

# 基于 DirectShow 的实时视频信息采集与压缩

赵德志, 吴 洁, 丁秋林

(南京航空航天大学, 江苏 南京 210016)

**摘要:** 论述采用 DirectShow 技术进行实时视频信息的采集和压缩存储的方法, 并给出详细的设计思路。设计中视频捕捉采用 WDM 驱动捕捉, 提高视频捕捉的效率和通用性; 采用基于内容的 MPEG-4 视频编码标准进行压缩, 得到较高的压缩比和压缩质量。该技术已经成功地应用到笔者参与开发的视频会议系统中, 有效提高了采集速度和视频质量。

**关键词:** DirectShow; 视频采集; 视频压缩; 过滤器

中图分类号: TN919.8

文献标识码: A

文章编号: 1005-3751(2006)01-0187-03

## Real-time Video Capture and Compression Based on DirectShow

ZHAO De-zhi, WU Jie, DING Qiu-lin

(Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

**Abstract:** Describes a method of using DirectShow to implement the video captures and compression, and explains it with particular designing thoughts. It adopts WDM as the video capture method, which can enhance the efficiency and generality of the video capture. It utilized MPEG-4, one of video coding standard based on content, to encode the video stream, which can result in higher encoding rate and better encoding quality. This technology has been used at Video Conference System, improving the speed and quality for capture greatly.

**Key words:** DirectShow; video capture; video compression; filter

### 0 引言

在信息时代, 网络和多媒体技术得到迅速发展, 并在视频会议系统、远程教学等系统中得到广泛应用。视频信息的采集与压缩是多媒体开发应用中的关键前提。视频采集是将一个视频流数字化, 然后存储在存储介质上或通过网络发送, 由于视频的数据量比较大, 因此需要先对采集的视频进行压缩。

以往进行视频采集、压缩主要通过微软的 VFW (Video For Windows) SDK 或采集卡提供的 SDK, 现在微软公司在 ActiveMovie 和 VFW 的基础上推出了新一代基于 COM 的多媒体开发包——DirectShow, 由于 DirectShow 支持更多的数据格式如 AVI、ASF、MPEG、DV 等和高品质的视频流采集和回放, 为视频的采集压缩提供了有效的途径。

### 1 DirectShow 系统结构及开发技术要点

DirectShow 的核心是被称之为过滤器(Filter)的插件

收稿日期: 2005-04-29

**作者简介:** 赵德志(1979—), 男, 辽宁普兰店人, 硕士研究生, 研究方向为系统集成与企业信息化; 吴 洁, 副教授, 研究方向为系统集成与企业信息化; 丁秋林, 教授, 博士生导师, 研究方向为数字化设计、制造与集成、企业信息化、系统集成/CIM/CAD/CAM。

式模块系统, 它被装配在过滤器图(filter graph)中, 用于管理在过滤器内部和过滤器之间处理的音视频数据流。过滤器图表管理器负责把这些过滤器连接起来进行管理。

DirectShow 系统结构图如图 1 所示。

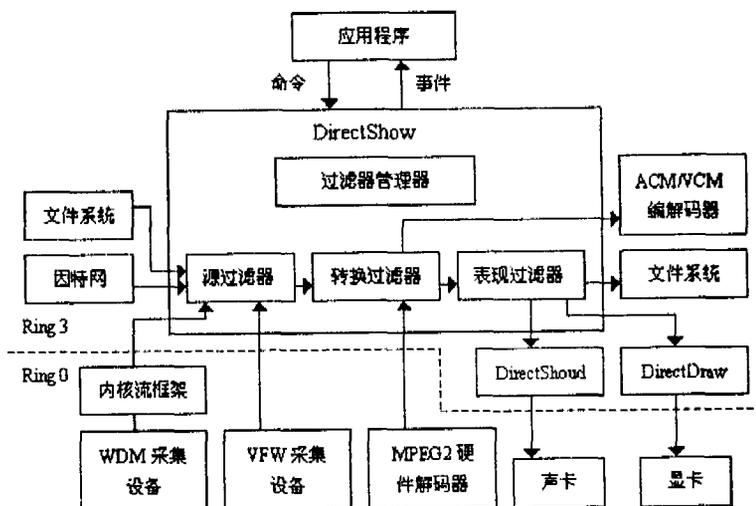


图 1 DirectShow 系统结构图

图中央最大的一块即是 DirectShow 系统, DirectShow 系统位于应用层中<sup>[1]</sup>。它使用一种叫 Filter Graph 的模型来管理整个数据流的处理过程; 参与数据处理的各个功能模块叫做 Filter; 各个 Filter 在 Filter Graph 中按一定的顺序连接成一条“流水线”协同工作。过滤器可分为 3 类, 即源过滤器(Source Filters)、转换过滤器(Transform Filters)

和表现过滤器 (Rendering Filters)。源过滤器用来从数据源获取数据,并将数据传入过滤器图表;转换过滤器用来获取、处理和传递数据;表现过滤器用来表现数据,典型地表现到硬件设备,如将视频输出到视频卡中。

DirectShow 开发完全建立在 COM 组件技术基础上,COM 组件是关于对象如何创建和消亡以及对象间相互通信的二进制标准。DirectShow 与 COM 紧密相连,它所有的功能都由 COM 接口 (Interface) 来构造和实现,过滤器暴露为 IbaseFilter 接口;过滤器引脚暴露为 Ipin 接口;过滤器图管理器暴露为 IGraphBuilder 接口等<sup>[2]</sup>。

典型的 DirectShow 应用程序开发遵循了几个步骤:

(1) 创建一个 Filter Graph Manager 实例,它通常通过 CoCreateInstance API 函数实现;

(2) 通过 Filter Graph Manager 创建一个 Filter Graph;

(3) 对 Filter Graph 进行控制及对事件做出响应处理。

下面以 RGB Pro-2 采集卡为例,介绍 DirectShow 技术在实时的视频采集与压缩中的应用。

## 2 视频采集的实现

可以用 DirectX SDK 自带的工具 GraphEdit 方便直观地来设计和测试过滤器图,笔者设计的视频采集过滤器图如图 2 所示。

因为 RGB Pro-2 Capture filter 没有预览引脚,因此要连接一个 Smart Tee Filter,该过滤器的作用是把数据分成两个数据流,一个数据采集流和一个预览流。

下面可以根据设计的过滤器图编程实现<sup>[3,4]</sup>。

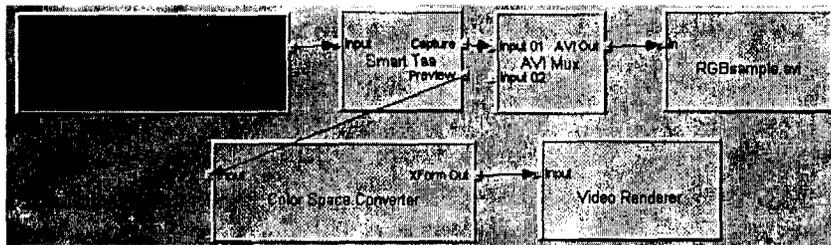


图 2 实时视频采集与预览的过滤器图

### 2.1 创建基本组件

```
IFilterGraph2 * g_pGraph=NULL;
```

```
ICaptureGraphBuilder2 * g_pCapture=NULL;
```

```
.....
```

```
//创建过滤器图管理器
```

```
CoCreateInstance (CLSID_FilterGraph, NULL, CLSCTX_INPROC, IID_IFilterGraph2, (void **) &g_pGraph);
```

```
//创建采集图构建器, ICaptureGraphBuilder2 接口取代早期版本中的 ICaptureGraphBuilder, 以支持采集更多的媒体类型
```

```
CoCreateInstance (CLSID_CaptureGraphBuilder2, NULL, CLSCTX_INPROC, IID_ICaptureGraphBuilder2, (void **) &g_pCapture);
```

```
//把建立好的过滤器图选入采集图构建器, 如果不调用 SetFiltergraph 方法, 则采集图构建器会自动建立一个空的过滤器图
```

```
g_pCapture->SetFiltergraph(g_pGraph);
```

### 2.2 枚举采集设备

采集设备在 DirectShow 框架下也是以过滤器的形式存在的<sup>[5]</sup>, WDM 驱动模型的采集设备使用 WDM Video Capture Filter (实现文件为 kswdmcap.ax), VFW 驱动模型的采集设备使用 VFW Captuer Filter (实现文件为 qcapi.dll)。应用程序只要枚举特定的类型目录, 就能知道系统中安装有多少个, 以及何种类型的采集设备。视频采集设备注册在 Video Capture Sources 目录下, 该类型目录的 CLSID 为 CLSID\_VideoInputDeviceCategory, 下面是枚举采集设备部分的实现:

```
//创建系统设备枚举器
```

```
CComPtr <ICreateDevEnum> pDevEnum=NULL;
```

```
CoCreateInstance (CLSID_SystemDeviceEnum, NULL, CLSCTX_INPROC, IID_ICreateDevEnum, (void **) &pDevEnum);
```

```
//创建视频采集设备枚举器
```

```
CComPtr <IEnumMoniker> pClassEnum=NULL;
```

```
pDevEnum->CreateClassEnumerator (CLSID_VideoInputDeviceCategory, &pClassEnum, 0);
```

```
ULONG cFetched;
```

```
IMoniker * pMoniker=NULL;
```

```
IBaseFilter * pSrc=NULL;
```

```
//采用设备列表中的第一个视频采集设备
```

```
if(S_OK=(pClassEnum->Next(1, &pMoniker, &cFetched)))
```

```
{
```

```
LPOLESTR strMonikerName=0;
```

```
//取得设备显示名
```

```
pMoniker->GetDisplayName (NULL, NULL, &strMonikerName);
```

```
.....
```

```
IBindCtx * pContext=0;
```

```
hr=CreateBindCtx(0, &pContext);
```

```
//根据取得的 Moniker 将源过滤器添加到过滤器图中
```

```
//AddSourceFilterForMoniker 方法是 Di-
```

```
rectX 9.0 中用来取代 DirectX 8.0 中的 IGraphBuilder::RenderFile()方法
```

```
hr=g_pGraph->AddSourceFilterForMoniker(pMoniker, pContext, strMonikerName, &pSrc);
```

```
}
```

### 2.3 视频采集与预览

DirectShow 采用 WDM 方式进行视频捕捉, 因此在实现视频回放功能时, 它直接把捕捉到的视频流从采集设备送到显示设备, 数据流在核心模式下传输, 减少了到用户模式的切换时间, 使视频的实时性得到保证。

```
//用 RenderStream 方法将视频源过滤器和预览连接起来
```

```
hr=g_pCapture->RenderStream (&PIN_CATEGORY_PREVIEW, &MEDIATYPE_Video, pBaseFilter, NULL, NULL);
```

```
//设置将采集数据的输出文件
```

```
hr=g_pCapture->SetOutputFileName (&MEDIASUBTYPE_
```

```

Avi,L"F:\RGBSample.avi", &pMux,NULL);
//用 RenderStream 将视频源过滤器和文件过滤器连接起来
hr = g_ pCapture -> RenderStream (&PIN_ CATEGORY_ CAP-
TURE, &MEDIATYPE_ Video, pSrcFilter, NULL, pMux);

```

如果既指定了 PIN\_ CATEGORY\_ CAPTURE, 又指定了 PIN\_ CATEGORY\_ PREVIEW, RenderStream 方法会自动插入一个 Smart Tee 过滤器将数据分流。

各个过滤器连接完毕后, 就可以调用 IMediaControl:: Run() 方法<sup>[4]</sup>进行视频的采集和预览了。

#### 2.4 控制采集过程

过滤器图的 IMediaControl 接口提供方法来运行、终止和暂停整个图, 但是如果图中有多个数据流, 想独立地控制它们则要使用 ICaptureGraphBuilder2:: ControlStream 方法<sup>[4]</sup>, 要采集流运行一段时间停止, 而预览流仍在运行, 代码如下:

```

IMediaControl * pControl;
REFERENCE_ TIME rtStart = 20000000, rtStop = 50000000;
//采集流在 2 秒时启动, 在 5 秒时停止
pBuilder -> ControlStream (&PIN_ CATEGORY_ CAPTURE,
&MEDIATYPE_ Video, pSrc, &rtStart, &rtStop, 0, 0);
pGraph -> QueryInterface (IID_ IMediaControl, (void * *)
&pControl);
pControl->Run();

```

### 3 视频压缩

如果视频数据都不经过压缩而直接保存为 AVI 文件, 那么最后的尺寸将是很大的。所以通常应该先考虑对视频内容进行压缩, 然后再保存到文件中。要在 DirectShow 体系结构中实现视频压缩, 可在过滤器图表中添加用来实现视频编码的过滤器<sup>[5]</sup>, 视频编码的压缩过滤器可以自己编写, 也可以购买或免费获得, 通过不同的编码过滤器可以得到不同的格式和压缩质量的文件, 笔者采用的是 Microsoft 提供的 Microsoft MPEG-4 Video Codec 编码器, 只要装入用户系统, 就可以在 DirectShow 中以过滤器的形式使用。对于文中所建立的过滤器图, 需要把 MPEG-4 Video Codec 过滤器添加到 Smart Tee 过滤器和 AVI Mux 过滤器之间。首先要通过系统设备枚举来找到该过滤器, 在目录 CLSID\_ VideoCompressorCategory 创建枚举器, 根据查找到的编码过滤器的 FriendlyName 来选择

“Microsoft MPEG-4 VKI Codec V1”过滤器, 将该过滤器加入到过滤器图中。

```

hr = g_ pGraph-> AddFilter (pCompressFilter, L“MS
MPEG-4 VKI Codec”);

```

最后调用 ICaptureGraphBuilder2 的 RenderStream 方法或者 IGraphBuilder 的 Connect 方法将采集过滤器、MPEG-4 编码过滤器、采集数据写入过滤器连接起来

```

hr = g_ pCapture-> RenderStream (&PIN_ CATEGO-
RY_ CAPTURE, &MEDIATYPE_ Video, pSrcFilter,
pCompressFilter, pMux);

```

DirectShow 系统是一个开放的系统, 要得到其他的媒体格式, 只需要安装该编码器将它作为过滤器加入到过滤器图中。例如要将采集到的视频保存为 MPEG 文件, 则需要用到 MPEG 的音视频编/解码过滤器, DirectShow 本身不提供这些过滤器, 因此必须自行开发。这些编/解码过滤器要用在实时采集场合时, 要求它们有较高的性能, 如果达不到实时处理要求, 采集时会产生丢帧。在项目应用中应用 Divx5 和微软的 MPEG-4 的编码器基本能取得较好的压缩比率的回放质量。

### 4 结束语

介绍了 DirectShow 技术在实时视频采集与压缩上的应用, 并将该技术应用到视频会议系统中, 取得良好的效果。在 Windows 平台下, 微软的 DirectShow 技术是开发多媒体应用程序快捷有效的方法, 现在大多数的视频采集卡都被 DirectShow 体系结构支持, 文中的代码稍加修改即可适用。

#### 参考文献:

- [1] Microsoft Corporation. Microsoft DirectX SDK 9.0 Document [M/CD]. US: Microsoft Corporation, 2004.
- [2] 李 岚. 基于 AVI 的流媒体多播技术研究[D]. 武汉: 华中师范大学, 2002. 39-51.
- [3] Rogerson D. COM 技术内幕——微软组件对象模型[M]. 北京: 清华大学出版社, 1999.
- [4] Microsoft Corporation. MSDN [EB/OL]. <http://msdn.microsoft.com/library/default.asp>, 2004.
- [5] 陆其明. DirectShow 实务精选[M]. 北京: 科学出版社, 2004.

(上接第 186 页)

系统的重要组成部分, 使整个远程教育系统的教材资源得以科学、规范的管理, 有效提高了管理工作的质量与效率。系统目前已经开发完成, 在多个客户端接入的情况下, 都能够迅速地完成任务, 得到了很好的效果。

#### 参考文献:

- [1] 丁 晶, 王行言. 基于 Servlet 和 JSP 技术的网络辅助教学系统的设计与实现[J]. 计算机工程, 2001, 27(11): 175-178.

- [2] 孙 莹, 许俊华, 张 毅, 等. MVC 编程模型在 Web 程序中的应用及 Java 实现[J]. 计算机工程与应用, 2001(17): 160-163.
- [3] 董卫军, 周警伟. MVC 在 Web 系统中的模式与应用[J]. 计算机仿真, 2003, 20(12): 111-114.
- [4] 吕西红. 高校教材管理信息系统的设计与实现[D]. 长沙: 国防科技大学, 2002.
- [5] 陈大鑫, 杨路明. MVC 设计模式在远程考试系统中的应用[J]. 微机发展, 2004, 14(10): 73-75.