利用DES实现图像序列合成视频

张玉静,陈圣俭
(华北电力大学 计算机系,北京 102206)

摘 要:多媒体文件格式和压缩标准繁多,目前大多数多媒体软件的开发只针对具体的文件格式,文件格式不够灵活,限制了其使用范围。介绍了一种利用DES实现图像序列合成视频方法实现的系统,该系统实现了图像序列合成视频、图像与视频合成视频、切换效果设置、背景音乐添加、编辑效果预览、编辑效果保存。与以往系统相比,该系统合成的视频清晰,过渡效果明显,采用MPEG-4编码,文件占用空间小,并且实现了多种文件格式的兼容,从而应用范围更广。

关键词:DES;图像合成;过渡效果;视频保存
中图分类号:TP39 **文献标识码**:A **文章编号**:1673-629X(2010)06-0209-04

Video Composed of Image Sequence Using DES

ZHANG Yu-jing, CHEN Sheng-jian
(Department of Computer Science, North China Electric Power University, Beijing 102206, China)

Abstract: There are lots of kinds of multimedia file formats and compression standards. At present the development of most of the multimedia software is only for specific file formats, file format is not flexible enough, limiting the scope of use for software. Introduce a system of composing video of image sequence using DES. The system implements video composed of image sequence, composed of image and video, transition set, background music add, edit preview, edit save. Compared with the previous system, the system makes video clear, transition obvious, memory space reduced for using MPEG-4 encoding, and realizes compatibility of a variety of file formats.

Key words: DES; image compose; effect of transition; video save

0 引 言

随着互联网和多媒体技术的发展,图像文件的载体越来越多样化,如BMP、GIF、JPEG、JP2、TIFF、PSD、PCX等图像文件格式。这些格式以高压缩比实现对原有图像信号的高质量编码,从而大大降低了媒体文件的大小,方便了存储和传输。在多媒体信息处理中,需要将不同的图像序列合成视频,在图像序列之间添加过渡效果,图像格式转换成为首要任务,文献[1~4]叙述了不同的方法。

视频图像作为多媒体应用中的主要媒体类型之一,因其庞大的数据量,给信息的存储和传输带来了很大的困难。因此,在多媒体应用中,对视频图像的压缩至关重要,同样在图像序列合成视频后,需要采用有效的方法实现视频的压缩[5~7],分析了MPEG-4视频编码方法。

Microsoft提供的DirectShow[8,9]技术简化了媒体播放、格式转换和媒体捕获方面的编程任务,提供了一系列用于控制媒体流属性的接口,为开发满足具体应用需求的媒体处理工具提供了极大的方便。而DES(DirectShow Editing Services)是一套基于DirectShow核心框架的编程接口,简化了视频编辑任务。用其开发的图像序列合成视频工具具有通用、扩展性强、合成速度快、易于集成等优点,在实际应用中取得了良好的效果。

1 系统实现原理

1.1 图像合成视频的基本原理

视频是活动的图像,正如像素是一幅数字图像的最小单元一样,一幅幅静止图像组成了视频,图像是视频的最小和最基本的单元。视频是由一系列图像组成的,与静止图像不同,视频是活动的图像。当以一定的速率将一幅幅画面投射到屏幕上时,由于人眼的视觉暂留效应,视觉就会产生动态画面的感觉,这就是视频的由来。对于人眼来说,若每秒播放24格(电影的播

收稿日期:2009-10-08;**修回日期**:2010-01-20
基金项目:国家自然科学基金面上项目(60775058);教育部科学技术研究重点项目(107028)
作者简介:张玉静(1983-),女,山东滨州人,硕士研究生,研究方向为多媒体技术;陈圣俭,博士后,教授,博士生导师,研究方向为计算机测控技术。

放速率)、25 帧(PAL 制电视的播放速率)或 30 帧(NTSC 制电视的播放速率)就会产生平滑和连续的画面效果。

如果在图像序列合成视频的过程中,按照一定的算法将两帧静态图像拼接成一系列不同的中间帧,加入到相邻的两帧图像之间,组成新的视频。在新的视频原本相邻的两帧图像切换处就实现了特定的过渡效果,如淡入淡出、交叉溶解、水纹等效果。

1.2 DES 技术

DES 是一套基于 DirectShow 核心框架的编程接口。其原理图如图 1 所示。利用 TimeLine(时间线)暴露的接口编辑图像序列,添加背景音乐;利用 XML Parser(XML 语法解析)语法转换模块,解析 TimeLine 的结构,输出到 XML 项目文件中,可以保存下来,同样它也可以读取 XML 项目文件,根据文件内容生成 Timeline 结构;利用 Render Engine(渲染引擎)根据 Timeline 结构,输出相应的媒体流,最终输出到显卡、声卡,用户就能感受到视频编辑效果;利用 Media Locator(媒体定位)用于媒体定位保存最近访问的多媒体素材文件位置的缓存,当程序试图打开一个多媒体素材文件失败时,通过历史记录找到该素材的位置。

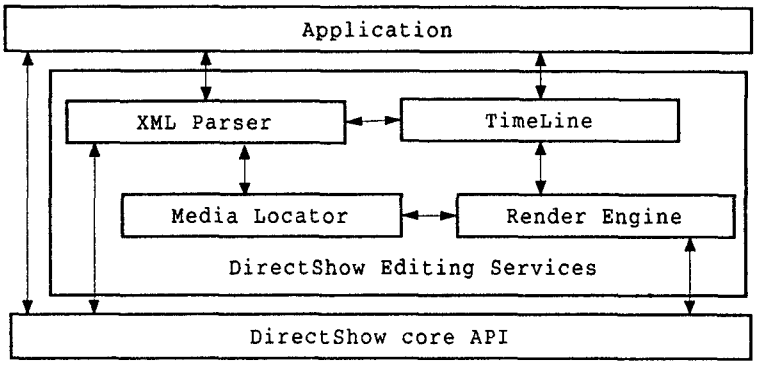

图 1 DES 系统结构图

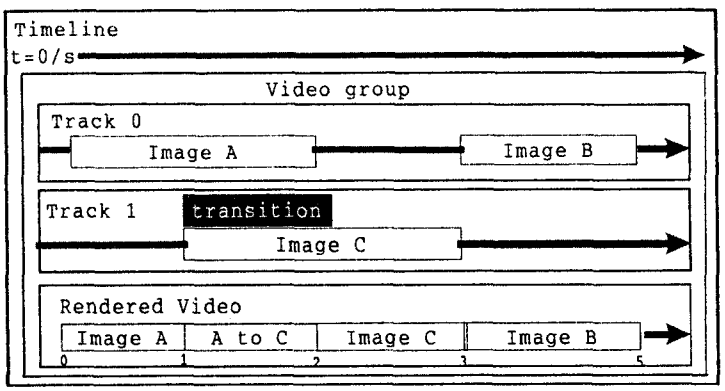

图 2 图像合成序列轨道编排

为了说明图像在时间线轨道中的编排,以图 2 说明。图 2 是图像合成视频的时间线序列,上方的箭头指明了时间方向,开始时间为 0s。在 0s 到 1s 内,输出 Image A;在 1s 到 2s 内,由于 Track1(轨道)的优先级高于 Track0,发生了从 Track0 到 Track1 的过渡。在过渡开始,输出的是 ImageA 的图像;在过渡结束处,输出的是 Track1 的 ImageC;在过渡的中间状态,是 ImageA 和 ImageC 的混合体。在 2s 到 3s 内,输出 ImageC;在 3s 处,由于优先级高的轨道上空,输出低轨道上的 Image B,这种切换是瞬时的。高优先级的轨道到低优先级的轨道是没有过渡状态的,除非特别设定。Render Video 展示了最终的输出视频效果。

2 系统实现与测试

2.1 系统流程

根据要实现的功能,设计构造了如图 3 所示的系统流程过滤器图表[3,4]。输入的图像,通过转换存储颜色模式、调整大小、控制帧率处理,转换为统一的视频图像格式,输入到视频控制组过滤器上;输入的视频,通过音、视频分离过滤器分离出视频流,经视频解码器解码,将解码后的视频流输入到视频组控制过滤器上。视频控制过滤器控制多路视频流,按其在要生成的视频文件中的显示顺序,输入到编码器编码压缩。音频也是同样的道理,只是编码器处理的是音频流。编码后的音频流和视频流,经音、视频复合器,将音频流和视频流合成系统流,经写文件器写入目标文件,实现了媒体文件的编辑保存。

上面的系统实现了图像编辑、视频编辑、音频编辑

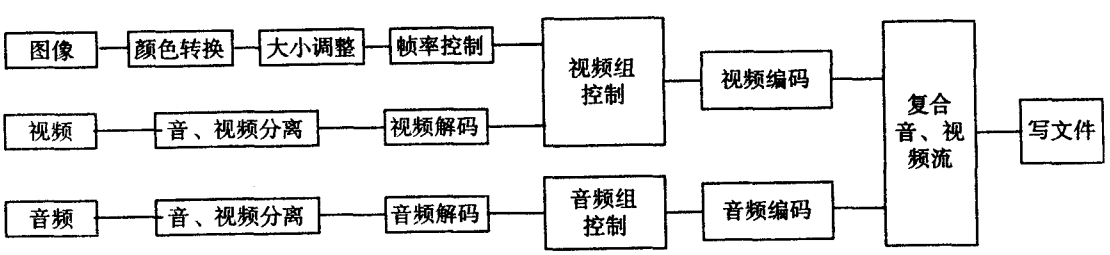

图 3 系统流程过滤器图表

功能,并保存到指定的文件,但是只有在文件保存后才能观看编辑效果。如果编辑效果不满意,则要进行重新编辑保存,增加了时间开销。为此增加了编辑预览功能,可随时观看编辑的效果,随时修改编辑效果,预览满意后再保存。实现预览功能,把图 3 中的视频组控制器和音频组控制器分别接到视频渲染器和音频渲染器,就可以预览编辑效果了。

2.2 系统的实现

2.2.1 时间线的构建

用 DES 实现图像序列合成视频首先必须构建图像序列合成视频的时间线模型,编辑图像信息。其构建过程如下:

(1)调用系统中提供的 IAMTimeline 接口创建一个空的时间线对象模型:

```
pTL: IAMTimeline;
hr := CoCreateInstance(CLSID_IAMTimeline,nil,
CLSCTX_INPROC_SERVER, IID_IAMTimeline,
pTL);
```

(2)这时创建的是一个空的时间线模型,接下来需要使用接口方法 IAMTimeline::CreateEmptyNode 创建各种 DES 对象。包括:IAMTimelineGroup(视频组 pVideoGroup)、IAMTimelineComp(视频 pVideoComp)、IAMTimelineTrack(视频 pVideoTrack)、IAMTimelineSrc(视频 pVideoSrc),IAMTIMELineObj(视频过渡对象 pTransObj)。

```
pVideoGroup: IAMTimelineGroup;
pVideoGroupObj: IAMTimelineObj;
pTL.CreateEmptyNode(pVideoGroupObj, TIMELINE_MAJOR_TYPE_GROUP);
pGroupObj.QueryInterface(IID_IAMTimelineGroup, pVideoGroup);
```

调用 IAMTimeline::CreateEmptyNode 成功后,可以得到一个 IAMTimelineObj 接口指针。也就是说,每个创造的 DES 对象上都实现了 IAMTimelineObj 接口。

(3) 接着在组中加入轨道。

```
pVideoComp.VTrackInsBefore(pVideoTrackObj,
-1);
pVideoTrackObj.QueryInterface(IID_IAMTimelineTrack,pVideoTrack)。
```

(4) 这是最关键的一步,设置媒体源的剪切时间和其在时间线上的时间,然后将其放到相应的轨道上。

```
pVideoSrcObj.SetStartStop;//设置时间线时间
pVideoSrcObj.SetMediaTimes;//设置媒体源时间
pVideoSrcObj.SetMediaName;//设置媒体源文件名
pVideoTrack.SrcAdd(pVideoSrcObj);//将媒体源添加到相应的轨道上
```

(5)接下来设置视频过渡效果。

```
pTransObj.SetSubObjectGuid; //设置过渡子对象
pTransObj.SetStartStop; //设置过渡对象的时间线时间(包括开始时间和结束时间)
pTransable: IAMTimelineTransable;
pVideoTrack.QueryInterface(IID_IAMTimelineTransable, pTransable);
pTransable.TransAdd(pTransObj)。
```

下面还可以在视频过渡对象上设置属性,通过 IPropertySetter 接口实现不同过渡对象上的不同效果。例如设置 SMPTE 过渡对象上的属性示例代码如下:

```
param: DEXTER_PARAM;
value : DEXTER_VALUE;
pProp: IPropertySetter;
CoCreateInstance(CLSID_PropertySetter, nil,
CLSCTX_INPROC_SERVER, IID_IPropertySetter,
pProp); //创建一个属性设置组件
//参数初始化
param.Name := WideString('MaskNum');
param.dispID := 0;
param.nValues := 1;
value.v := 126;
value.rt := 0;
value.dwInterp := 0;
pProp.AddProp(param, value);
pTrackTransObj.SetPropertySetter(pProp); //在过渡对象上设置属性
```

2.2.2 预览功能的实现

创建好时间线后,创建基本渲染引擎 IRenderEngine,它的作用是通过已经建立好的时间线构建滤波器图(FilterGraph)供预览或者输出文件。所以,需把时间线的信息传递给它。调用 ConnectFrontEnd 构建滤波器前端,调用 RenderOutputPins 连接音视频渲染 Filter。至此,滤波器图已成功建立,调用 IVideoWindow 接口的函数,设置播放窗口,调用 IMediaControl 接口的 Run 函数即可进行预览,调用 IMediaEvent 接口的 WaitForCompletion 函数等待预览结束。

```
pRenderEngine : IRenderEngine;
CoCreateInstance(CLSID_RenderEngine, nil,
CLSCTX_INPROC_SERVER, IID_IRenderEngine,
pRenderEngine); //创建基本渲染引擎
pRender.SetTimelineObject(pTL); //确定要渲染的时间线
```

```
pRenderEngine.ConnectFrontEnd( );//构建 graph 的前端
pRenderEngine.RenderOutputPins( );//将前端出来的引脚根据媒体类型分别连接到音视频渲染器,完成 graph 的构建
pGraph: IGraphBuilder;
pControl: IMediaControl;
pEvent: IMediaEvent;
pRender.GetFilterGraph(&pGraph);
pGraph.QueryInterface(IID_IMediaControl, pControl);
pGraph.QueryInterface(IID_IMediaEvent, pEvent);
pControl.Run();//运行 Filter Graph
pEvent.WaitForCompletion(INFINITE, evCode)//等待预览结束
```

2.2.3 保存功能的实现

创建好时间线和前端后,前端输出的是非压缩的音频流和视频流,而要保存的是压缩数据,根据图 3 系统流程过滤器图表,需要加入音视频编码器、音视频复合器、文件写入滤波器。以实现 MPEG-4 格式的文件为例。

第一步,将视频编码器、音频编码器和复用器以及文件写入滤波器加入到滤波器图中。

第二步,得到组的个数及输出引脚指针,根据引脚的媒体类型将其连接到相应的编码器上。

```
pPin : IPin;
pTL.GetGroupCount(NumGroups);
for i := 0 to NumGroups - 1 to
begin
if (pRenderEngine.GetGroupOutputPin (i, pPin) = = S_OK) then
if Succeeded (GetMediaType (pPin)) then
//返回值为 TRUE,则引脚输出的是视流
ConnectPinToFilter(pGraph,pPin,pVideoEncoder)
else
ConnectPinToFilter(pGraph,pPin,pAudioEncoder);
pPin := nil;
end;
```

第三步,将视频,将视频编码器和音频编码器滤波器连接到复用器滤波器上。

```
ConnectFilters(pGraph,pVideoEncoder,pMux);
ConnectFilters(pGraph,pAudioEncoder,pMux);
```

第四步,连接复用器和文件写入滤波器。

```
ConnectFilters(pGraph,pMux,pfilewriter);
```

第五步,创建 MPEG-4 输出文件。

```
pSink: IFileSinkFilter;
pFilewriter.QueryInterface (IID_IFileSinkFilter, pSink);
pSink.SetFileName(WideString(strSaveFile), nil);
```

最后调用 IMediaControl 接口的 Run 函数运行,调用 Stop 函数停止,调用 IMediaEvent 接口的 WaitForCompletion(INFINITE, evCode)等待保存结束。

2.3 系统测试

在 Delphi 环境下实现了图像序列合成视频,可以加入不同的过渡效果,设置图像显示和过渡时间,随时预览,随时保存,并可以为编辑好的视频添加不同的背景音乐。系统经过测试可以加载常见的图像格式,操作简便。编辑完成保存后,点击预览按钮,回放的画面流畅,无马赛克现象,无明显延时。点击保存按钮可保存串编结果,保存后的文件用终极解码等软件对该文件进行回放,回放的画面流畅,效果令人满意。

下面是用本系统将图像合成视频后的几种过渡效果(见图 4)。

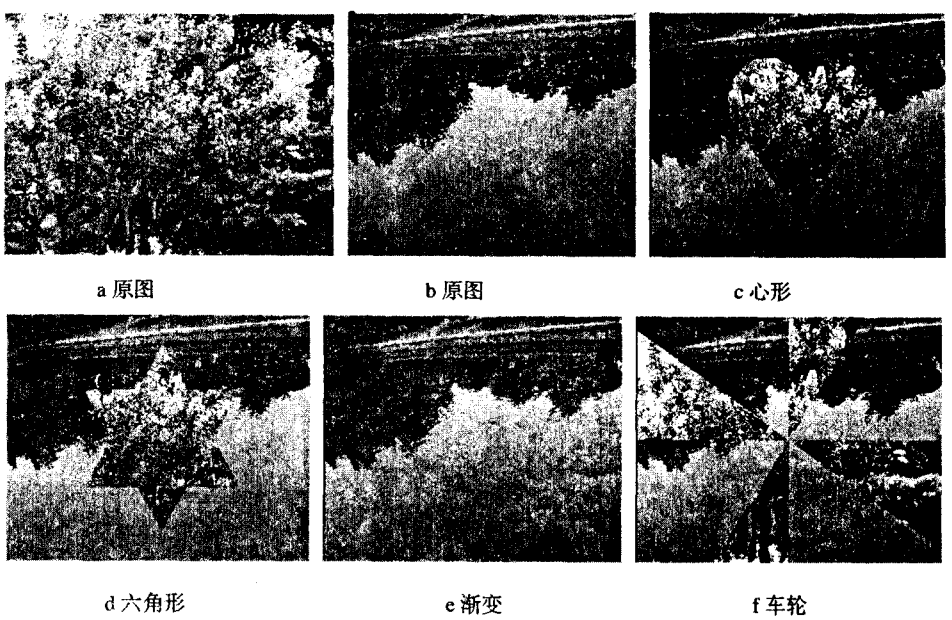

图 4 视频过渡效果

(下转第 216 页)

如果显示：

Processing file deploy. wsdd

<Admin>Done processing</Admin>

则表示发布服务成功。

调用 Web 服务则采用静态 stubs 方式。stubs 是为了隐藏远端调用而在系统里建立的一个代理，使用 stubs 可以封装远程调用的复杂细节，其他类可以像调用本地类一样通过 stubs 使用远程功能。Axis 提供的 WSDL2Java 工具可以通过 Web Service 的 WSDL 文件，自动生成 Java stubs 类。对已部署的名为 svmID 的 Web 服务，使用 WSDL2Java 工具会在 web - inf 目录下，生成一个 svmID 目录，其中包括 4 个 java 类文件：

SVMID_ PortType. java

SVMID Service. java

SVMID ServiceLocator. java

SVMID SoapBindingStub. java

之后在编写 JSP 客户端时就可以直接调用这些 stubs 类，从而完成相应的 Web 服务调用。

3 结束语

实践证明，支持向量机用于农业生产决策领域是可行的，并且，本研究将径向基函数类型的 SVM 封装成 Web 服务提供给用户，用户只需将自己的请求和数据发送给相应的远程服务就可以获得决策结果，轻松享受按需决策服务。Web 服务的相关标准都是 W3C 的开放协议，具有与平台和操作系统无关的特性，采用 Web 服务技术构建的开放、松耦合的数字农业生产智能决策系统在数据共享、软件重用与降低系统集成成本等方面具有明显优势。

本研究中的智能决策方法——径向基函数类型的 SVM，实际上就是一个分类器，可以用于农业生产中的分类问题。模型的构造过程就是通过对样本数据的学习，自动确定支持向量的个数和支持向量的值，无须人工参与，从而使模型的构造更容易、更客观、更高效。它克服了传统 RBF 网络或多层感知器需要根据经验确定参数，局部最优等缺点，是一种较理想的非线性计算工具。但是，通过实验农业领域的不同决策问题发现，训练样本数据的选取对于模型的准确性影响很大，其中训练样本数、数据的维数和数据的相关性等等都是很重要的影响因素，在实际应用中对于数据预处理环节还需要更深入的研究。

参考文献：

[1] 武向良，高聚林，赵于东，等. 农业专家系统研究进展及发展方向[J]. 农机化研究，2008(1)：235 - 238.

[2] 陈月华，胡晓光，张长利. 基于机器视觉的小麦害虫分割算法研究[J]. 农业工程学报，2007，23(12)：187 - 191.

[3] 陈冬冬，杨 春. 支持向量回归机在农业供应链预测中的应用[J]. 四川农业大学学报，2008，26(3)：290 - 292.

[4] Vapnik V N. The Nature of Statistical Learning Theory[M]. New York：Springer，1995.

[5] 杨 斌，路 游. 基于统计学习理论的支持向量机的分类方法[J]. 计算机技术与发展，2006，16(11)：56 - 58.

[6] 韩力群. 人工神经网络教程[M]. 北京：北京邮电大学出版社，2006.

[7] 王晓东，姜 浩. Web Service 同传统分布式技术的比较分析[J]. 计算机技术与发展，2008，18(3)：125 - 127.

[8] 微软公司. XML Web Service 开发[M]. 北京：高等教育出版社，2004.

(上接第 212 页)

3 结束语

由于多媒体技术的发展，人们对多媒体的制作要求越来越高，图像序列合成视频作为其中一个方面，其实用化、普及化具有很强的现实意义。本系统基于时间线管理多媒体，特技合成更加方便可靠；随时预览编辑效果的功能，增加了系统的灵活性；保存格式的多样性，增加了编辑的视频的应用范围；同时本系统支持微软 DirectX 的第三方插件使用，使软件更易于扩展。

参考文献：

[1] 胡春华，曹元大，张 磊. 基于 DirectShow 的视频流媒体存储系统的设计与实现[J]. 计算机工程与设计，2003(11)：31 - 33.

[2] 张 勇，罗 静. 基于 DirectShow 的多媒体文件音视频的重新压缩[J]. 现代电视技术，2005(5)：87 - 91.

[3] 陈志峰，田裕鹏，王珊珊. MPEG - 2 音视频编辑软件的实现方案[J]. 现代电子技术，2009(10)：100 - 103.

[4] 胡海峰，陈 喜，张文渊，等. DirectShow 非线性音频、视频编辑应用的实现[J]. 微计算机应用，2004，25(1)：58 - 63.

[5] 韩 云，陈祖爵，郑尚志. MPEG - 4 编码技术应用及 FPGA 实现[J]. 计算机技术与发展，2007，17(10)：219 - 222.

[6] 严权锋. 基于 MPEG - 4 的综合抗误码方法[J]. 计算机技术与发展，2008，18(4)：169 - 173.

[7] Wiegand T，Sullivan G J，Bjntegaard G，et al. Overview of the H. 264/AVC Video Coding Standard[J]. IEEE Trans，Circuits Syst，Video Technol.，2003，1(7)：560 - 578.

[8] 陆其明. DirectShow 开发指南[M]. 北京：清华大学出版社，2003.

[9] Microsoft Corporation. DirectX 9. 0 Programmerps Reference [CP/DK]. 2002.