

基于 ActiveX/VegaPrime 的交互视景 仿真平台封装的研究

王志乐 于 辉 许路航¹

(海军航空工程学院青岛分院虚拟仿真研究室, 青岛 266041; 青岛大学信息工程学院¹, 青岛 266071)

摘 要 针对 Vega Prime 开发的视景仿真程序复杂度高、共性多、仿真领域多、开发语言单一的特点, 提出了采用组件技术, 结合 GL Studio 功能插件, 实现了集虚拟仪表、Creator 节点控制、视景仿真平台各模块类库在内的封装, 屏蔽了虚拟仿真平台使用的复杂性和开发语言的限制。采用 Windows 消息机制, 确保交互的可靠性, 解决了视景封装后焦点丢失和消息无法截获的问题。通过 ActiveX 能够动态加载场景, 实现与场景的实时交互和控制, 直观快速的完成虚拟仿真软件的开发。文中在多个语言环境下通过该控件实现了多种虚拟仿真软件的开发, 实验证明具有开发效率高、稳定性好的优点。

关键词 Vega Prime 虚拟仪表 视景仿真 ActiveX 虚拟仿真

中图法分类号 TP391.9; **文献标志码** A

随着计算机软、硬件技术的突飞猛进, 计算机图形学在各行各业的应用得到了迅速发展^[1]。虚拟现实、科学可视化和计算机动画已成为近年来计算机图形学的三大研究方向。视景仿真作为虚拟现实最重要的组成部分已经越来越成熟的被运用到各个领域, 如建筑工程、城市规划、辅助教学、军事模拟训练、乃至航空航天。视景仿真技术给了人们用三维空间的、实时交互的、真实感强的图形图像表现事物的能力, 使人们能够形象、直观地了解研究对象的运行情况并对问题进行定位和研究分析^[2]。视景仿真是计算机技术、图形处理与图像生成技术、立体影响和音响技术、信息合成技术、显示技术等诸多高新技术的综合运用^[3], 以建立一个具有身临其境的沉浸感、能与复杂系统进行交互、并能促进构想与创造的环境。

目前实时视景仿真领域主流的高层开发平台是 MultiGen Vega Prime^[4], 该平台提供了大量的函数库用于场景渲染和驱动控制, 能够与 Multigen Creator 三维建模工具配合使用, 实现对三维模型

DOF 节点的实时控制, 同时可以嵌入虚拟仪表软件 GL Studio, 实现与仪表的实时交互。然而, 开发的过程非常繁琐, 开发语言局限在 C++, 对 Windows 编程技术要求较高, 而且不同虚拟仿真软件对视景的实时渲染流程、交互方式是一样的, 对于开发人员来说既要花费大量的时间研究平台的类库, 又要浪费大量时间在代码的编写与优化上。因此, 本文提出了基于 ActiveX 技术实现集虚拟仪表软件 GL Studio、Creator 节点控制在内的视景类库的封装, 提高视景仿真软件开发的效率, 减少软件开发的复杂性, 使视景开发适合所有 Windows 编程语言。

1 基于 ActiveX 的视景封装环境的构建

1.1 ActiveX 组件技术

ActiveX 技术是基于组件对象模型 (COM-Component Object Module) 和 Windows32 API (Application Program Interface) 的一种与具体编程语言无关的技术, 同时也是一种微软公司提供给广大开发人员把计算机桌面环境与 Internet 及其大量资源集成起来的技术^[5]。

ActiveX 引入“软件组件”这个新概念。一个大

2009年9月10日收到

第一作者简介: 王志乐 (1983—), 男, 江苏兴化人, 学士, 助理工程师, 研究方向: 视景仿真、仪表仿真。

型软件由许多小模块组成,这些模块便是“组件”。组件的发展方向是能跨应用程序、跨网络运行。组件对象技术将彻底改变软件生产模式^[6],用户可以大量购买专业化软件组件来构建自己的应用系统,从而大幅度提高工作效率,缩短产品的生产周期。

1.2 视景封装环境的构建

针对 Vega Prime/GL Studio 类库的功能和开发语言的要求,根据面向对象的设计思想,采用 VC + . NET 作为 ActiveX 封装平台。在 VC ++ . NET 新建工程项目中选择 MFC ActiveX 控件,VC ++ . NET 会自动生成一个基于 MFC 的 ActiveX 框架。

视景仿真环境要求具备场景环境渲染的实时性、交互的可靠性以及特效仿真的逼真性和可控制性。因此基于 Vega Prime 视景仿真应用系统开发的流程在开发框架中定义一个用于帧循环的时钟,确保渲染的实时性。通过在 ActiveX 中添加各种方法封装 Vega Prime 中的类模块功能、DOF 控制及 GL Studio 虚拟仪表的输入属性功能,使用户在任何开发语言下通过调用相应的接口方便的完成视觉仿真的控制。Vega Prime 平台提供的人机交互方式主要是通过 Windows 消息机制实现的,因此在封装视的过程中为了交互的一致性和稳定性也采用此方式。

2 视景仿真平台封装的总体设计

基于 Vega Prime 的视景仿真系统的开发主要包括两个步骤:一、场景、模型、仪表创建完成之后,必须通过 Vega Prime 提供的图形用户界面 Lynx 进行加载,并进行一些必要的设置包括窗口、通道、观察者、场景、环境等之后,生成 ACF 文件,该文件是仿真应用的初级配置。二、调用 Vega Prime 的库函数,加载 ACF 文件,完成视景的加载,仿真过程中调用相应库函数对 ACF 文件中的任何对象及仪表进行动态仿真,实现实时交互。

因此根据视景仿真的任务和软件开发的过程设计如下封装结构(见图 1)。

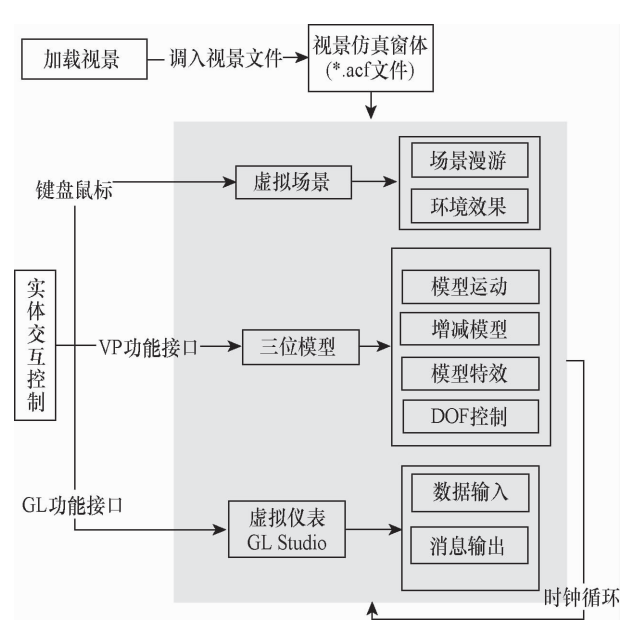


图 1 视景平台封装结构

2.1 视景加载

Vega Prime 视景加载遵循的开发流程为^[7,8]:初始化 vp::initialize,创建 vpApp 实例,载入场景文件 *. acf,配置 configure,帧循环 runloop。因此需要设计一个合理的接口,使用户直接调用组件的该接口即可完成视景的加载,屏蔽加载的开发流程。该接口命名为 setupVp。根据 Vega Prime 视景加载流程实现代码如下:

```
CvpApp * m_App; //应用程序对象
VARIANT_BOOL CvpCtrl::setupVp(LPCTSTR _acfFile)
{
    //判断输入文件是否存在。
    if(_acfFile == "")
        return false;
    int result = vp:: initialize(0, (char * *) "");
    assert(result == vsgu::SUCCESS);
    m_App = new CvpApp;
    result = vpKernel::instance() -> define(_acfFile);
    assert(result == vsgu::SUCCESS);
    m_App -> configure(); //初始配置
    vpWindow * vpWin = * vpWindow::begin(); //获取 VP 窗口
    vpWin -> setParent(m_hWnd); //设置 VP 父窗口
    .....
    m_App -> runloop();
    //定义帧循环时钟
```

```
const int TIMER_ID = 1;
int returnValue = SetTimer(TIMER_ID, 33, NULL);
assert(returnValue == TIMER_ID);
return VARIANT_TRUE;
}
```

帧循环时钟定义好之后在时钟消息响应函数 OnTimer 中实现帧循环,完成视景加载。

2.2 人机场景交互

一个先进的仿真系统必须有一个与之相应的高效的人机交互界面^[8]。交互性主要是指参与者通过使用专门设备,用人类的自然技能实现对模拟环境的考察与操作,它是人机和谐的关键因素。在视景仿真中,交互性主要体现在两方面:一方面是计算机通过声音、图像等反馈给用户的信息,使用户在获取各种数据的同时得到感官上的各种刺激;另一方面是用户通过鼠标、键盘等输入设备对模型、视点进行操作控制,良好的可控性可以方便用户获取更佳的视觉效果和更加全面的试验数据。

Vega Prime 平台提供的人机交互方式是通过 Windows 消息机制实现的,在 ACF 文件中场景的漫游使通过键盘和鼠标实现的。但是基于 ActiveX 封装的视景组件加载场景后将无法通过键盘进行漫游,即无法截获键盘消息,达不到交互要求。因此本文通过窗口句柄的获取,利用 Windows 消息实现场景控制^[9],使仿真平台的封装达到的交互实时性响应和可靠性运行指标要求是一关键技术。

解决封装的交互实时性需要掌握 Windows 消息截获的特点和窗口句柄获取的方法。常用的窗口消息处理函数 WindowPro 无法截获 WM_CHAR 消息,因此封装的视景将无法实现键盘交互,虚函数 PreTranslateMessage 是消息在送给 TranslateMessage 函数之前被调用的,绝大多数本窗口的消息都要通过 PreTranslateMessage,所以通过此函数截获键盘消息非常可靠,实现过程如下:

```
BOOL CvpCtrl::PreTranslateMessage(MSG * pMsg)
{
    switch (pMsg->message)
    {
        case WM_KEYDOWN:
        case WM_KEYUP:
```

```
switch (pMsg->wParam)
{
    case VK_DOWN:
        .....
        vpWindow * vpWin = * vpWindow::begin();
        //SendMessage
        HWND hWnd = vpWin->getWindow();
        ::SendMessage (hWnd, pMsg->message, pMsg->wParam, pMsg->lParam);
    }
}
break;
}
```

2.3 Vega Prime 功能模块的封装

Vega Prime 的功能模块相当多,也相当全面,主要包括:环境模块 vpEnv、vp 模块、海洋模块 vpMarine、特效模块 vpFx、运动模块 vpMotion 等。各个模块包含了大量类,通过 Vega Prime 提供的图形用户界面 LynX 可以完成各个类对象的创建,同时可以设置各种对象之间的父子关系,仿真程序通过查找各对象的名称完成对视景中各个对象的控制。因此对于模块封装也就是要把各个模块中的类进行二次封装,能够方便的进行对象的创建与控制,避免通过 LynX 进行静态的设置,同时也使仿真程序突破了语言的限制。由于各个模块中的类封装的方法和流程基本一致,因此以 vp 模块为例,调用 vpObject 类向场景中增加模型对象,封装流程(见图 2)。

2.3.1 DOF 节点对象的控制

DOF(Degree of Freedom)技术,可以使模型对象具有活动的能力^[10],控制模型的某个部分按照设置的自由度进行移动,如飞机的起落架、坐舱盖、襟翼等。DOF 功能在 MultGen Creator 中实现的,主要包括创建 DOF 节点,建立局部坐标系,设置 DOF 节点相对于局部坐标系在一定的自由度范围内运动等步骤。

在视景仿真过程中对模型自由的控制,需要先获取 DOF 节点,然后控制其转动的参数。但是 Vega Prime 并没有关于 DOF 控制的模块,给视景仿真平台的封装提供了难度。实际视景仿真系统中 DOF 的控制可能在多个模型的多个部位随时交互发生,

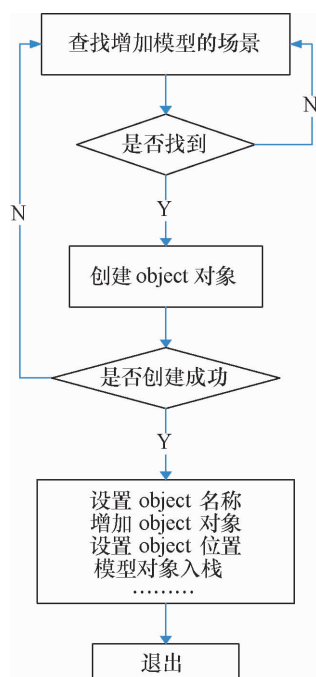


图2 视景封装流程

所以对于 DOF 的控制,需要设计一个合理的结构,使封装的接口能够对 DOF 进行灵活的控制。

在封装接口的设计过程中,嵌入 Vega 提供的类文件 vsDof,

定义 DOF 节点管理列表,完成 DOF 节点的注册。调用封装的接口对 DOF 节点的交互控制,利用节点注册的序号获得对应的 DOF 节点对象,调用 setRotate 和 setTranslate 函数即可实现 DOF 节点六自由度的控制。

设计的注册接口参数包含模型对象名称 objName、DOF 节点名称 DOFName、节点注册序号 regNo。具体实现如下:

```

vsDof * DOF[100]; //DOF 节点管理列表

void dofReg ( LPCTSTR objName, LPCTSTR DOFName, SHORT
regNo) //接口名称定义
{
    char * sDof = (char *) DOFName;
    vpObject * m_obj = vpObject::find( objName);
    vpObject::const_iterator_named nit;
    nite = m_obj -> end_named();
    for ( nit = m_obj -> begin_named(); nit! = nite; ++ nit)
    { //DOF 节点的查找
        char * s = (char *) ( * nit) -> getName();
    }
}
    
```

```

int ir = strcmp(s,sDOF);
if(0 == ir)
    DOF[ indexNo] = (vsDOF * ) ( * nit); //存贮注册号
}
    
```

DOF 节点的六自由度接口 dofnoRotate 和 dofnoTranslate 参数分别包括:节点注册号 regNo、六自由度变量(h,p,r,x,y,z)。

```
DOF[ regNo] -> setRotate ( h, p, r, true);
```

```
DOF[ regNo] -> setTranslate( x, y, z, true);
```

视景仿真中对 DOF 的控制通过调用 dofReg、dofnoRotate、dofnoTranslate 便可以快速实现。

2.3.2 GL Studio 虚拟仪表的交互显示

2004 年 4 月份发行的 VegaPrime2.0 和 GL Studio 推出了实用的虚拟仪表在三维虚拟场景中的交互版本,而这之前仅仅限于平面的交互。

GL Studio 是 DISTI 公司为仪表仿真软件开发提供的一套系统解决方案^[11,12],是一个独立平台的快速原型工具,用来创建实时的、三维的、二维照片级的交互仪表。用户可以利用其图形交互界面以所见即所得的方式来完成仪表面板的制作,通过其代码编辑器来完成仪表内部逻辑仿真^[13]。其代码生成器能够将用户的制作结果自动生成 C++ 和 OpenGL 源代码,用户既可以将其代码进行单独编译也可嵌入到其它程序中进行编译,从而避免了大量繁琐的底层 OpenGL 开发细节。

因此,通过 GL Studio 可视化平台快速完成仪表的制作,打包成类库。在 Vega Prime 中通过 LynX 调用 vpGLStudio 模块,根据制作的仪表类库创建三维场景中的仪表对象,对仪表对象完成拼接设置,也可以在视景加载后通过封装的接口动态设置,由于仪表的名称和位置是固定不变的,建议通过 LynX 可视化界面设置。

仪表从功能特性可以分为两类:一类是接受外部数据的输入完成显示功能,如刻度带、指针、文本框等。另一类需要发送数据如按钮、开关、选钮等。因此 GL Studio 仿真软件提供了属性(Property)接口用于外部对仪表对象的数据输入,回调函数(Call-back)用于仪表内部的逻辑交互。对于视觉仿真平

台的封装需要提供相应接口完成对仪表属性的数据输入,通过调用 vpGLStudio 模块的 vpGLStudioComponent 类完成。仪表显示接口 glsSetAttribute 封装如下:

```
vpGLStudioComponent * m_gls; //仪表指针
//查找输入数据的仪表名称
m_gls = vpGLStudioComponent::find(glsName);
//完成仪表属性的赋值
m_gls->setAttrib(attribName, attriValue);
```

函数 setAttrib 提供了两个参数,attribName 表示仪表制作过程中提供的数据输入属性接口,attriValue 表示外部输入的数据值。视景仿真过程中通过调用 glsSetAttribute 接口即可完成仪表的交互显示,(见图 3)。



图 3 仪表封装的交互显示

3 组件的应用

目前,该组件已在多种编程语言(如:VB.NET、VC.NET、C#、JAVA、DELPHI)的应用中完成了多型飞机的整机飞行仿真训练系统,机型包括固定翼战斗机、旋翼战斗机、大型运输机。所开发的系统视景的实时性、交互性都非常理想。以开发的某型对手机的仿真训练系统为例,(见图 4)。



图 4 某型对手机视景仿真

视景加载后观察视景性能参数(见图 5),帧率保持在 59 Hz,App 线程执行周期为 2 ms—3 ms 范围内变化,Cull 裁减线程执行周期小于 3 ms,Draw 绘制线程执行周期小于 12 ms,因此视景仿真的实

时性、稳定性非常好。

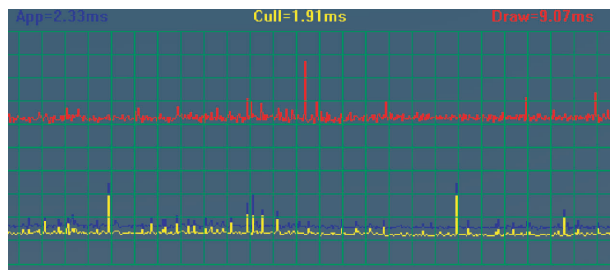


图 5 视景加载的性能参数

4 结论

采用 ActiveX 组件机制,实现虚拟现实软件(Vega Prime、GL Studio)的封装,屏蔽了虚拟现实软件使用的复杂性。扩展了 Vega Prime 只能用 C++ 进行编程的限制。实现了视景开发的通用性方法和流程,极大的提供了虚拟仿真装备系统的开发效率,具有很好的实用价值。

参考文献

- 1 陈 辉,孙汉秋. 虚拟体空间中的触觉雕刻. 计算机学报,2002;25(9):994—1000
- 2 郑 然,张国峰,戴树岭,等. 具有动态编辑功能的视景仿真平台的开发. 系统仿真学报,2008;20(5):1229—1233
- 3 吴家铸. 视景仿真技术及应用. 北京:西安电子科技大学出版社,2001
- 4 柴 毅,史晶晶,冯大龙,等. 基于 Vega 的航天发射场视景仿真系统实现. 计算机仿真,2007;24(6):62—65
- 5 徐洪安,王 民,费仁元,等. 基于 ActiveX 技术的 B/S 模式远程监控系统. 制造技术与机床,2004;1:74—76
- 6 付 超,杨善林,胡小建,等. 基于 ActiveX 的虚拟控制系统通用集成平台研究. 系统仿真学报,2006;18(9):2451—2454
- 7 Multigen-Paradigm Inc. Vega prime programmer's guide version 2.0. USA:Multigen-Paradigm Inc., 2005
- 8 莫卫国,胡 飞. 基于 Vega 的视景仿真系统中的人机交互技术. 计算机仿真,2007;24(10):177—179
- 9 于城蛟,刘 更,王海伟,等. 基于 Windows 消息的软件界面集成技术研究. 计算机仿真,2007;24(12):278—281
- 10 孙崇波,唐永刚,宁建国,等. 基于 Creator/Vega Prime 的虚拟航母漫游设计. 计算机仿真,2007;1(24):222—224

(下转第 7394 页)

EEG Feature Extraction Based on FastICA and Fisher Discriminant Criterion

MOU Hua-ying

(College of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, P. R. China)

[Abstract] Brain-computer interaction(BCI) establishes a direct communication and control channel between human and computer or other electronic device by electroencephalogram (EEG). P300-based speller paradigm is an common communication between Brain and computer. A feature extraction method based on P300 visual evoked potential is introduced. Three subjects is used to analyse. Fixed point of independent component analysis (FastICA) and Fisher discriminant criterion are imploied to implement the feature extraction ,and uses support vector machines (SVM) to classify EEG signal. Compared with the feature extraction based on PCA and Fisher discriminant criterion, it has a good ability to extract feature.

[Key words] brain-computer interface independent component analysis Fisher discriminant criterion support vector machines

(上接第 7390 页)

11 北京华力创通科技有限公司. GL Studio: 真实仪表面板开发工具. 北京: 软件世界, 2007; (2): 44—45

12 樊世友, 邸彦强, 朱元昌. GL Studio 软件在视景仿真建模中的应用. 计算机工程, 2002, (3): 260—261

13 Distributed Simulation Technology Inc. Gl studio user's guide ver3.0. USA: Disti Inc, 2005

Research of Interactive Scene Simulation Platform Based on ActiveX/Vega Prime

WANG Zhi-le, YU Hui, XU Lu-hang¹

(Naval Aeronautical Engineering Institute Qingdao Branch, Qingdao 266041, P. R. China;
Information Engineering College of Qingdao University¹, Qingdao 266071, P. R. China)

[Abstract] To deal with the complicated, common, more simulation fields and single develop language of scene simulation program in Vega Prime, the method of component technology combined with GL Studio plug-in functions was presented. The encapsulation is implemented including virtual instruments, creator node control, each module class library of scene simulation platform, shielding the complicated use of virtual simulation platform and the restrictions of develop language. Using windows message mechanism to ensure the reliability of interaction to solve the problem of focus's lost and the message after the issue can not be intercepted. Scenes can be dynamically loaded through ActiveX, to achieve real-time interaction and control with the scene, fast and direct to complete the development of virtual simulation software. A variety of virtual simulation software's develop is implemented through the ActiveX control in multiple languages environment, the experiment proves to have developed high efficiency and good stability advantages.

[Key words] Vega Prime virtual instrument scene simulation ActiveX virtual simulation