

# 基于 DirectShow 的多线程视频采集系统

熊超, 陆起涌, 田小芳

(复旦大学电子工程系, 上海 200433)

**摘要:** 立体视觉系统是拟人机器人重要组成部分。为满足立体视觉系统的特殊要求, 提出了基于 DirectShow 技术的多线程视频采集系统。该系统实现了同步采集 2 个 USB 摄像头数据, 具有结构简单、实现方便的特点, 是立体视觉开发的良好平台。

**关键词:** DirectShow; 视频采集; 过滤器; 过滤器图管理器; 多线程

## Multi-thread Video Capture System Based on DirectShow

XIONG Chao, LU Qiyong, TIAN Xiaofang

(Department of Electronic Engineering, Fudan University, Shanghai 200433)

**【Abstract】** Stereo vision system is a vital part of the anthropomorphic robot. To satisfy the special requirement of the stereo vision system, a multi-thread video capture system based on DirectShow is proposed. The system is good development platform, which is simple and can be used to synchronically capture video streams of two same type of USB cameras.

**【Key words】** DirectShow; Video capture; Filter; Filter graph manager; Multi-thread

立体视觉系统是拟人机器人的重要组成部分, 是近年来的一个研究热点。如何同步采集机器人“左右眼”的图像信息, 是实现机器人拟人视觉的前提条件。目前常用视频采集技术主要有以下 3 种: 基于 VFW (Video for Windows) 技术的视频采集, 基于 DirectShow 技术的视频采集, 基于视频采集卡厂商提供 SDK 的视频采集技术<sup>[1,2]</sup>。

本文利用 DirectShow 技术, 结合 Windows 多线程编程, 实现了同步读取 2 个通用串行总线 (Universal Serial Bus, USB) 接口数字摄像头, 在 PC 环境下构建了一个实用的立体视觉研究平台。该方法实现简单, 无需另配视频采集卡, 通过普通的 USB 摄像头即可搭建多视频同步采集系统。同时, 该系统还充分利用了 DirectShow 中模块化程序开发的特点, 具有良好的二次开发和升级能力, 可以作为机器人立体视觉系统的研究平台。

### 1 DirectShow 基本原理

#### 1.1 DirectShow 简介

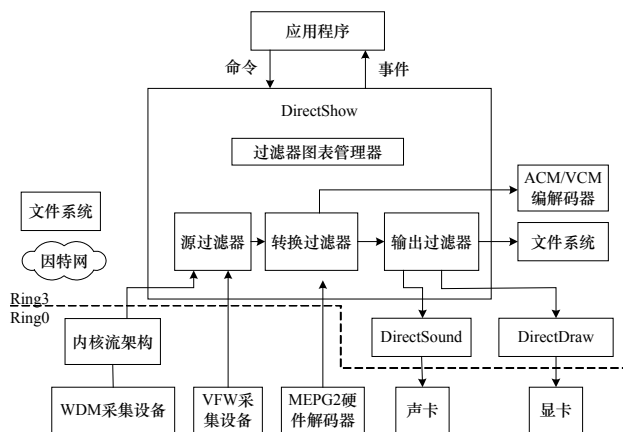


图 1 DirectShow 系统

DirectShow 是微软 DirectX 软件开发包的一个成员, 为在

Windows 平台上处理各种格式的媒体文件的回放、音视频采集等高性能要求的多媒体应用, 提供了完整的解决方案<sup>[3]</sup>。DirectShow 是基于 COM 的应用系统, 其核心是被称为过滤器的插件模块。如图 1 所示, 中央部分是 DirectShow 系统, 位于应用层中。过滤器在过滤器图表中配置, 通过过滤器图表管理器检查过滤器之间的连接, 并且控制过滤器之间的数据流动。

过滤器可以大致分为 3 类: 源过滤器, 负责获取数据; 转换过滤器, 负责数据的格式转换, 然后将数据继续往下传; 输出过滤器, 负责数据的最终去向, 多媒体输出或输出到文件存储。

视频采集设备根据采用的驱动类型不同可以分为 2 种: VFW (Video for Windows) 和 WDM (Windows Driver Model), WDM 是 VFW 的替代模型。WDM 可以通过 COM 接口 IAMCrossbar、IAMStreamConfig 和 IAMVideoProcAmp 编程实现输入端子选择, 显示质量控制, 而 VFW 只能将驱动程序定制的设置对话框直接显示给用户选择, 不适合做实时连续的视频采集。DirectShow 支持 WDM 驱动模式。

采集设备有友好名字和显示名字 2 个基本属性, 友好名字形式如“USB PC CAM-168”, 用于显示给用户识别设备, 同种型号的友好名字是一样的; 显示名字为字符串, 用户很难理解, 但显示名字唯一地标识了每一个设备。

#### 1.2 DirectShow 一般开发过程

开发 DirectShow 应用程序, 一般分为 3 个部分, 如图 2 所示<sup>[4]</sup>。

(1) 创建一个过滤器图管理器; (2) 根据实际的应用, 创建

**作者简介:** 熊超 (1981—), 男, 硕士生, 主研方向: 多媒体技术, 计算机视觉, 嵌入式系统; 陆起涌, 硕士、副教授; 田小芳, 硕士生

**收稿日期:** 2005-10-22 **E-mail:** xch@fudannac.org

一条完整的过滤器链路；(3)调用过滤器图管理器或过滤器上的各个接口方法进行控制，并完成过滤器图管理器与应用程序的事件交互。

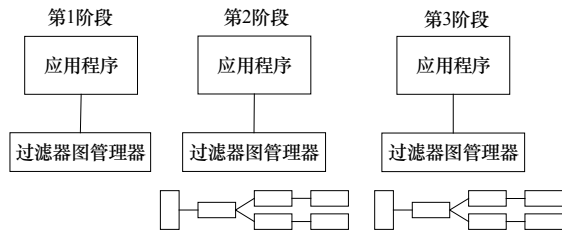


图 2 DirectShow 应用程序一般开发过程

为了方便验证构建的过滤器链路是否能够实现设计的目标，可以使用微软公司提供的 DirectX 工具集中的 GraphEdit 查看程序中过滤器的连接情况。使用 GraphEdit，在程序中创建过滤器图管理器之后，应该立即将其注册到活动对象表。

本文提出的基于 DirectShow 的多线程视频采集系统符合上述的开发过程，系统整体结构将在 2.1 节介绍，具体程序实现将在 2.2 节中说明。

## 2 多线程视频采集系统程序实现

### 2.1 系统简介

系统实现如图 3 所示。

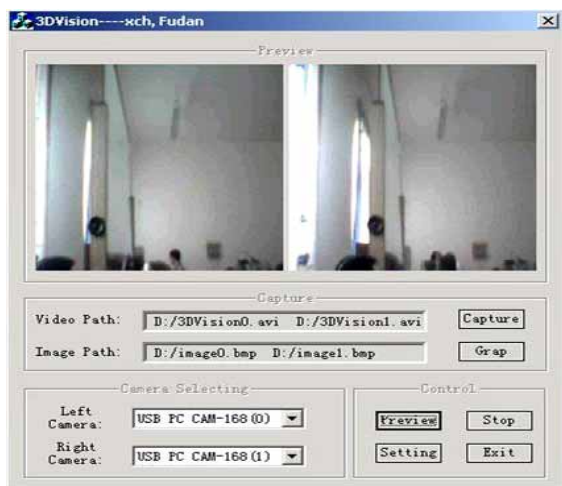


图 3 基于 DirectShow 的多线程视频采集

本系统目标是用于辅助机器人双目视觉系统的开发，必须选用相同型号的 2 个摄像头，对于不同型号的摄像头其方法相同。这 2 个摄像头具有相同的友好名字，因此本系统根据摄像头的现实名字构建源过滤器，同时获得对应的友好名字，并在友好名字末尾添加“(0)”、“(1)”，显示给用户进行机器人“左右眼”的设置。

2 个摄像头对应的显示名字分别为 @device:pnp:\usb#vid\_0c45&pid\_613c#6&199e80eb&0&3#{65e8773d-8f56-11d0-a3b9-00a0c9223196}\global 和 @device:pnp:\usb#vid\_0c45&pid\_6130#6&199e80eb&0&3#{65e8773d-8f56-11d0-a3b9-00a0c9223196}\global。

源过滤器创建完成后，分别加入到相应过滤器图管理器 mGraph0、mGraph1，并创建过滤器图，如图 4 所示。由于需要同步采集视频，本文提出利用 Windows 的多线程编程技术，在应用程序中创建两个子线程分别调用 mGraph0、mGraph1 上的各种接口和控制功能，可以实现视频预览、停止，视频采集，图像抓取等功能。

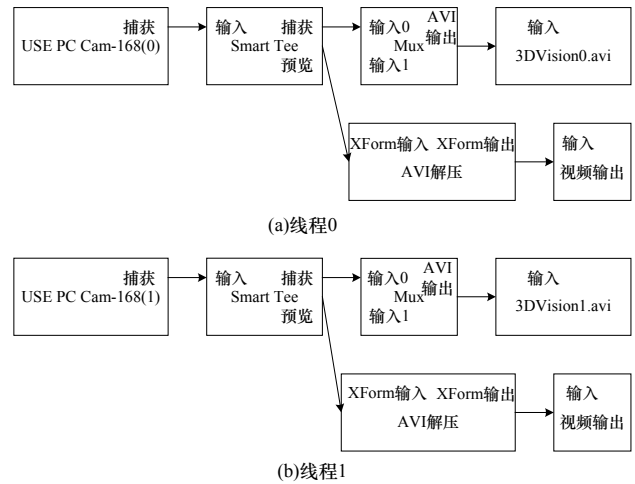


图 4 过滤器

### 2.2 系统程序实现

进行 DirectShow 应用程序开发首先需要配置开发环境，应用程序应该包含 DShow.h 头文件，链接库文件 Strmiids.lib 和 Quartz.lib。

图 5 给出了基于 DirectShow 的多线程视频采集系统的程序流程。

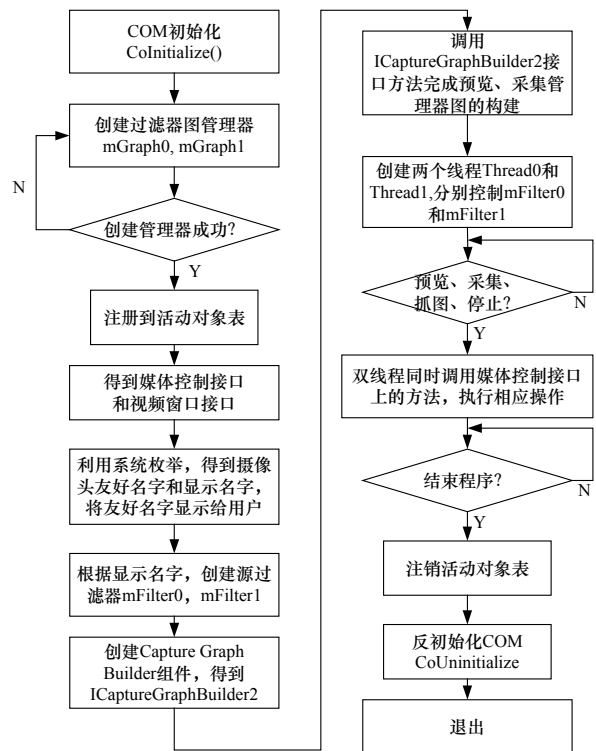


图 5 系统程序流程

(1)在头文件中定义下面的类型变量：

```

IGraphBuilder * mGraph0, mGraph1; //过滤器图管理器
ICaptureGraphBuilder2 * mBuilder0, * mBuilder1;
//Capture Graph Builder 组件接口
IBaseFilter * mFilter0, * mFilter1; //采集设备过滤器
IBaseFilter * pMux0, * pMux1; //AVI 合成接口
IVideoWindow * pVW0, * pVW1; //视频窗口接口
IMediaControl * pMC0, * pMC1; //媒体控制接口

```

(2)DirectShow 应用程序是一种 COM 客户程序，使用 COM 函数前需要进行初始化。初始化 COM 调用 CoInitialize()，反初始化 COM 调用 CoUninitialize()。

(3)调用 COM 函数创建过滤器图管理器:

```
CoCreateInstance(CLSID_FilterGraph,NULL,CLSCTX_INPROC_SERVER, IID_IGraphBuilder, (void **)&mGraph);
```

(4)通过过滤器图管理器得到媒体控制接口和视频窗口接口,只需要调用 mGraph0, mGraph1 上的接口方法 QueryInterface();

(5)利用系统枚举获得摄像头的属性(友好名字、显示名字等),并根据显示名字创建源过滤器 mFilter0, mFilter1。

1)创建系统枚举器组件对象,获得 ICreateDevEnum 接口:

```
CoCreateInstance(CLSID_SystemDeviceEnum,NULL,CLSCTX_ALL, IID_ICreateDevEnum, (void **)&enumHardware);
```

2)使用接口方法 ICreateDevEnum::CreateClassEnumerator 为视频输入类型目录创建一个枚举器,并获得 IEnumMoniker 接口:

```
enumHardware->CreateClassEnumerator(CLSID_VideoInputDeviceCategory,&enumMoniker,0);
```

3)调用 IEnumMoniker 接口上的方法 Next 枚举指定目录下所有设备:

```
enumMoniker->Next(1,&moniker2,&fetched);
```

4)调用 IMoniker::BindToStorage 之后,可以得到友好名字和显示名字。

5)调用 IMoniker::BindToObject 将设备生成过滤器,再调用 IFilterGraph::AddFilter,将其加入到过滤器图中。

(6)构建预览和采集过滤器图。这部分工作可以通过 ICaptureGraphBuilder2 接口完成。首先得到 ICaptureGraphBuilder2 接口 mBuilder0, mBuilder1:

```
CoCreateInstance(CLSID_CaptureGraphBuilder2,NULL,CLSCTX_INPROC_SERVER, IID_ICaptureGraphBuilder2, (void **)&mBuilder);
```

设置过滤器图管理器对象指针:

```
mBuilder->SetFiltergraph(mGraph);
```

然后就可以创建预览和采集过滤器图:

```
mBuilder->RenderStream(&PIN_CATEGORY_PREVIEW,&MEDIATYPE_Video,mFilter0, NULL, NULL);
```

```
mBuilder->SetOutputFileName(&MEDIASUBTYPE_Avi,L"D:\\3DVision0.avi",&pMux0, NULL);
```

```
mBuilder->RenderStream(&PIN_CATEGORY_CAPTURE,&MEDIATYPE_Video,mFilter0,NULL, pMux0);
```

(7)调用函数 AfxBeginThread(), 创建 2 个线程 Thread0 和 Thread1, 并给予相应的定义, 分别管理 mGraph0 和 mGraph1。

(8)适当调用 pMC 接口上的方法, 可以方便地控制媒体, 例如 pMC->Run(); 使先前创建好的过滤器图运行, 同时进行预览和采集。

### 3 结束语

本文提出了结合 DirectShow 和多线程技术, 实现了双摄像头的同步读取, 构建了一个机器人立体视觉研究的平台, 较好地解决了机器人“双目”同步采集数据的问题。

本系统实现简单, 无需视频采集卡等设备, 成本节约, 并具有良好的二次开发和升级能力。

#### 参考文献

- 1 蔡龙华, 何道清, 李永乐. 基于 DirectShow 的视频捕获[J]. 计算机与现代化, 2003, (8): 81-85.
- 2 朱永光, 孙正顺, 赵南元. 基于 C++ 的视频采集程序设计及时钟器的使用[J]. 计算机工程与应用, 2002, 38(24): 142-143.
- 3 陆其明. DirectShow 开发指南[M]. 北京: 清华大学出版社, 2003.
- 4 Microsoft Corporation. Microsoft DirectX 9.0 SDK Document[M]. Microsoft Corporation, 2002.

(上接第 206 页)

### 3.3 语义块识别算法

用户输入的句子可以表示为正规式  $S(cS)^*$ , 即至少有一个语义块  $S$ , 若干不同语义块之间通过连接词  $c$  进行连接。语义块识别的任务就是将这些根边节点全部分离出来, 将复杂句子分成若干个单语句, 在后面的步骤中分别处理。

**定义 2** 在语法分析树中, 形如  $[i, j, S \rightarrow a \cdot]$  的边称为语义块根边节点, 以它为根的节点形成一个语义块  $S$ 。其中, 产生式左边为  $S$ , 右边的当前匹配点必须在末尾, 即待匹配项为空。一颗语法分析树包括一个或多个这样的语义块根边节点。如果一个语义块根边节点的子节点中不再包括语义块根边节点, 那么该语义块根边节点称为最小语义块根边节点。

根据前面语义分析得出的语法分析树, 通过对树的遍历可以将所有的最小语义块根边节点提取出来。算法描述如下:

```
public Edge[] GetMeanPart(Edge head)
{Edge[] retval;
if (head is of [i,j,S→a·] and none of children is of [i,j,S→a·])
//如果是最小语义块根边节点则加入结果集
    retval.add( head);
    return retval;
}
for each children e of head
    retval.add( getMeanPart(e) );
return retval;
}
```

将该算法中得到的若干个语义块根边节点分别使用槽

和值识别算法就完成了对用户多语义块自然语句的识别和数据信息提取。

### 4 总结和展望

根据以上算法成功地用 Java 实现了自然语言处理的语法分析, 改善了传统模式匹配方法和将规则编入程序方法的弱点。在实现过程中充分考虑了系统的可扩展性和可维护性, 测试结果证明该系统是成功的。

将来的研究方向是: (1)对算法的改进, 考虑使用机器学习的方法对系统进行扩展, 使得系统可以自己维护字典和词性概率矩阵, 包括新词学习等。(2)进一步对用户语言习惯进行研究和分析, 完善自然语言处理算法。

#### 参考文献

- 1 McTear M F. Spoken Dialogue Technology: Enabling the Conversational User Interface[J]. ACM Computing Surveys, 2002, 34(1): 90-169.
- 2 Wallace R S. The Anatomy of ALICE[EB/OL]. <http://www.alicebot.org/anatomy.html>, 2005.
- 3 Earley J. An Efficient Context-free Parsing Algorithm[J]. Communications of the ACM, 1970, 13(2): 94-102.
- 4 Andrew M, Meng G. Question Answering[R]. Department of Computer Science, University of Sheffield, 2002.
- 5 Cutting D, Kupiec J, Pedersen J. A Practical Part-of-speech Tagger[C]. Proceedings of the 3th Conference on Applied Natural Language Processing, USA, 1992: 133-140.