

基于 Visual C++ 的视频流捕捉方法

冯茂岩

(江苏海事职业技术学院 南京 210070)

摘要 本文首先介绍了微软公司推出的关于数字视频的一个软件包—VFW 和实用于大容量动态图像的 AVI 格式。其次,重点介绍了 VFW 技术中 AVICap 视频捕捉模块的技术原理,并给出了一个视频流捕捉的算法。

关键词 VFW, 音频视频交叉, 视频流捕捉

A Method of Capturing Video Stream Based on Visual C++

FENG Mao-Yan

(Information Engineering Department, Jinagsu Maritime Institute, Nanjing 210070)

Abstract This paper firstly introduces a software kit about digital video—VFW, provided by Microsoft Corporation and AVI file format that fits for huge capacity of dynamic image. Then, it introduces the technical principle of AVICap video capture block in VFW technology, and an algorithm on the video stream capturing is put forward.

Keywords Video for windows, Audio/video interleaved, Video stream sapture

1 引言

随着计算机和多媒体技术的发展,视频技术由黑白到彩色,再到今天的高清晰度数字视频,具有多媒体信息处理能力的计算机已经越来越普遍。在当今十分流行的桌面视频会议、可视电话以及远程视频监控系统等多媒体应用中,获得数字视频是一个关键前提。Video For Windows(简称 VFW)的出现,给视频捕捉编程带来了很大的方便。它能使应用程序数字化并播放从传统模拟视频源得到的视频剪辑。为解决数字视频数据量大的问题,需要对数据进行压缩,于是在本文中引进了 Audio/Video Interleaved(简称 AVI)文件标准。该文件格式实用于大容量的动态图像,可进行帧索引和域索引。

本文在介绍 VFW 技术中的 AVICap 视频捕捉模块的技术原理的基础上,提出了一个视频流捕捉的算法,有效地解决了获取数字视频的问题。

2 VFW 及 AVI 介绍

VFW 是 Microsoft 公司在 1992 年推出的关于数字视频的一个软件包。它能够使用应用程序数字化并播放从传统模拟视频源得到的视频剪辑。VFW 主要由 6 个模块组成,它们是 AVICap. DLL(负责视频捕捉,这是本文需要重点介绍的部分),MSVIDEO. DLL(包括 DrawDib 等视频绘制函数,用于处理屏幕上的视频操作),MCI. AVI. DRV(MCI 命令解释器),AVIFILE. DLL(可对视频剪辑进行编辑等操作),压缩管理器(负责管理用于视频压缩、解压缩的编解码器),音频压缩管理器 ACM(音频压缩、解压缩编解码器)。本文主要介绍 AVICap 视频捕捉模块的技术原理。

AVICap 支持流式视频结构和单帧实时捕捉,该类提供了对 MCI 视频源的完全控制,开发人员可以任意确定视频捕捉的起点和终点,并且可对其中一个点即单帧进行访问。利用 AVICap 捕捉窗口可实现如下功能:

- (1) 将一个视频流和一个音频流捕捉到一个 AVI 文件中;
- (2) 动态地同视频和音频器进行连接或断开;
- (3) 以 Overlay 或 Preview 模式对输入的视频流进行实时显示;
- (4) 在捕捉时可指定所用的文件名并能将捕捉文件中的内容拷贝到另一个文件中;
- (5) 可设置捕获速率;
- (6) 显示控制视频源、视频流格式和视频压缩的对话框;
- (7) 创建保存或载入调色板;
- (8) 将图像和相关的调色板拷贝到剪贴板;
- (9) 将捕获的一个单帧图像保存为 DIB 文件格式。

微软公司的 AVI 文件格式是一种有 RIFF 指定的应用程序,用来对音频/视频流进行捕捉、编辑和播放的文件格式,并已成为视频图像的 Windows 系统标准。AVI 文件包含不同类型的多重数据流。大多数 AVI 文件里都包含有音频和视频流。有些简单的 AVI 文件可能只包含视频流,而有些更为特殊的 AVI 文件可能只包括一个控制道或 MIDI 道作为一个附加的数据流。本文要介绍的是同时包含音频和视频数据的 AVI 文件。AVIRIFF 的文件结构如下:

```
RIFF('AVI'
LIST('hdr1'
'avih'((Main AVI Header))//主 AVI 信息头数据
LIST('str1'
'strh'((Stream Header))//流的头信息数据
'strf'((Stream Format))//流的格式信息数据
['strd'(Additional Header Data)]//可选的额外的头信息数据
['strn'(Optional Stream Name)]//可选的流的名字
)
)
LIST('movi'
{ SubChunk | LIST('rec'
SubChunk1
SubChunk2
.....
)
}
```

```

    )
    ['idxl' < AVIIndex>] //可选的 AVI 索引数据块
)

```

LIST 块和索引数据块是 RIFF“AVI”数据块的子数据块。LIST“hdrl”数据块定义了数据格式,LIST“movi”数据块包含了 AVI 流的数据,LIST“idxl”数据块是可选择的数据块。AVI 文件须对以上三个分量以适当的顺序保存。

在 AVI 文件的主首部结构 MainAVIHeader 中,有几个重要的成员变量定义如下:

dwMicroSecPerForm:指定视频流的两个帧之间的间隔;

dwMaxBytePerSec:指定文件的最大传输速率,该数值指出了系统为了显示 AVI 流每秒钟必须处理的字节数;

dwTotalFrames:指定文件里的数据帧总数;

dwStreams:指定文件中的流的数目,本文中同时有音频和视频流,所以该变量的值为 2;

dwSuggestedBufferSize:为读取文件指定缓冲区的大小。该值应包含文件的最大数据块。如果设定为 0 或太小,播放软件必须在播放期间重新分配内存,从而影响播放性能。

dwWidth 和 dwHeight:指定 AVI 文件的画面的宽度和高度,以像素为单位;

dwStart 和 dwLength:指定 AVI 文件的开始时间和文件总长度。dwStart 通常设置为 0。

限于篇幅,对于 AVI 文件的其他数据结构不在此详述,读者可参考 MSDN 联机文档。

3 视频流捕捉过程

下面是视频流捕捉的一般流程。

(1)首先要利用宏 capCreateCaptureWindow 创建一个视频捕捉窗口。宏的声明如下:

```

HWND VFWAPI capCreateCaptureWindow( LPCTSTR
lpzWindowName,DWORD dwStyle,

```

```

int x,int y,int nWidth,int nHeight,HWND hWnd,int
nID);

```

下面的代码创建了一个窗口左上坐标为(0,0)右下坐标为(120,140)命名为“My Capture Window”的视频捕捉窗口。

```

hWnd = capCreateCaptureWindow((LPCTSTR)" My Capture
Window", WS_CHILD | WS_VISIBLE, 0, 0, 120, 140,
(HWND)hWndParent, (int)nID);

```

(2)利用宏 capGetDriverDescription 获取系统中可用的捕捉设备。宏的声明如下:

```

BOOL VFWAPI capGetDriverDescription ( WORD
wDriverIndex,LPSTR lpzName,INT cbName,LPSTR lpz-
Ver,INT cbVer);

```

下面的代码显示了从系统中获得可用捕捉设备的方法:

```

Char szDeviceName[80]; Char szDeviceVersion[80];
For(wIndex=0;wIndex<10;wIndex++)
{
if(capGetDriverDescription(wIndex,szDeviceName,sizeof(szDevice-
Name),szDeviceVersion,sizeof(szDeviceVersion))
{ //把得到的驱动器名字加入已安装捕捉驱动器的名字列表里
//以便让用户选择要使用的驱动器
}
}
}

```

(3)利用宏 capDriverConnect 连接捕捉设备。宏的声明如下:

```

BOOL capDriverConnect(hWnd,iIndex);hWnd 表示捕
捉窗口,iIndex 表示设备序号。

```

(4)利用宏 capDriverGetCaps 获取捕捉设备信息。宏声

明如下:

```

BOOL capDriverGetCaps(hWnd,psCaps,wSize);hWnd
表示捕捉窗口,psCaps 表示 CAPDRIVERCAPS 指针,dwSize
表示结构的大小。在这里用到了结构体 CAPDRIVERCAPS,
该结构体的定义在 vfw.h 文件中。

```

(5)利用宏 capGetStatus 获取当前的捕捉窗口信息。宏的声明如下:

```

BOOL capGetStatus(hWnd,s,wSize);参数 hWnd 和
wSize 与前面相同,s 为指向结构 CAPSTATUS 的指针。结
构 CAPSTATUS 的定义在 vfw.h 中。

```

(6)利用宏 capGetVideoFormat 获取视频的图像格式。宏的声明如下:

```

DWORD capGetVideoFormat(hWnd,psVideoFormat,
wSize);参数 hWnd 和 wSize 与前面同,psVideoFormat 为指
向 BITMAPINFO 结构的指针,结构 BITMAPINFO 的定义
如下:

```

```

typedef struct tagBITMAPINFO{
    BITMAPINFOHEADER bmiHeader; //位图信息头
    RGBQUAD bmiColors[1]; //位图颜色表
}BITMAPINFO;

```

bmiHeader 为位图信息头结构 BITMAPINFOHEADER 变量,BITMAPINFOHEADER 结构定义了位图的尺寸,使用的调色板,压缩类型,分辨率等信息;位图颜色表 bmiColors^[1]中有很多个 RGBQUAD 结构,每个 RGBQUAD 数据结构中定义了红、绿、蓝三色的亮度值,RGBQUAD 数据结构的个数由结构 BITMAPINFOHEADER 的成员变量 biBitCount 来确定。获取当前视频图像格式的算法如下:

```

LPBITMAPINFO lpbi; DWORD dwSize;
dwSize=capGetVideoFormatSize(hWnd);
lpbi=GlobalAllocPtr(GHND,dwSize);
capGetVideoFormat(hWnd,lpbi,dwSize);

```

(7)利用宏 capSetAudioFormat 设置声音格式(视频数据是由图像和声音组成)。宏的声明如下:

```

DWORD capSetAudioFormat(hWnd,psAudioFormat,
wSize);其中参数 psAudioFormat 为指向结构 WAVEFOR-
MATEX 的指针,用于设置视频声音的格式。

```

下面的代码将视频声音设置为 11KHz、PCM8 字节的格式:

```

WAVEFORMATEX wfex;
wfex.wFormatTag=WAVE_FORMAT_PCM; //波形设备的格式
类型
wfex.nChannels=2; //声道数,1 为自然声音,2 为人工合成声音
wfex.nSamplesPerSec=11025; //每秒钟一个声道播放、录制的声音
样品
wfex.nAvgBytesPerSec=22050; //数据的传输速率
wfex.nBlockAlign=2; //音频数据的块分布
wfex.wBitsPerSample=8; //每个声音样品的字节数
wfex.cbSize=0; //附加字节数
capSetAudioFormat(hWnd,&wfex,sizeof(WAVEFORMATEX));

```

(8)利用宏 capCaptureSetup 设置视频流。宏声明如下:

```

BOOL capCaptureSetup(hWnd,s,wSize);其中,参数 s
为指向 CAPTUREPARMS 结构的指针,该结构包含了用于
接收、设置视频流的属性,其定义在 vfw.h 中。CAP-
TUREPARMS 结构较大,实际使用中并不需要设置其所以
成员,一般只需根据需要设置其中几项即可。用下面的代码
可设置视频流的帧速度:
CAPTUREPARMS CaptureParms;
Float FramePerSec=10.0;
capCaptureSetup(hWnd,&CaptureParms,sizeof(CAP-
TUREPARMS));
CaptureParms.dwRequestMicroSecPerFrame=(DWORD)(1.0e6/
FramePerSec);
capCaptureSetup(hWnd,&CaptureParms,sizeof(CAP-
TUREPARMS));

```

(下转第 153 页)

- IEEE Int'l Conf. on Data Mining, November 2001
- 6 Cook D, Holder L. Substructure discovery using minimal description length and background knowledge. *Journal of Artificial Intelligence Research*, 1994, 1: 231~255
- 7 Yoshida K, Motoda H. CLIP: Concept learning from inference patterns. *Artificial Intelligence*, 1995, 75(1): 63~92
- 8 Asai T, Abe K, Kawasoe S, et al. Efficient substructure discovery from large semi-structured data. In: 2nd SIAM Int'l Conference on Data Mining, April 2002
- 9 Zaki M J. Efficiently mining frequent trees in a forest. In: SIGKDD'2002 Edmonton, Alberta, Canada
- 10 杨沛, 郑启伦, 彭宏, 等. PFTM: 一种基于投影的频繁子树挖掘算法. *计算机科学*, 2005, 32(2): 206~209
- 11 Golgher R D, Silva P, Laender A. Automatic Web news extraction using tree edit distance. In: WWW-04, 2004
- 12 Liu B, Grossman R, Zhai Y H. Mining Web Pages for Data Records. In: WWW-05, 2005

(上接第 133 页)

AVICap 提供了预览(Preview)模式和叠加(Overlay)模式显示视频。在预览模式中视频帧先从捕获硬件传到系统内存,接着采用 GDI 函数在捕获窗口显示;在叠加模式中,叠加视频的显示不经过 VGA 卡,叠加视频的硬件将 VGA 的输出信号与其自身的输出信号合并,形成组合信号显示在计算机的监视器上。可以利用宏 capPreview 启动和停止预览模式,同时需要用宏 capPreviewRate 设置预览播放频率。如:

```
capPreviewRate(hWnd, 66); //预览播放频率
capPreview(hWnd, TRUE); //启动预览模式
```

可用宏 capOverlay 来启动和停止叠加模式,如:

```
CAPDRIVERCAPS CapDrvCaps;
capDriverGetCaps(hWnd, &CapDrvCaps, sizeof(CAPDRIVERCAPS));
if(CapDrvCaps.fHasOverlay) //检查设备是否支持叠加模式
capOverlay(hWnd, TRUE); //启动叠加模式
```

(9)利用宏 capFileSetCaptureFile 指定为保存视频剪辑

所需的文件名,利用宏 capFileAlloc 为文件分配内存缓冲区。其用法如下:

```
Char szCaptureFile[]="MyCap. AVI";
capFileSetCaptureFile(hWnd, szCaptureFile);
capFileAlloc(hWnd, 1024L * 1024L * 5);
```

(10)利用宏 capCaptureSequence 启动视频捕捉,利用宏 capFileSaveAs 将捕捉的视频数据保存到文件中。视频启动和保存的代码如下:

```
Char szNewName[]="NewFile. AVI";
capCaptureSequence(hWnd);
capFileSaveAs(hWnd, szNewName);
```

上面介绍了视频捕捉的主要流程,为了更好地控制视频流捕捉过程,还需要用到其他的 AVICap 宏。如宏 capDlgVideoSource、宏 capDlgVideoFormat;宏 capDlgVideoDisplay、分别用来显示视频对话框、格式和显示。图 1 是本算法的主要流程图。

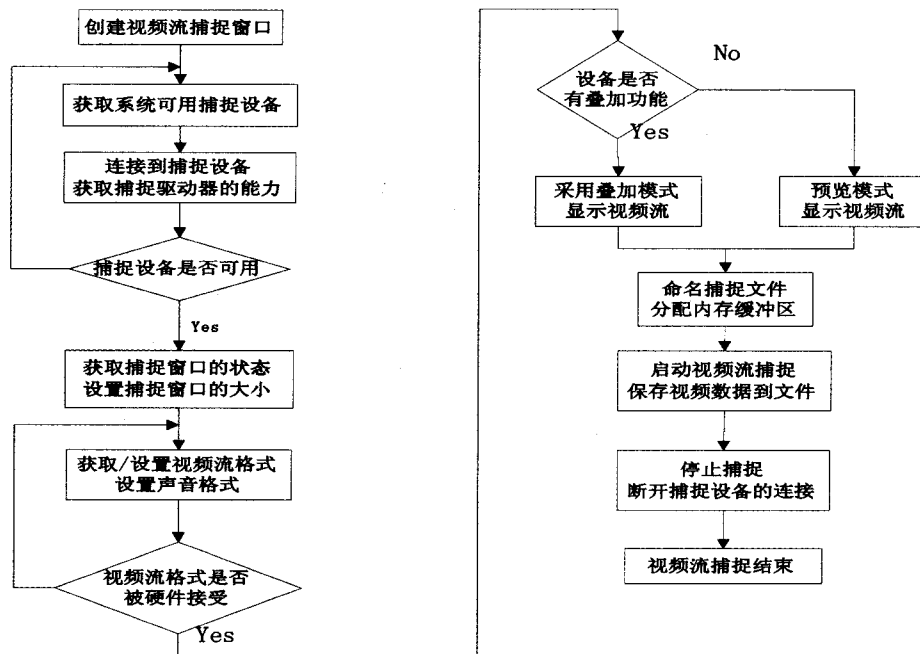


图 1

结束语 本文综述了 VFW 技术中的 AVICap 视频捕捉模块的技术原理,用 Visual C++ 编程实现了 AVI 格式视频流的捕捉算法,并且应用在桌面视频会议、可视电话等多媒体应用中。提出的算法对于从电视机、录像机等外部设备获取数字视频并进行编辑、存储、传输等具有重要的实用价值。

参考文献

- 1 朱仲杰,蒋刚毅,郁梅,等. 一种基于 MPEG-2 的立体视频编码中

的视差匹配快速算法[J]. *电路与系统学报*, 2003, 8(1)

- 2 查锦发,陈莘萌. 音、视频数据捕捉方法[J]. *计算机工程*, 2003, 29(12)
- 3 刘炜玮. *Visual C++ 视频/音频开发实用工程案例精选*[M]. 人民邮电出版社, 2004
- 4 王洪涛. *深入剖析 Visual C++ 编程技术及应用实例*[M]. 人民邮电出版社, 2003