

基于 Directshow 与数字减影技术的视频监控系统^①

Video Monitoring System on Directshow and Digital Subtraction Angiography

周永刚 张有忱 (北京化工大学 机电学院 北京 100029)

摘要: 随着计算机技术的进步,视频监控技术已经在计算机领域得到了发展,但是目前市场上的视频监控设备都比较昂贵,因此不能普遍的应用。本文介绍就是一种性价比很高的监控系统,系统利用常见的 USB 摄像头为视频捕捉设备,通过 Directshow 技术实现计算机对摄像头的控制;通过数字减影技术实现对摄像头捕获到图像的处理。该系统可以有效地监测到监控区域物体位置的变化,从而达到监控报警的目的,而且成本低,使用方便可以被广泛的应用到各个领域。

关键词: Directshow 数字减影 视频监控 VC usb 摄像头

1 引言

视频监控技术是为了满足银行、交通、化工厂等要害部门和社会领域的安防需要而发展起来的一门独立的技术。早期的监控主要基于本地模拟图像,主要由摄像机、视频矩阵、监视器、录像机组成,通过视频线、控制电缆连接,但只能实现监视的效果,不能自动报警。随着计算机多媒体技术与数字图像处理技术的发展,现在的监控系统已经可以实现实时视频监控并自动报警。

本文利用 VC++ 编程,结合 Directshow 与数字减影技术来实现视频监控报警系统。

2 实现原理

本文介绍的监控软件,通过 Directshow 实现对 USB 摄像头视频的捕获;利用数字减影技术实现对图像的处理完成图像对比的功能。

2.1 Directshow 工作原理

Directshow 是微软公司提供的一套在 Windows 平台上进行媒体流处理的多媒体开发包,与 DirectX 开发包一起发布。是微软公司推出的新一代基于 COM 组件对象模型的系统,由许多模块化的软件

组件组成。Directshow 采用多线程多任务的方式,因此我们利用 Directshow 支持多线程的特性,用一个线程实现数据的采集与回放,另一个线程进行图像处理,从而达到视频监控的目的。

在 Directshow 中最基本的概念是过滤器 (Filter)。一个过滤器就是一个 COM 对象,也是一个具有特定功能的模块,它能够对数据流进行操作,如读取、拷贝和写入文件等。Directshow 提供了许多过滤器,大多数过滤器按照功能可以分成三类:源过滤器 (Source Filter)、转换过滤器 (Transform Filter) 和渲染过滤器 (Rendering Filter)。每个过滤器都有引脚 (Pin),相当于输入、输出插头,Filter 与 Filter 通过 Pin 连接构成 Filter Graph 然后与外接设备连接组成一个完整的系统

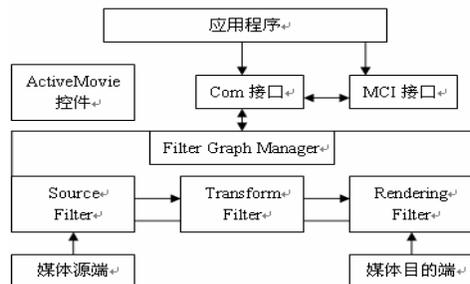


图 1 Directshow 体系结构图

① 收稿时间:2008-08-21

2.1.1 实现 USB 摄像头视频的捕获

图 2 是程序中 Filter 在 GraphEdit 中的调试链路，其中“USB PC Camera-168”是一个视频捕捉 Filter，图中我们也可以看出这是一个带有 USB 接口的数码摄像头，负责采集图像并向下一级 Filter 传送数据；“Video Render” 用来接受上一级传送的数据并以视频方式显示出来。

要想在程序中使用它们就要创建 Filter Graph(过滤器图表)，把 Filter 有机的连接在一起发挥作用。具体步骤如下：

①在 DSControl 类中创建几个接口全局变量

```

IGraphBuilder*      m_pGB;           //
Filter Graph
ICaptureGraphBuilder2* m_pBuilder; //视频
采集过滤器
IBaseFilter*        m_pFCapture;    //摄像头
的 Filter
    
```

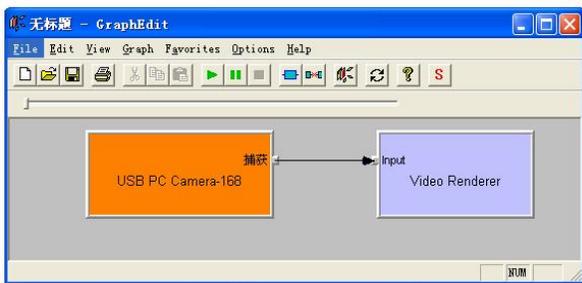


图 2 Filter 在 GraphEdit 中的调试链路

②为了简化 Filter Graph 的构建，DirectShow 提供了辅助构建组件，其中 CLSID_CaptureGraphBuilder2 组件（提供了 ICaptureGraphBuilder2 接口）可以进行视频的采集。可以通过这个接口方便地创建视频采集过滤器图表，然后再将它添加到 IGraphBuilder 图表管理器中(如图 3)。

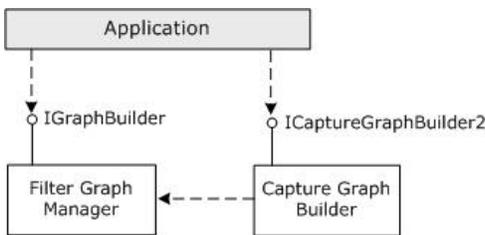


图 3 添加 ICaptureGraphBuilder2

代码如下：

```

// 创建 Filter Graph
hr = CoCreateInstance(CLSID_FilterGraph,
NULL,CLSCTX_INPROC,IID_IGraphBuilder,(void**
)&m_pGB);
if(FAILED(hr)){
return hr;
}
// 创建视频捕捉所需的
CaptureGraphBuilder
hr=CoCreateInstance(CLSID_CaptureGraphBuild
er2,
NULL,CLSCTX_INPROC,IID_ICaptureGraphBuilde
r2,(void**)&m_pBuilder);
if(FAILED(hr)){
return hr;
}
m_pBuilder->SetFiltergraph(m_pGB); // 设置
Graph
    
```

③通过设备枚举方式获得设备友好名称(FriendlyName)从而创建 Video Capture Filter(图 2 “USB PC Camera-168”), 并把它加入到 Filter Graph 中。在构造 Video Capture Filter 前，我们必须列举出系统的所有采集设备，根据枚举找到的结果，把 Video Capture Filter 加入 Filter Graph 中去。步骤如下：

①使用 CoCreateInstance 函数创建一个系统枚举组建对象，并获得 ICreateDevEnum 接口。代码如下：

```

ICreateDevEnum *pCreateDevEnum;
HRESULT hr=CoCreateInstance(CLSID_System
DeviceEnum,NULL,CLSCTX_INPROC_SERVER,IID_
ICreateDevEnum, (void**)&pCreateDevEnum);
    
```

②使用接口方法 ICreateDevEnum : CreateClassEnumerator 为指定的类型目录创建一个枚举器并获得 IEnumMoniker 接口。代码如下：

```

hr=pCreateDevEnum->CreateClassEnumerator
(CLSID_VideoInputDeviceCategory,&pEm, 0);
CComPtr<IEnumMoniker> pEm;
    
```

③使用接口方法 IEnumMoniker: Next 枚举指定类型目录下所有的设备标识。代码如下：

```
while(hr = pEm->Next(1, &pM, &cFetched),
hr==S_OK)
{……// 具体代码省略}
```

④调用 `IMoniker: BindToObject` 将设备标识绑定成一个 `Directshow Filter`, 随后调用 `IFilterGraph: AddFilter` 加入到 `FilterGraph` 中就可以完成设备的枚举工作。代码如下:

```
pM->BindToObject(0, 0, IID_IBaseFilter,
(void**)pFilter);
hr = m_pGB->AddFilter(m_pBF, L"Capture
Filter");
```

2.2 数字图像处理

数字图像处理 (`Digital Image Processing`) 又称为计算机图像处理, 它是指将图像信号转换成数字信号并利用计算机对其进行处理的过程。数字图像处理是一门交叉学科, 它涉及到数学、光学、信息科学、工程技术、计算机技术等众多学科。因此图像处理技术在许多应用领域受到广泛重视并取得了重大的开拓性成就, 属于这些领域的有航空航天、生物医学工程、工业检测、机器人视觉、公安司法、军事制导、文化艺术等, 使图像处理成为一门引人注目、前景远大的新型学科。

图像处理包括图像基本运算、图像变换、图像增强、图像复原等技术。本文介绍的是图像处理中基本运算中点运算的减法运算。图像相减常用于检测变化及运动的物体, 图像相减运算又称为图像差分运算。差分方法可以分为可控制环境下的简单差分法和基于背景模型的差分法。我们使用的基于背景模型的差分法即减影技术。减影技术又称为差影法, 是指将同一景物在不同时间拍摄的图像或同一景物在不同波段的图像相减。差值图像提供了图像间的差异信息, 能用于指导动态监测、运动目标的检测和跟踪、图像背景的消除及目标识别等。

2.2.1 数字减影函数

软件要实现的功能就是通过设定一个背景图片, 通过下一帧的图像信息与背景信息相减比较得出新图像是否有改变, 由此来达到视频监控的功能。具体实现步骤如下:

①定义三个指针用于存储图像信息分别为:

`BYTE* pIn` 新图的数据区指针, `BYTE* pReduce` 背景数据指针, `BYTE* pOut` 相减后图像数据指针。

②存储背景。通过 `Directshow` 连接摄像头后, 调用 `memcpy ()` 函数存储背景信息。代码如下:

```
if(pDNS->m_bGetBackBmp){
// m_bGetBackBmp 用于判断是否获取背景图
memcpy(pReduce,pb,BufferLen); // BufferLen
为数据区长度
pDNS->m_bGetBackBmp = FALSE;
}
```

③进行图像相减。代码如下:

```
if(!pReduce) return;//判断如果没有背景图就不处理
int differentPoint = 0; //记录不同点的数目
for(int i = 0; i < m_bmpBufferLen; i++){
pOut[i] = abs(pIn[i] - pReduce[i]); // 新图信息与背景信息相减, 取差值的绝对值
if(pOut[i] > 32){ // 32 为灵敏系数此值因摄像头而异, 与噪音有关, 相差大于 32 就认为是不同的点
differentPoint ++; // 不同点增加
pOut[i] = pIn[i]; // 把不同点赋回它的颜色
}
```



图 4 软件获取背景

如图 4 所示, 将摄像头安装好后, 按下“开始摄像”按钮, 系统通过 `Directshow` 连接摄像头并取得摄像头所捕获的图片信息显示在监控画面中; 确定所要的背景后, 按下“获取背景图”按钮得到所要的背景。此时由于监控画面与背景图信息相同, 因此减影画面显示为无像素的黑色图像。 (下转第 67 页)



图 5 系统监控区域发生变化

如图 5 所示, 拿走监视中的物体后, 新的图片信息与背景信息相减后将不同的信息显示在减影画面中, 而与背景相同的信息依然显示为黑色。通过设定与背景图差异点的上限, 当监控的图像超过这个上限时, 系统就报警, 记录发生状况的时间。

3 结束语

通过实验, 可以证明通过利用 Directshow 与数

字减影技术可以实现视频监控报警的作用。此系统适用于监视环境发生改变的情况下, 通过软件将所要监视的环境做为背景, 当监控的环境发生变化时 (如有人的移动或物体位置的改变), 系统就会自动报警。这与市面上出售的监控系统相比有很高的性价比, 并且软件还处于开发阶段, 通过后期不断完善还可以实现更多的功能。

参考文献

- 1 陆其明. DirectShow 开发指南. 北京: 清华大学出版社, 2003. 135 - 156.
- 2 陈天华. 数字图像处理. 北京: 清华大学出版社, 2007. 208 - 225.
- 3 蔡龙华, 何道清, 李永乐. 基于 DirectShow 技术的视频捕获. 计算机与现代化, 2003, (8): 81 - 84.
- 4 袁莉茹. 图像减影技术在自动报靶系统中的应用. 成都信息工程学院学报, 2006, (3): 370 - 372.