Smart Vision Software
设计说明书

2013 年 3 月 4 日

目录：

1 引言

1.1 目的

编写此文档的目的是：①宏观、详细和准确地定义该视频监控系统的概要设计，以利于指导该系统后续的开发工作；②文档描述的概要设计作为该项目最终验收的标准和依据；③给程序员提供维护依据，为后期的维修工作提供便利。

1.2 背景

- 系统名称：Smart Vision
- 任务提出者、开发者：Dream Team（梦之缘工作坊）
- 用户：各行业重点部门等需要实时监控的场所

1.3 定义、缩写词、略语

- IP Camera：即网络摄像机，其内置数字化压缩控制器和基于 WEB 的操作系统，经压缩加密后的视频数据可通过局域网、Internet 或无线网络送至终端用户。
- 云台：安装、固定摄像机的支撑设备。
- 帧率（Frame Rate）：用于测量显示帧数的量度。
- UI：即 User Interface（用户界面）的简称。
- MJPEG：为 24-bit 的"true-color"影像标准，MJPEG 的工作是将 RGB 格式的影像转换成 YCrCB 格式，目的是为了减少档案大小，一般约可减少 1/3~1/2 左右。
- JPEG：一种支持 8 位和 24 位色彩的压缩位图格式，适合在网络（Internet）上传输。
- CGI：计算机网关接口，为外部应用程序与 Web 服务器之间的接口标准。

1.4 参考资料

- Camera Vision 开源软件。下载网址为：http://www.codeproject.com/Articles/15537/Camera-Vision-video-surveillance-on-C

1.5 郑重声明

本软件参考开源软件 Camera Vision 底层设计，把基于.net Framework 1.1 的 Camera Vision 底层代码重新编写整理，移植到.net Framework 4.0，并基于其 Mjpeg.dll 和 jpeg.dll，设计开发了对 EasyN IP Camera 定制的底层视频源连接支持库 EasyN.dll。沿用 videosource 接口设计，在其上进行上层设计开发，Camera Vision 仅有视频流播放功能，而本软件 Smart Vision 具备录制拍照回放检测等多种监控模式，更具备使用价值，自制图片视频一体化播放器，对历史数据进行有效管理，方便查询。对所有种类的 Camera 都具备录制拍照检测等基本功能，而我们对 EasyN IP Camera 的功能进行了定制，设计了 CameraCGI 类，分装了一系列对远程 Camera 的操作方法，可以设置图像参数，云台控制，报警设置等等，扩展监控能力，所以只要提供摄像头的接口我们就能对其进行功能定制。

2 总体设计

2.1 需求规定

所要设计与实现的系统是一个独立的软件系统，适用于需要实时监控的重要场所。该系统主要分为四大模块：

①摄像头配置模块： 实现对摄像头的添加、删除、参数设置（针对 EasyN IP Camera）和删除功能。

②视图模块：实现多摄像头视频的同页面显示和监控画面的全屏显示；实现对云台的上、下、左、右动作和镜头的调焦变倍的远程操控，并且可在多路摄像机之间进行选择。

③监控模块：提供视频录制、画面拍照以及回放已获取图像功能；同时，实现监控摄像头的地理位置显示。

④报警模块：在 IP Camera 检测到异常情况时，通过邮件、移动布防等方式及时反馈给用户。

2.2 运行环境

（1）硬环境

- 1 台 PC 机；

- 硬盘存储空间 4G 及以上（用于存放录制视频）；
- 1G 内存以上，P3 500MHz 以上级 CPU；

- 1 台无线路由器，搭建监控局域网和连接 Internet，网速要求 1M 以上；
- 2 台 EasyN IP Camera，内置服务器，提供 CGI 访问接口。

（2）支持软件

针对 windows 操作系统特别优化的内核，全面支持 Microsoft win32 平台和 win7 64 平台系列操作系统；Net Framework 4.0 架构；Microsoft Office Access，VS2010 C#。

2.3 系统基本设计概念

SmartVision 智能监控系统由 PC 机和 IP Camera 组成，PC 机通过 SmartVision 软件控制局域网内或者 Internet 上的 IP Camera，接收实时视频流，通过录制拍照移动布防等方式进行全方位智能监控。

系统概念总图如下图所示：

Fig1. 系统概念总图

2.4 软件层次架构设计

本软件采用典型三层架构软件设计模型，分为 UI 显示层，数据逻辑层和视频流访问层，层与层之间通过接口联系传递信息，分层设计，结构分明开发效率高，易于代码维护和后期功能扩展，如图所示。

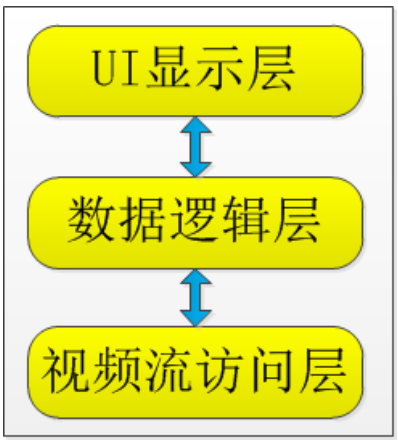

Fig2.软件设计架构模型图

（1）视频流访问层：

通过 Ivideosource 接口类，把各种视频源封装在底层，使系统更加稳定更易于扩展。该层为上层提供 Jpeg 格式，Mjpeg 视频流和本地摄像头的访问，基于 Jpeg，Mjpeg 和我们为 EasyN IP Camera 定制了底层接口，提供多种方式访问。如此的设计架构很方便视频源的扩展，而不影响上层设计。底层库最终以 dll 动态链接库的形式生成，视频流访问层的结构如图所示：

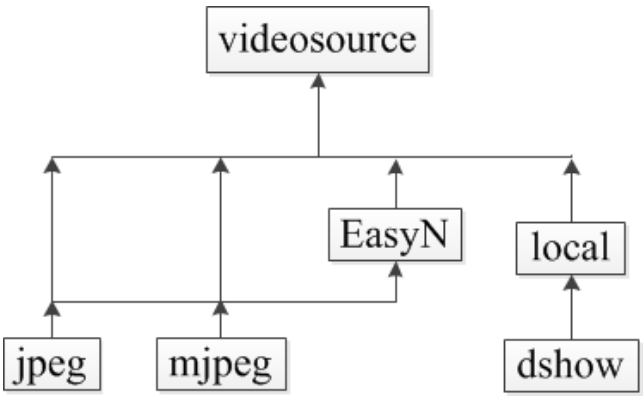

Fig3.视频流访问层结构图

（2）数据逻辑层

数据逻辑层包括各种功能的实现部分，连接底层和显示层，功能设计如下。

拍照：拍照的实现原理是：首先获取当前选中的摄像头，若当前选中了摄像头，则其通过 cameraToEdit.TakePhoto()将摄像头传过来了数据帧保存为 JPG 格式的图片并保存到指定目录；若当前未选中摄像头，则提示未选中摄像头。

定时抓拍：定时抓拍的原理和拍照是一样的，主要的区别是：定时抓拍采用定时器，没定时一段时间抓拍一张照片。

录制功能：不同视频源视频流帧率不同，但播放帧率均采用每秒 25 帧，为了实现最大程度得再现录制实况，采用定时器实现每秒装载 25 帧，在没有新帧到来时，重复装载最后一帧，以下是设计框图

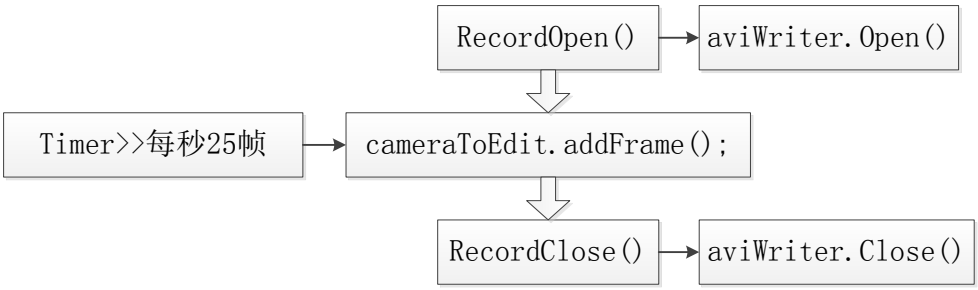

Fig4. 录制功能设计图

异常检测：采用前后两帧对比算法 TwoFramesDifferenceDetector()，返回实现对异常的监测，并设定阀值来设定监测灵敏度，以下设计框图：

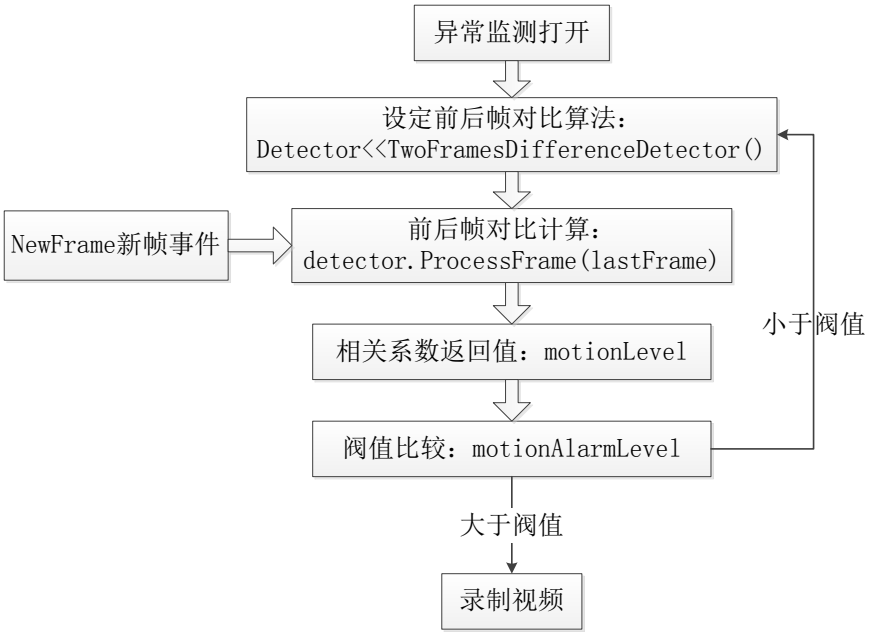

Fig5. 异常检测功能设计图

历史回放：包含两方面的内容：在功能方面，其首先需要能够将录制好的视频和图片加载到列表当中，并且实时刷新该文件信息；同时其也需要能够对这些文件进行编辑，这里采用的方式是打开录制好的视频和图片的所在文件夹，这样就可以按照需要进行文件编辑。在内容方面，其需要完善的功能也是历史回放模块的主要内容，即播放录制好的视频和图片，同时这里还实现了对图片进行选择性播放，这样就便于用户查看图片。

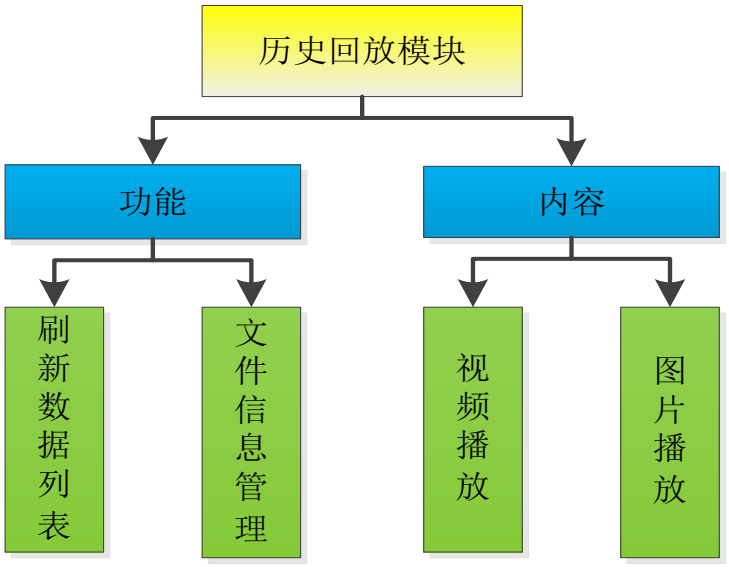

Fig6. 历史回放功能设计图

（3）UI 显示层：

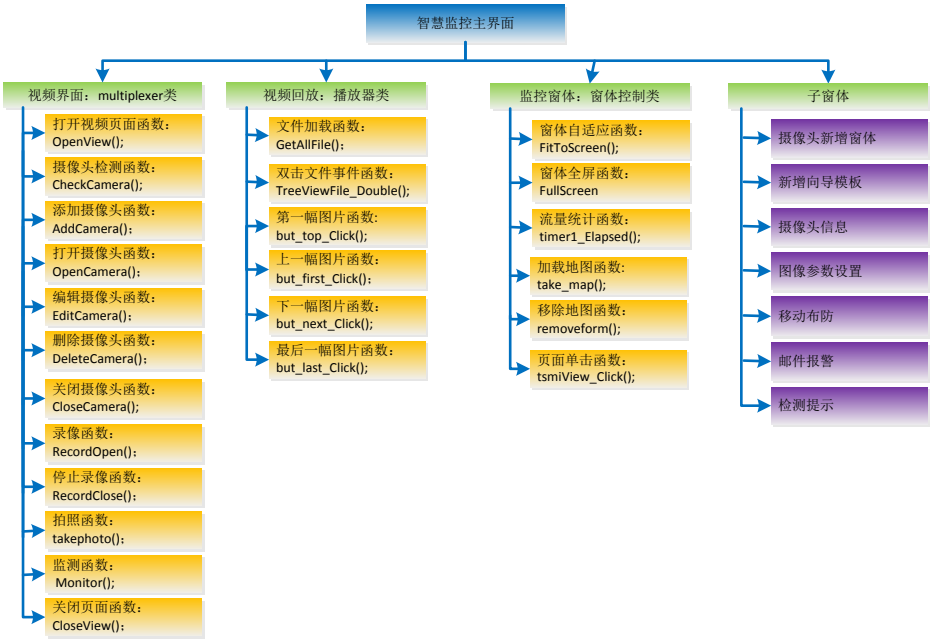

Fig7.系统软件主界面分布

CameraVision 主窗体类主要包含了三个成员变量：即 multiplexer 类，视频回放类和窗体监控类。通过操控这三个成员变量类可以实现摄像头的添加，删除，录制，拍照和回放等功能。以下为主要成员变量的说明。

Table1.主要成员变量及其中函数说明

主要成员变量	函数	函数说明
Multiplexer 类：通过此函数，可以打开摄像头页面，实现将摄像头的监控画面进行实时显示。	OpenView()	可以打开摄像头页面，实现将摄像头的监控画面进行实时显示。
	CheckCamera()	摄像头检测函数，主要用于检测页面上已经添加了摄像头。
	AddCamera()	可以实现向页面添加摄像头。
	OpenCamera()	用于将已添加了的摄像头在页面中打开显示。
	EditCamera()	用于对选中的摄像头信息进行编辑修改。
	DeleteCamera()	实现将选中的摄像头从列表中进行删除。
	CloseCamera()	将选中的摄像头从页面当中删除，但是未从摄像头列表中删除。
	RecordOpen()	用于对选中的摄像头进行视频录制。
	RecordClose()	用于停止对选中摄像头的录制。
	takephoto()	用于对选中摄像头进行拍照功能。
	Monitor()	用于对选中的摄像头进行视频监控。
	CloseView()	用于关闭界面中的摄像头页面。
播放器类：其成员函数主要有 6 个，主要是实现对录制好的视频和排好的照片进	GetAllFile()	用于获取图片和视频视频文件夹下面所有的文件。
	TreeViewFile_Double()	为 TreeView 节点的双击函数，用于实现对双击的图片和视频进行回放。
	but_top_Click()	实现播放图片文件夹中的第一幅图

行回放功能		片。
	but_first_Click()	实现播放当前图片的上一幅图片。
	but_next_Click()	实现播放当前图片的下一幅图片。
	but_last_Click()	实现播放图片文件夹中的最后一幅图片。
窗体控制类：通过对其函数的调用可以实现对窗体的简单操作	FitToScreen()	实现窗体最大化。
	FullScreen()	实现窗体全屏。
	timer1_Elapsed()	用于统计软件的流量大小。
	take_map()	实现加载地图模块。
	removeform()	实现移除已加载的地图模块。
	tsmiView_Click()	单击页面出发的事件。

除了上面主要的三个类，其添加了 7 个子窗体，用于获取和设置软件信息。具体子窗体功能如 Table2.所示。

Tabel2.窗体和功能说明

窗体	功能说明
摄像头新增窗体	新增摄像头窗体模块,用于获取新增摄像头的参数信息。
新增向导模板	新增摄像头窗体模板,作为摄像头新增窗体类的容器。
摄像头信息	用于显示摄像头的信息参数。
图像参数设置	用于设置图片参数。
移动布防	用于检测摄像头检测到有物体移动发送警报功能。
邮件报警	由于设置摄像头检测到异常以邮件形式发送警报。
检测提示	当摄像头检测到异常时，PC 提出提示信息进行提示。

3 类结构设计

3.1 自定义组控件类

自定义组控件类是为各种窗体服务的，内嵌入窗体容器中完成特定功能。

主要的自定义组控件类有：Multiplexer 组件，CameraWindow 组件，地图控件，菜单栏单选控件，云台控制控件等。

Table3.组件及实现功能

组件	功能
Multiplexer	Multiplexer 多路视频播放容器，视频监控画面的底层总容器，上面内嵌多个 CameraWindow 组件。
CameraWindow	视频播放窗，在每个 CameraWindow 中呈现特定摄像头的实时画面，和 Camera 类绑定，完成和底层视频源的连接，在屏幕上绘制视频新帧。

其中 Multiplexer，CameraWindow，Camera 的结构从属关系如下图所示（非继承关系）：

Fig8.组件与 Camera 类结构关系

Table4.控件及实现功能

控件	功能
地图控件	网页方式调用 Google map API，内嵌入窗体中，用于定位 IP Camera（如 Fig9.所示）
菜单栏单选控件	主菜单选项 C#不提供单选功能，整合单选按钮重制菜单选项，实现视图模式的单选效果（如 Fig10.所示）。
云台控制控件	自制云台控制盘，内嵌入 CameraWindow 视频窗实现用户操

	作交互，通过 Camera CGI 接口远程控制 IP Camera 云台（如 Fig11.所示）。

地图控件：基于 Google Map API V2，实现窗口的地图部分。C#读取数据库数据，通过调用 Movemarker()和 Flyto()呈现在地图上面；同时，在 Google Map 上右击窗口可以同时获取摄像头信息和地理经纬度两方面的信息，这样就将添加的摄像头和经纬度匹配起来了，可以实现通过双击摄像头，跳到该摄像头在地图上的地理位置。

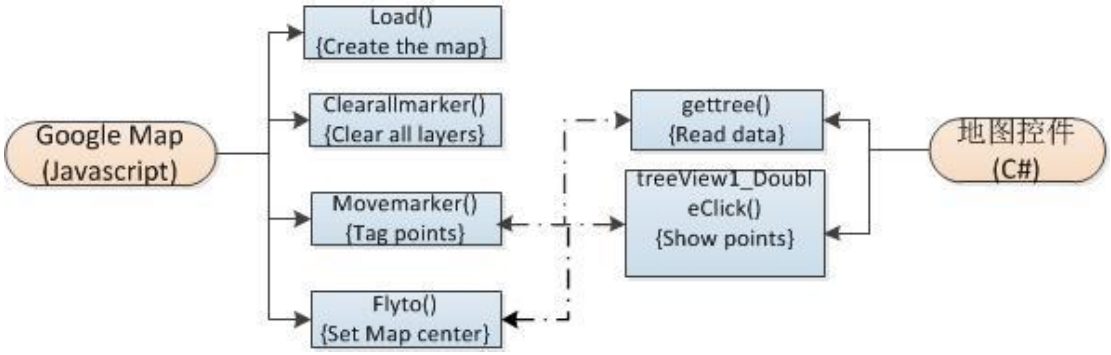

Fig9. 地图控件设计框图

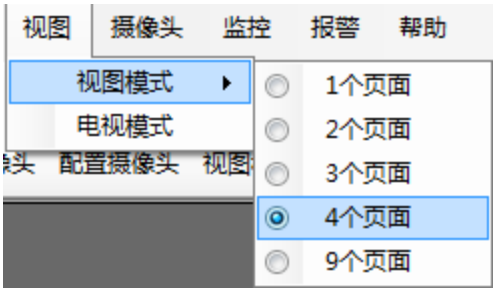

Fig10.视图模式单选效果图

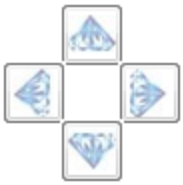

Fig11.云台控

3.2 子窗体类

子窗体类继承系统 Form 父类，实现子窗体信息呈现，完成用户交互，子窗体列表如 Fig8.所示。

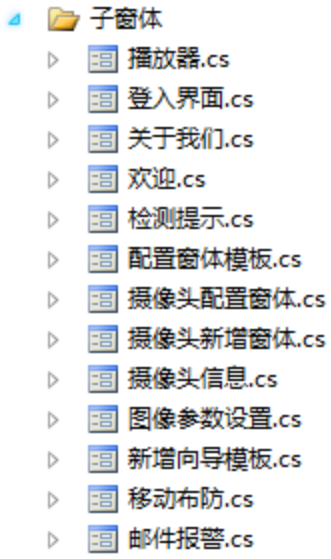

Fig12.子窗体类列表

部分核心子窗体说明如 Table5.所示。

Table5.部分核心子窗体及其说明

窗体	说明
播放器子窗体	内置 axWindowsMediaPlayer 播放器组件和 picturebox 组件，一体化回放录制视频和历史照片。
欢迎和登陆子窗体	利用定时器来实现淡化的欢迎见面，连接数据库，设置系统软件的访问权限。
摄像头相关子窗体	连接底层视频源，实现摄像头添加配置修改等操作。
CGI 接口类子窗体	通过调用 Cameracgi 接口，实现 Camera 报警设置，图像参数设置等。

3.3 核心类

核心类完成了系统最核心的功能，即对摄像头的一系列定义或操作，以 Camera 类为中心延生出整个核心类关系地图，以携带特定摄像头的所有信息的 Camera 类为纽带，把不同的核心类联系在一起，协同工作，关系地图如图 Fig8. 所示。

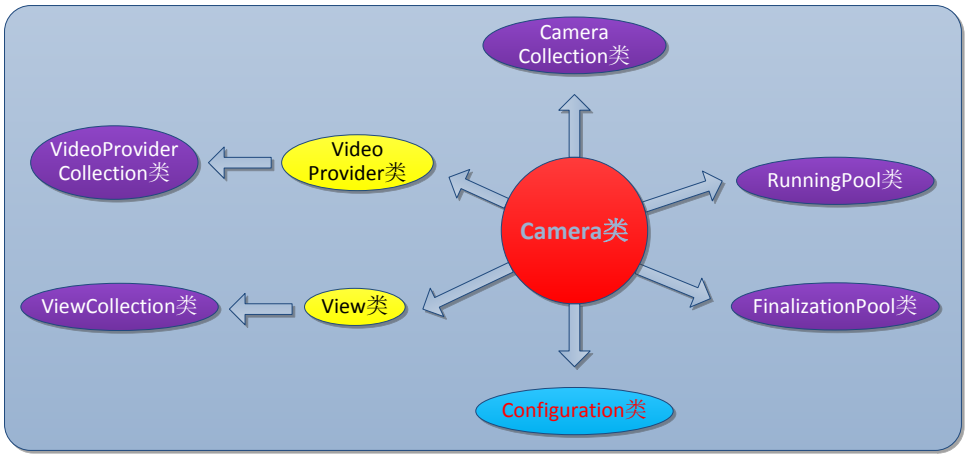

Fig13.不同核心类关系图

其中，VideoProvider 类为摄像头的视频源类，从底层获取相应视频源的连接支持。View 类提供了不同 Camera 在主窗体同时播放时的布局信息，连接自定义组件类中的 Multiplexer 多路视频播放容器，进行实时视频流播放。Configuration 类是整个系统的配置信息类，包括摄像头集合的信息，主窗体的信息等等，同时也提供底层数据的连接功能用于保存加载配置信息。

外层紫色类均继承于系统集合类抽象基类 CollectionBase，给相应的类提供集合存储和查询提取。其中 RunningPool 类和 FinalizationPool 类，是视频流线程的集合，用于创建打开和关闭摄像头实时视频流连接线程。

3.4 EasyN IP Camera 深入定制类

对 IP Camera 的深入定制如云台控制，图参设置等，需要拥有特定 Camera 的开发接口，SmartVision 软件对 EasyN IP Camera 进行了深入定制开发，通过它的 CGI 接口，实现对摄像头的控制，提高用户的监控能力。CameraCGI 类定义了 CGI 网络连接的标准模式，上层应用只要按照此标准调用 CameraCGI 类，传递特定参数即可实现对摄像头的控制，CameraCGI 类分装了底层网络连接实现函数，上层应用非常方便扩展，如下图所示本软件深入定制的 4 项控制项。

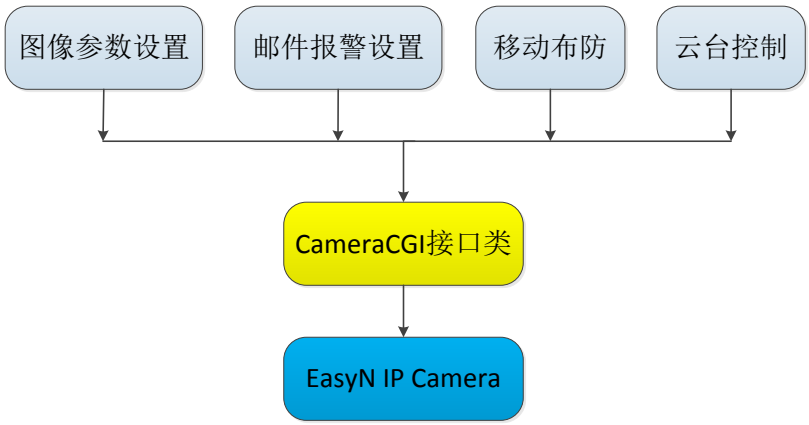

Fig14.软件深入定制关系图

CameraCGI 通过后台线程进行接口连接：

```
// 分辨率设置
public void set_Resolution(int value)
{
        string cgiURL =
string.Format("{0}/camera_control.cgi?param=0&value={1}", source, value);
        cgiConnect(cgiURL);
 }
// cgi 连接函数
private void cgiConnect(string cgiURL)
{
     //创建后台线程
      backgroundWorker1 = new BackgroundWorker();
     //注册事件
      backgroundWorker1.DoWork += new
System.ComponentModel.DoWorkEventHandler(this.backgroundWorker1_DoWork);
      backgroundWorker1.RunWorkerAsync(cgiURL);
}
//后台线程连接网络
private void backgroundWorker1_DoWork(object sender, DoWorkEventArgs e)
{
```

```
        try
        {
            BackgroundWorker bw = (BackgroundWorker)sender;
            string url = e.Argument.ToString();
            WebRequest req = WebRequest.Create(url);
            req.Credentials = new NetworkCredential(login, password);  //IP Camera 访问验证
            WebResponse resp = req.GetResponse();
            resp.Close();
        }catch(Exception)
       {}
}
```

4 自定义事件设计

CameraEventHandler 自定义事件执行结构图如下：

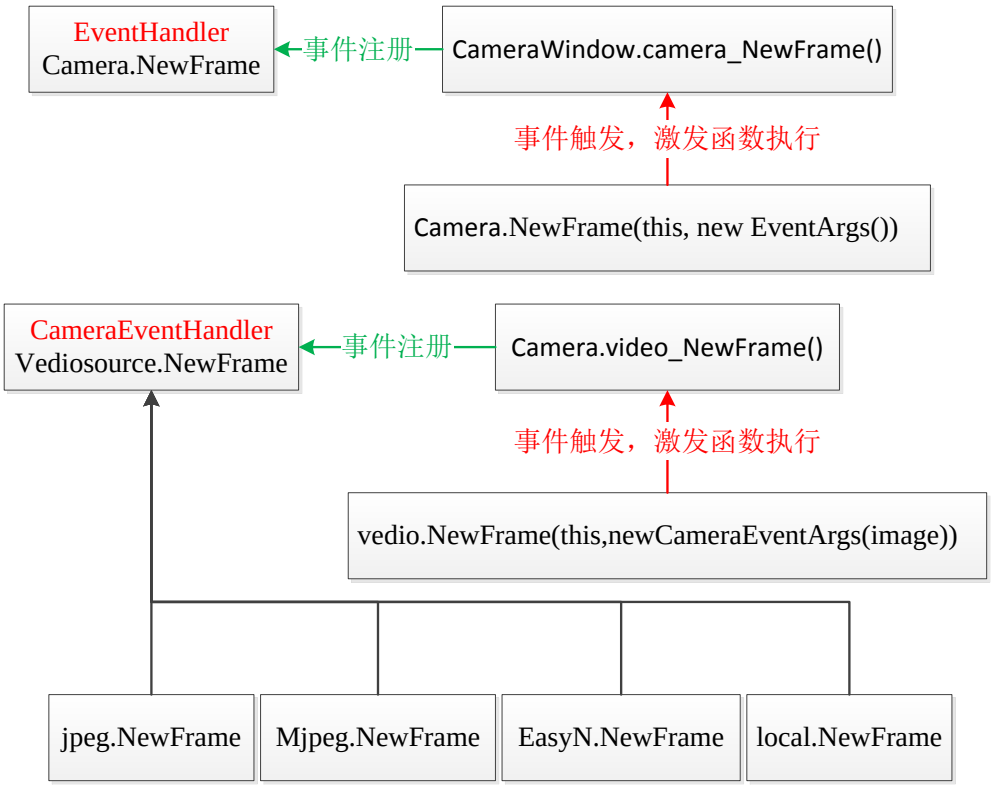

Fig15. CameraEventHandler 事件执行结构图

在 videosource.event 内

```
//创建委托，用于事件类型的定义
public delegate void CameraEventHandler(object sender, CameraEventArgs e);
```

- 对于下层在每个视频源实现类中声明绑定此委托的 NewFrame 事件,当有新图帧到达底层时触发此 NewFrame 事件，并且通过 videosource 接口传递信息，激发 Camera 类中的注册进 videosource NewFrame 事件的函数 video_NewFrame().

以 Local Capture Device 为例，：

```
//创建事件
public event CameraEventHandler NewFrame;
//当有新帧传到时，触发 NewFrame 事件，信息通过接口传递
protected void OnNewFrame(Bitmap image)
{
    framesReceived++;
    if ((!stopEvent.WaitOne(0, true)) && (NewFrame != null))
        NewFrame(this, new CameraEventArgs(image));
}
```

- 对于上层，在创建 Camera 时一方面创建自己的 EventHandler NewFrame 事件，用于传递动作给 Camerawindow 类来实时绘制视频 流，另一方面将 video_NewFrame()函数注册进接口 videosource 的 NewFrame 事件中：当接口 videosource NewFrame 被触发时（传递底层视频源实现类的信息），激发 Camera 类中注册进 videosource NewFrame 事件的函数 video_NewFrame()，Camera 类内执行图帧的精确拷贝，同时触发 Camera 类自己 EventHandler NewFrame 事件，激发 CameraWindow 类中注册进 Camera 类 EventHandler NewFrame 事件的函数 camera_NewFrame()，进行窗体重绘，呈现实时视频流，代码举例如下：

```
// 声明 Camera 类自己的 NewFrame 事件
NewFrame(this, new EventArgs());
// 创建接口
private IVideoSource videoSource = null;
// 注册事件，当底层 NewFrame 事件发生时，激发函数 video_NewFrame
IVideoSource videoSource.NewFrame += new
CameraEventHandler(video_NewFrame);
// 被注册进接口事件的函数，图帧精确拷贝和触发 Camera 类自己的
NewFrame 事件
private void video_NewFrame(object sender, CameraEventArgs e)
{
    // 线程加锁
    Monitor.Enter(this);
    // 清理旧帧
    if (lastFrame != null)
    {
        lastFrame.Dispose();
     }
     // 复制图片
     lastFrame = (Bitmap)e.Bitmap.Clone();
     width = lastFrame.Width;
     height = lastFrame.Height;

     // 解锁
     Monitor.Exit(this);

     // 通知客户端 camera 自己的新帧事件
     if (NewFrame != null)
     {
```

```
                NewFrame(this, new EventArgs());
            }
        }
    //以下为 Camerawindow 类内的相关函数
    //将 camera_NewFrame()函数注册进 Camera 类的 NewFrame 事件
    camera.NewFrame -= new EventHandler(camera_NewFrame);
    //Camera 类的 NewFrame 事件被触发时，激发此函数重绘视窗
    private void camera_NewFrame(object sender, System.EventArgs e)
    {
        Invalidate();       //重绘视频窗
    }
```

注意

Camera NewFrame 事件为 EventHandler 类事件，

视频源 NewFarme 事件为 CameraEventHandler 事件。

由此可见，正是通过此 CameraEventHandler 类事件，实现了层次间信息及时传递，实现实时视频流呈现。

5 团队分工

Table6.团队分工一览

团队成员	主要负责
金林波	负责软件的总体架构设计，编写了视频流访问层和数据逻辑层。视频流访问层把基于.net Framework 1.1 的 Camera Vision 开源底层代码重新编写整理，移植到.net Framework 4.0，并基于其 Mjpeg.dll 和 jpeg.dll，设计开发了对 EasyN IP Camera 定制的底层视频源连接支持库 EasyN.dll。数据逻辑层设计开发了软件的拍照录制异常监测等基本功能，利用定时器实现每秒 25 帧的精确视频录制，异常监测采用前后帧对比算法，同时开发设计了针对 EasyN IP Camera 的 CameraCGI 接口类，基于此实现了云台控制移

	动布防等定制功能。
郭水林	1.完成 UI 层界面设计，合理布局各个模块和组件在窗体的具体位置； 2.设计菜单栏和工具栏中各个功能组件的排布及其实现方式； 3.搭建后台数据库，实现对用户登录信息和添加摄像头信息进行管理； 4.实现对摄像头进行添加、删除、配置等功能的编辑； 5.实现对摄像头视图窗口进行管理，即实现对摄像头进行多页面和全屏模式显示； 6.完成对摄像头进行拍照、录制、定时抓拍和异常检测等功能的编辑； 7.实现对录制视频和照片进行拍照和管理等功能； 8.调试软件，排除在软件运行中出现的 DEGBUG。
高翠芸	1.用户手册的编写； 2.地图程序的设计，主要包括界面的优化，搜索栏的导入，以及经纬度信息的定点显示；