开启检测功能”。

1. 操作记录显示模块

负责监控人员对系统的操作记录显示，包括选择视频源、关闭视频、开启检测、关闭检测等。

1. 功能选择模块

监控人员可以自由的选择是否开启检测功能。若选择开启检测功能，则视频显示会立即显示后面未播放视频的检测图像。

1. 检测算法模块

主要负责对开启检测功能后对视频图像进行检测分析，识别检测目标，统计进出人员数，并检测后的结果交由显示界面进行显示。

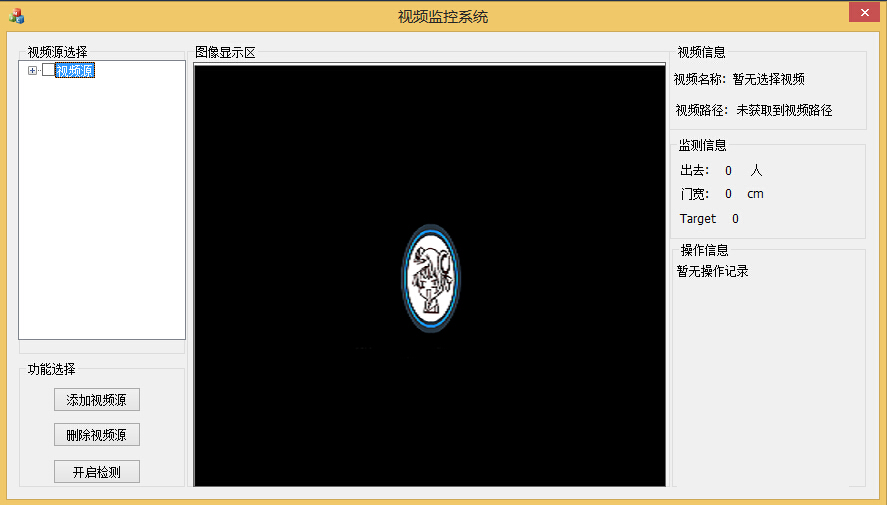
系统具体软件结构图如图4.2.1所示



图4.2.1

4.3 基于MFC的用户界面实现

系统整体界面设计如图4.3.1所示

图4.3.1

各个具体模块实现如下：

1. 视频源选择模块



图4.3.2

从数据源中获取添加的视频信息，其中包括自定义的视频名和视频的路径。监控人员选择需要监控的视频源，左键单击视频名前的CheckBox框，系统自动加载该视频，并在视频显示模块显示相应的视频画面和在视频信息处显示视频名以及视频路径。具体流程图如图4.3.3所示

图4.3.3

视频源选择模块可以进行视频的选择播放以及关闭正在播放的视频。若监控人员选择的视频不存在(视频路径为空或者视频格式不正确)时，系统给出出错提示，不会出现程序异常。

在选中视频时，若选中父节点，则将子节点全部选中，如子节点被选中，则对应的父节点也应被选中，在OnNMClickTreeSource消息中实现方法如下:

CPoint oPoint;

UINT nFlag;

GetCursorPos(&oPoint);

m\_treeContrl.ScreenToClient(&oPoint);

HTREEITEM oSelectItem = m\_treeContrl.HitTest(oPoint, &nFlag);

if (oSelectItem == NULL)

{

return;

}

if (nFlag & TVHT\_ONITEMSTATEICON)

{

BOOL bCheck = !m\_treeContrl.GetCheck(oSelectItem);

if (bCheck)

{

PlayVideo(m\_treeContrl.GetItemText(oSelectItem));

//AfxMessageBox(m\_treeContrl.GetItemText(oSelectItem));

}

else

{

CloseVideo();

}

// 为了保持一致化属性控件的选中状态，需设置当前选中项的复选框为改变后的状态

m\_treeContrl.SetCheck(oSelectItem, bCheck);

// 一致化树形控件复选框状态

ConsistentChildCheck(oSelectItem);

ConsistentParentCheck(oSelectItem);

// 将当前选中项的复选框状态复原，MFC会自动响应复选框状态的改变绘制

m\_treeContrl.SetCheck(oSelectItem, !bCheck);

}

1. 功能选择模块

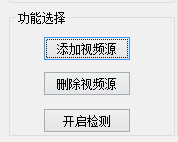


图4.3.4

功能选择负责对视频的一些简单操作，包括添加视频源，删除已经存在的视频源以及选择视频源后是否开启检测功能。具体流程图如图4.3.5所示



图4.3.5

添加视频源功能，在具体实现过程中，我们使用一个继承CDialogEx的对话框CAddVideo来实现视频源的添加。界面如图4.3.6所示：

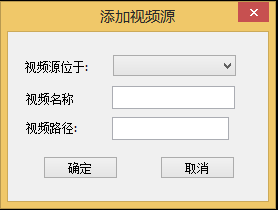


图4.3.6

其中用到的数据结构和方法说明如下：

CComboBox m\_source; // 视频类型

CString m\_name; // 视频名称

CString m\_path; // 视频路径

virtual BOOL OnInitDialog(); // 继承方法，重写以增加对话框的初始化

afx\_msg void OnBnClickedButtonSure(); // 点击确定后发生

afx\_msg void OnBnClickedButtonCancle(); // 点击取消后发生

bool CheckVideoName(int videoType); // 验证视频名称是否重复

void StorageVideoInfo(int type); // 存储视频源信息

删除视频源功能，在具体实现过程中，我们使用一个继承CDialogEx的对话框CDeleteVideo来实现视频源的添加。界面如图4.3.7所示：

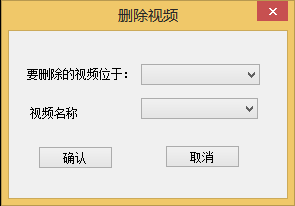


图4.3.7

其中用到的数据结构和方法说明如下：

afx\_msg void OnBnClickedButtonOk(); //点击确定

afx\_msg void OnBnClickedButtonCancle();//点击取消

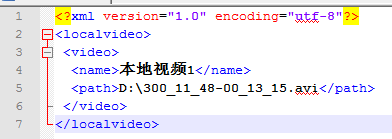
afx\_msg void OnCbnDropdownComboName();//视频名称下拉框点击时发生

virtual BOOL OnInitDialog(); //继承而来，用于初始化对话框

CComboBox m\_name; // 视频名称

CComboBox m\_parent; // 视频所处的节点

视频源存储于xml文件中，分为本地视频、网络视频，分别为C盘data目录下的”LocalData.xml”和”InterData.xml”中，根节点使用localvideo和Intervideo作为根节点来识别存储的是本地视频还是网络视频，然后在根节点使用video节点表示一个视频，在video节点下使用name和path分别表示该视频的视频名称和视频路径，具体存储格式如下图所示：



删除视频源则是将对应的xml文件中表示该视频的video节点以及该节点下的name和path删除。

开启视频监测后，若已选择视频源，则界面显示如图所示:



图4.3.8

1. 图像显示区模块

图像显示界面如图所示:

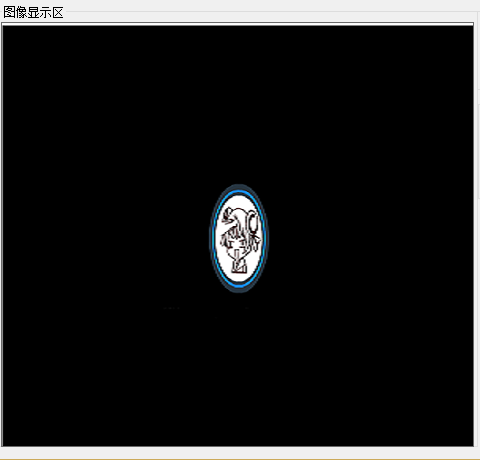


图4.3.9

没有选择视频源或者视频源播放完毕时，界面显示如图所示，反之显示相应的视频画面。其对应的变量名称为

IplImage\* m\_img; // 显示图像

在显示视频图像时，首先使用MFC中的OnTimer方法开启计时器，然后使用Opencv中的IplImage对象，获取视频的每帧图像。然后使用双缓冲技术将视频图像显示出来。

开启定时器方法: SetTimer(1, 150, NULL);

获取视频图像的每帧图像: IplImage\* tep = video->GetCurrentFrame();

使用双缓冲技术防止图像显示时闪烁:

CRect rect; // 获取绘制设备

CDC \*pDC = GetDlgItem(IDC\_STATIC\_PIC)->GetDC();

HDC hDC = pDC->GetSafeHdc();

GetDlgItem(IDC\_STATIC\_PIC)->GetClientRect(&rect);

// 采用双缓冲作图

CDC dcMem; // 用于缓冲作图的内存DC

CBitmap bmp; //内存中承载临时图象的位图

dcMem.CreateCompatibleDC(NULL); //依附窗口DC创建兼容内存DC

//创建兼容位图

bmp.CreateCompatibleBitmap(pDC, rect.Width(), rect.Height()); dcMem.SelectObject(&bmp);//将位图选择进内存DC

//按原来背景填充客户区，不然会是黑色

dcMem.FillSolidRect(rect, pDC->GetBkColor()); dcMem.SelectStockObject(NULL\_BRUSH);

CvvImage cimg;

cimg.CopyOf(m\_img);

cimg.DrawToHDC(dcMem, &rect);

//将内存DC上的图象拷贝到前台

pDC->BitBlt(0, 0, rect.Width(), rect.Height(), &dcMem, 0, 0, SRCCOPY);

dcMem.DeleteDC(); //删除DC

bmp.DeleteObject(); //删除位图

1. 视频信息模块



图4.3.10

负责显示选择视频源后对应的视频名称和视频路径。若未播放视频或者视频播放结束后将出现视频名称为暂无选择视频，视频路径为未获取到视频路径。其中所使用党的数据类型和方法如下:

// IDC\_STATIC\_NAME为视频名称所对应的ID

SetDlgItemText(IDC\_STATIC\_NAME, videoname);

// IDC\_STATIC\_PATH为视频路径所对应的ID

SetDlgItemText(IDC\_STATIC\_PATH, videopath);

1. 检测信息模块

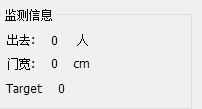


图4.3.11

负责显示开启检测功能后，显示图像检测信息。在开启视频检测后，检测信息每次根据当前检测到的当前的视频图像信息进行动态显示，即每检测一张图片视频，则检测信息将动态的更新一次。其中:

出去人数的ID为：IDC\_STATIC\_OUT

门的宽度的ID为: IDC\_STATIC\_DOOR

目标的ID为: IDC\_STATIC\_TARGET

动态更新方法为:

CString temp;

temp.Format(\_T("%d"), outCount);

SetDlgItemText(IDC\_STATIC\_OUT, temp);

temp.Format(\_T("%d"), doorlen);

SetDlgItemText(IDC\_STATIC\_DOOR, temp);

temp.Format(\_T("%d"), targetcount);

SetDlgItemText(IDC\_STATIC\_TARGET, temp);

1. 操作信息模块



图4.3.12

负责对操作人员操作界面的操作记录的显示。在这里，可以看见操作人员对软件的操作信息(打开视频，关闭视频。添加视频源。删除视频源等等)。

操作信息的ID为: IDC\_STATIC\_OPERATE

显示操作信息的方法为:

CString temp;

GetDlgItemText(IDC\_STATIC\_OPERATE, temp);

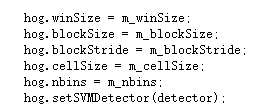
temp = temp + "\n" + message;

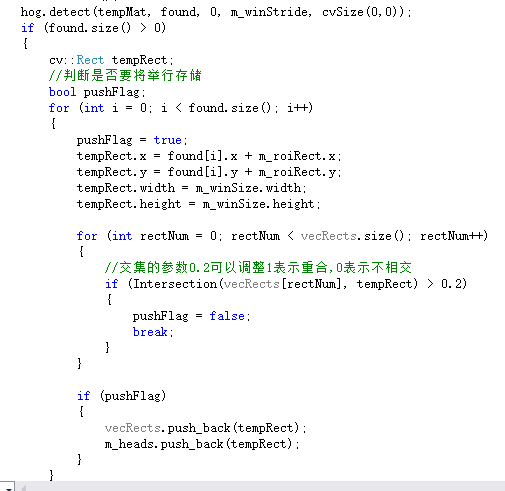
SetDlgItemText(IDC\_STATIC\_OPERATE, temp);

4.4 基于OpenCV的图像处理实现

4.4.1 人头检测的实现

使用OpenCV中的HOGDescriptor hog对象来实现人头特征的提取。首先加载“HogDetector.txt”的特征数据，使用setSVMDetector方法给用于对hog特征进行分类的svm模型的系数赋值，然后hog中的单尺度检测算法进行检测，具体实现如图所示:





其中：

winSize ： 窗口的大小

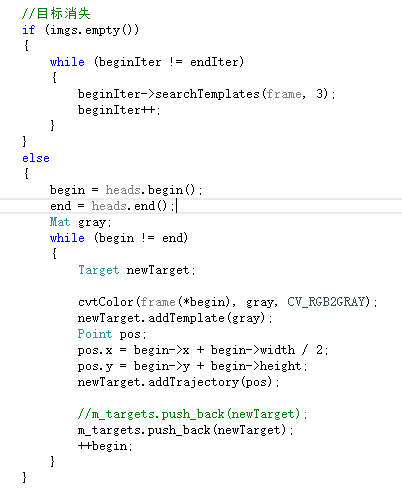
blockSize ：块的大小

cellSize： 胞元的大小

nbins： 方向bin的个数 nBins表示在一个胞元（cell）中统计梯度的方向数目，例如nBins=9时，在一个胞元内统计9个方向的梯度直方图，每个方向为360/9=40度。

4.4.2 目标跟踪

目标跟踪即将检测到目标后对目标进行跟踪，用于判断后续的异常发生时进行捕捉和截图保存。具体代码实现如下:



4.5 本章小结

本章主要介绍了系统的组成以及各个模块之间的实现。在图像检测中，介绍了人头检测和跟踪的主要实现。本章从系统总体上进行分析，根据设计原则，从软件角度详细介绍了视频源添加/删除，图像检测，图像显示、视频信息以及检测信息等各个功能。

第5章 测试、总结与展望

5.1 测试

本次测试主要测试门禁视频监控系统的视频源添加、删除、视频播放以及开启检测功能，实现相关功能的正常运行。测试方案如下:

|  |  |  |  |
| --- | --- | --- | --- |
| 测试功能 | 测试用例 | 预期结果 | 实际结果 |
| 视频源添加 | 1. 运行门禁监控系统，点击添加视频源 | 1.弹出添加视频源对话框，且正常展示一个ComBox下拉框，不可编辑，两个编辑框。 | 符合预期结果 |
| 1. 点击确定 | 出现提示，不可进行保存操作 |
| 1. 依次填写视频源信息，点击保存，然后查看对应的xml数据文件 | 保存成功，且对应的xml数据文件中有此次记录 |
| 视频源删除 | 1. 点击删除视频源按钮 | 弹出删除视频对话框，且该对话框上正常展示两个下拉选择框， | 符合预期结果 |
| 1. 点击确定 | 出现提示信息，不能进行删除操作 |
| 1. 依次选择删除的视频源信息，点击确定，然后查看对应的xml数据文件 | 删除成功，且对应的xml数据文件没有该视频的记录 |
| 视频播放 | 选择一个视频文件，左键点击视频名前的单选框 | 视频名前的单选框被选中，且对应的父项也被选中，视频显示界面显示对应的视频 | 符合预期结果 |
| 开启视频检测 | 单击开启检测功能按钮 | 图像显示界面立即显示检测后的图像，若有符合检测结果的，则在图像上有相应的标识 | 符合预期结果 |
| …… | …… | …… | …… |

5.2 总结

本文的研究课题是“门禁监控视频软件设计与实现”，主要研究的内容和目的是:

选择OpenCV中的模板匹配算法进行视频图像的人物检测，以及复杂的数学算法进行人物跟踪和判断是否出门禁，设计以OpenCV为运动框架基础的运动检测、分析、跟踪系统。硬件采用一般采用的视频摄像头，结合MFC进行界面设计综合分析运动物体的运动情况。

随着数字化、网络化、智能化的步伐加快，视频监控系统的应用范围逐步扩大，在其功能、性能、可靠性等方向提出了更高的要求。虽然现在的门禁监控系统已经基本上满足了用户的需求，但是，仍然存在许多缺点。我们了解了现有的视频监控系统所处的环境及绵连的问题，在前人工作的基础上，设计了一套全新的门禁视频监控系统。采用模块化结构设计视频监控系统的软件部分，给出了系统的实现结果。本文主要的工作和创新点如下:

1. 以Visual Studio 2013为开发环境，基于对话框模式开发了系统应用程序，系统界面设计简洁、美观、人性化、适合不同层次的使用者操作。
2. 根据项目实际需求，采用模块化的结构设计，将系统进行功能模块划分。设计出了一个稳定、高效、人性化的门禁监控系统。
3. 在传统的门禁监控视频上进行了创新。解放了视频监控人员的双手，提高了工作效率。使传统的门禁监控系统更加智能化，更加符合现代人的实用。
4. 系统扩展、升级新强。系统在设计的初期，根据系统设计原则，为系统以后的升级，扩展做好的基础。

5.3 展望

本系统虽然基本上满足了预期的效果，但由于开发周期短，系统仍然存在许多的不完善的地方，需要进一步改进：

1. 随着现在网络的普及，越来越多的系统都采用B/S或者C/S结构，所以系统可以采用C/S或者B/S结构，将系统的检测功能移到服务器上，而操作人员只需要远程点击检测功能，由客户端发送视频图像，在服务器上进行检测，返回检测后的视频图像信息。
2. 视频检测过程中，占用太多的CPU和内存，在操作系统为:Win8.1专业版，CPU为:Inter(R) Core(TM) i3-2330M 内存为6G的环境下，开启检测功能耗掉35%的CPU,40%的内存，所以可以考虑使用更高效率的处理算法进行性能的优化。
3. 系统在安全性方面还需要进一步的加强。为用户提供可靠的保障。
4. 在系统的开发过程中，对视频图像处理的技术有一定的了解，但由于作者本人能力和时间的限制，对视频图像处理处理的难免有些不足之处，不免有些错误，所以还需要在该方面做进一步的研究。

参考文献

1. 陈东伟,韩娜.嵌入式数据库在基于多核处理器的视频监控中的应用〔J〕.郑州大学2007(4):52一55.
2. 杨富国.Visua1 C++程序开发案例解析[M].北京:清华大学出版社;北京交通大学出版社 2006.
3. 王华. 智能视频监控系统设计与实现. 山东大学 2013
4. 刘瑞祯.OpenCV教程基础篇[M]. 北京：北京航空航天大学出版社 2007
5. http://baike.baidu.com/link?url=Okljl6hY78kqeVfZlH0P6QYE6adILhpsApQuG2Jy9fXrYa\_XXIhI\_UexDUA4DbhS6Gif1Axs5ys-uktUwsLFJf9Td0EWqV3sOAqlPxINJF3
6. http://www.opencv.org.cn/opencvdoc/2.3.2/html/doc/tutorials/imgproc/histograms/template\_matching/template\_matching.html?highlight=matchtemplate
7. 赵玉萍.基于Visual C++的视频监控系统的软件设计与实现.太原.太原理工大学
8. [侯俊杰](http://baike.baidu.com/view/1886563.htm).深入浅出MFC.华中科技大学出版社
9. 黄佳.基于OPENCV的计算机视觉技术研究.上海.华中理工大学
10. Maggioni C and Kammerer B. "Gesture Computer: history,design, and applications Computer Vision of Human-Machine Interaction[J]". Cambridge Univ. Press, 1998.
11. Haritaoglu I,Harwood D,Davis L. UW4: real-time surveillance of people and their activities[J]H. IEEE Trans PaRem Analysis and Machine Intlligence,2000,22(8): 809—830
12. 郑金荣.视频监控系统中运动目标跟踪算法的研究[D].硕士学位论文,江南大学,2009
13. 徐振兴.基于机器视觉的行人检测和跟踪技术研宄[D].硕士学位论文,浙江理工大学,2012
14. 牛胜石. 基于 AdaBoost 和 SVM 的人头检测.硕士学位论文,中南民族大学，2010

致谢

光阴似箭,四年的本科生学习生活很快就要结束了。在西南科技大学的这四年里,我周围发生了很大的变化,最重要的是我四年前制定的目标实现了,四年让我收获颇丰。

本论文的工作是在我的导师刘畅老师的悉心指导下完成的。在本课题的研究以及论文的撰写的过程中,刘老师给与我精心的指导;在生活中刘老师也给与我了无微不至的关怀。正是这些指导和关怀使得我在本科生的学习期间取得了很大的进步。刘老师严谨的治学态度、科学的工作方法以及朴实的生活作风对我有着深远的影响。在此向刘畅老师表达衷心的谢意。

前几届的师兄任新宇对于我的工作学习以及论文都提出了许多宝贵的意见,对于论文的内容以及存在的问题提供了许多指导以及帮助。同时感谢所有在我学习期间教导过我的老师,谢谢你们在我求学的过程中给与的关怀和帮助,你们的一言一行都将成为我人生未来的宝贵财富。

在课题研究以及论文的撰写过程中,许多同学对我论文中的研究工作也提供了热情的帮助,在此向他们也表达我的感激之情。最后特别感谢我的父母,他们一直在我的身边默默地鼓励、关心、支持我,是他们的理解和支持使能够专注于研究和学习,顺利完成我的学业。