

基于 Vega 的飞行视景仿真系统设计

贾鑫¹ 张延园¹ 张靖² 樊峰峰¹

(西北工业大学计算机学院¹,西安 710129;中航工业第一飞机设计研究院²,西安 710089)

摘要 在飞行视景仿真的仪表建模中,专用仪表设计软件或 Vega 的仪表工具存在开发成本、显示效果及系统扩展性等不足。鉴于此,研究了基于 Multigen Creator/Vega 的复杂飞行仪表模型的建模方法,介绍了飞行视景仿真系统的模块化结构及开发该系统所用到的关键技术。在所给出的关键技术中,详细介绍了如何设计及驱动 Creator 软件建立的仪表实体模型以及开发基于 Vega 的视景驱动程序,并给出了相关的程序代码。阐述了在三维场景中,如何实现飞行模型的运动及仪表模型各模块的精确显示。该系统可以比较真实的模拟交互式的飞行过程,提供了一个通用的仿真平台。

关键词 飞行仿真 仪表模型 视景仿真

中图法分类号 TP391.9; **文献标志码** A

随着计算机技术的发展,视景仿真软件开发平台日益完善,虚拟现实技术也随之日益成熟。虚拟现实(Virtual Reality)是用计算机生成逼真的三维世界,人作为参与者通过适当装置对虚拟世界进行体验和交互作用^[1]。可视化仿真^[2],是接收实时参数,以二维或三维图形图像的形式综合的仿真过程,以实现不同角度、层次、视觉的仿真效果。随着计算机技术的发展,视景仿真软件开发平台日益完善,虚拟现实技术也随之日益成熟。

本文详细介绍了在视景仿真软件中开发模拟飞行系统的设计与实现过程,在三维场景中,通过对飞行场景及飞行状态的控制,来达到模拟飞行的效果。在飞行仿真的仪表模型中,文献[3]中采用基于 Vega 提供的模型,这种模型显示效果不佳,对于复杂仪表很难构建。文献[4]中使用 GI Studio 来建立仪表,该软件可以建立复杂的虚拟仪表模型,但开发成本高昂,与 Vega 等视景仿真工具集成还需要购买专门的模块,其本身包括了大量的编程接口,与系统的集成复杂。在此背景下,提出了一种新的解决方案,使用 Creator 统一建立飞行视景仿真

中的各个模块,通过 Vega 的图形功能自行驱动复杂仪表。即首先利用 Multigen Creator 软件构件模拟飞行中所需构件的三维实体模型,然后在 VC 环境下,调用 Vega API 相关函数来驱动模型文件,从而建立具有一定交互能力的可视化仿真系统。该仿真系统根据输入的飞行状态数据模拟飞行视景环境,输出给仿真窗口,相应的飞行参数通过驱动飞行仪表模型来显示输出。

1 仿真系统简介及模块化结构

飞行仿真系统的主要功能是在三维实景空间中,通过用户的操作控制,模拟飞机的飞行视景环境变化。

飞行视景仿真是交互式的仿真过程,其模拟的对象是飞行状态跟踪视景的变化。飞机的状态信息,如航姿、航向、飞行高度、飞行速度、地理信息、自动导航信息等飞行控制参数是通过飞行仪表来显示输出。在飞行过程中,通过航空仪表和三维地理环境来综合体验飞行过程。

飞行视景仿真系统的模块结构如图 1 所示。

2011年6月10日收到

第一作者简介:贾鑫(1987—),男,陕西人,硕士研究生,研究方向:计算机软件与理论。

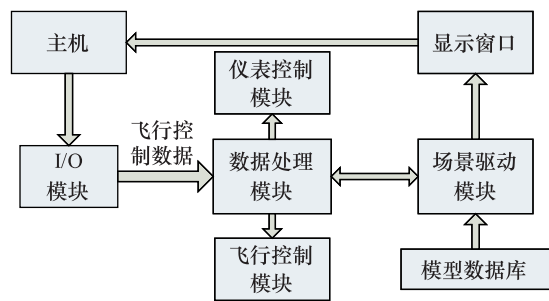


图1 系统模块结构图

图1中,用户通过主机控制操作I/O模块指令,产生飞行控制数据驱动数据处理模块。数据处理模块通过与场景驱动交互,获取飞行状态的各个参数信息,利用仪表及飞行控制模块进行飞行数据的计算处理,得到飞行状态数据。场景驱动模块在飞行数据的基础上调用系统模型数据库,在显示窗口中实时渲染仿真结果。

2 开发环境和系统接口

仿真开发环境采用 Visual C++ 6.0,采用 Multigen Creator 构件模型数据库,采用 Vega Prime 进行视景驱动。

Multigen Creator 是美国 Multigen Paradigm 公司开发的业界领先的三维建模工具^[5]。Creator 软件设计理念完全针对可视化仿真应用,拥有功能强大的开发工具,可以快速建立满足需要的三维模型^[6]。Vega 是 Multigen Paradigm 公司的可视化仿真软件,是一种用于实时仿真和虚拟现实应用的高性能环境和工具。基于 Vega 的仿真系统开发通常包括三个主要的步骤:创建三维模型、生成虚拟场景、编写可执行程序驱动场景仿真。Vega 利用 Lynx 界面通过可视化操作建立起三维场景模型,并保存在 ADF 文件中,而后应用程序通过调用 Vega 函数库对已建好的三维场景进行渲染驱动^[7]。

3 仿真系统开发

模拟飞行仿真环境的建立,需要提供飞行数据

的显示以及飞行控制的操作。主要的技术包括仪表实际模型的建立、运动状态的控制、仿真系统驱动的实现等。

3.1 建立仪表模型

利用 Multigen Creator 软件来创建仿真场景的三维模型,包括综合显示器及其关键构件建模,如图2所示。整个仪表模型分为主姿态、横滚角、速度、高度、垂直速度、航向、地理信息这几个构件。每个构件要模拟真实的飞行环境,在仿真时分别单独对其进行驱动。在建模时采用了 DOF (Degree of Freedom) 技术,即针对每个运动部件加入 DOF 节点,他们控制其各自节点按照设置的自由度范围进行移动或者旋转。同时,在速度及高度的模型中,使用了 Clip、Switch 及 Text 节点,实现其相应的刻度显示。最后通过 PhotoShop 软件绘制仪表效果图像,对各个部分添加相应的纹理,使仿真仪表的效果更加逼真。

整个仪表模型包括多个平面的平行排列,需要删除仪表模型中的冗余多边形和背面多边形,减少系统所需渲染的多边形数目,提高仿真效率。

在开发仿真程序时,根据飞行姿态及控制参数调用 Vega API 相关函数进行 DOF 节点自由度设置,可以使飞机在运动过程中,实时显示飞行姿态的相关参数信息。

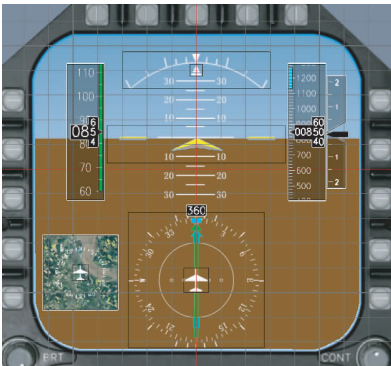


图2 仪表模型

3.2 运动状态的控制^[8]

在模拟飞行过程中,需要控制仿真模型的各个运动模块。针对 Vega 对象及 Creator 模型中的 DOF 节点,精确控制运动方向与范围,模拟仪表的数据

显示和飞行状态。仪表的显示一般需要极高的精度,通过 Vega 中三维坐标的计算和编程,可以实现任意精度的要求,自定义程度比较高。

飞机的飞行状态需要空间 6 个自由度的同时确定,如图 3 所示。 $x、y、z、h、p、r$,分别代表三维坐标值和沿三个轴的转向角, $h、p、r$ 的正方向分别为 $Z、X、Y$ 轴的正方向,分别对应飞机六自由度模型的 $(x,y,z,\psi,\theta,\varphi)$ 。飞行过程中,飞机质心为坐标原点,通过三个坐标轴来定位飞行位置,通过绕坐标轴的旋转来模拟飞行的角速度。

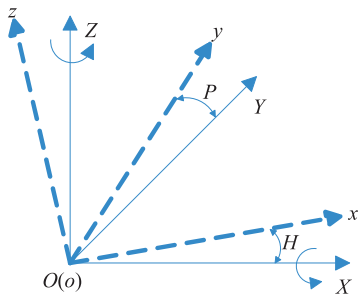


图3 飞机的六个自由度
注:Heading(H)表示飞行航向,Pitch(P)表示飞行俯仰角。

飞行仪表的坐标系同样包括六个自由度。仪表中每个部件拥有各自的局部坐标系,在独立节点内部坐标系运动。通过驱动仪表模型的多个构件,将其节点运动范围与仪表数据显示范围对应,精确定义仪表显示精度,实现仪表模型的运动显示。与仪表构建不同,飞行姿态是在三维地形中,按照场景坐标系来运动。整个飞行运动的各个模块,其位移和旋转都是由六自由度模型的改变来实现。

飞行姿态及仪表运动在 Vega 下的数学表示,是将飞机的运动过程的变化分解到三维视景驱动的每一帧内。飞行过程中的俯仰角和横滚角通过坐标变换和投影,分解在三维坐标轴上,然后调整飞行姿态来显示。飞行过程中,通过观察视点的不断调整,从不同角度观察场景的运动状态。

定义 Δf 为每一帧的位移量,则当前飞行姿态下该位移量经过坐标投影在场景坐标系内的偏移公示如下:

$$\begin{bmatrix} x_f \\ y_f \\ z_f \end{bmatrix} = \begin{bmatrix} -\sin H \cos P \\ \cos H \cos P \\ \sin P \end{bmatrix} \Delta f \tag{1}$$

运动模型的自由度变化为:

$$\begin{bmatrix} x_N \\ y_N \\ z_N \\ \varphi_N \\ \theta_N \\ \varphi_N \end{bmatrix} = \begin{bmatrix} x_{N-i} \\ y_{N-i} \\ z_{N-i} \\ \varphi_{N-i} \\ \theta_{N-i} \\ \varphi_{N-i} \end{bmatrix} + \begin{bmatrix} x_f \\ y_f \\ z_f \\ \varphi_f \\ \theta_f \\ \varphi_f \end{bmatrix} i \tag{2}$$

式(2)中, i 表示显示的帧数。 $x_f,y_f,z_f,\varphi_f,\theta_f,\varphi_f$ 表示在各自由度范围内每一帧的变化率。通过确定显示帧率,调整显示效果,实现整个飞行仿真系统的运动。

3.3 飞行视景仿真的实现

飞行视景仿真系统的建立^[9]通过将 Vega 的 API 链接到 VisualC++ 中,将 .adf 文件作为 VC 工程参数启动运行。

下面是 Vega 线程和成员函数:

```
vgInitSys(); //系统初始化
vgDefineSys(argv[1]); //读取配置文件
vgConfigSys(); //完成系统配置
/* 获得主窗口、通道、环境及观察者句柄 */
win = vgGetWin( 0 );
obs = vgGetObserv( 0 );
chan = vgGetObservChan( obs, 0 );
gfx = vgGetObservGfx( obs );
env = vgGetEnv( 0 );
while ( 1 ) //Vega 主循环
{
    vgSyncFrame(); //帧同步处理
    vgFrame(); //当前帧处理
    flightCacluate(); //飞行参数的计算
    change( win, gfx, chan, env );
    //控制附加通道显示
}
```

实现飞行视景仿真就是通过控制飞机及仪表模型数据在场景中的位置变化,来模拟飞行状态的改变。Vega 为场景中的每个目标及目标中的节点都建立了位置属性。通过 `vgGetPos()` 函数获得当前

调用的模块位置参数,通过 `vgPos()` 函数重新设置。

在飞行仿真系统中,飞行状态和仪表模型的显示是通过调用 `vgPosVec()` 及 `vgRotMat()` 修正当前模型的矢量位置参数。通过仪表及飞行控制模块的计算,来调整至目标位置,然后对三维场景进行渲染。

飞行状态的交互控制可以通过键盘指令调用 `vgGetWinKey()` 函数获取响应;也可以使用串口操作杆,调用 `joyGetPosEx()` 来接受 I/O 数据,模拟飞行过程,调整仪表显示状态。

仿真系统的控制流程如图 4 所示:

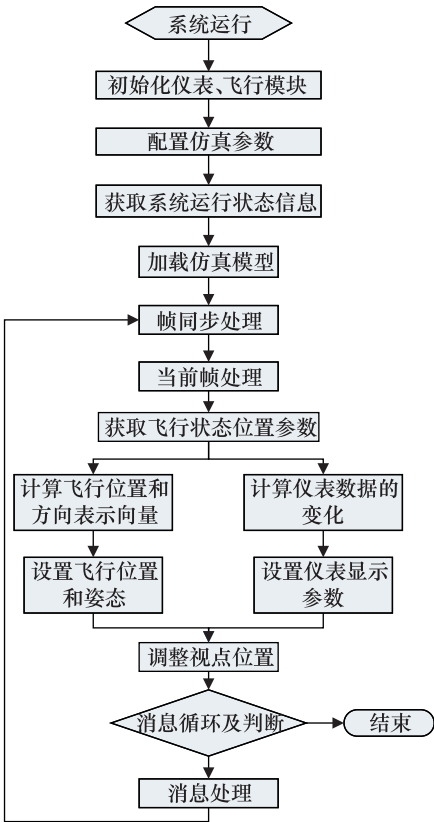


图 4 飞行视景仿真系统控制流程

4 仿真实验结果

在构建的仿真系统中,控制输入产生模拟的飞控数据,综合显示飞行状态中的各种信息。最后的仿真效果如图 5 所示。



图 5 仿真窗口显示

其中飞行地理环境通过仿真窗口来显示,仪表模型显示了当前的飞行参数。

5 小结

本文给出了飞行视景仿真系统的模块划分,详细说明了仪表及飞行控制的设计及实现方法。利用 Multigen Creator 建立了仪表及地形构件的模型,通过调用 Vega API 函数在三维地形内驱动模拟飞行过程,提供多种操作方式,完成了具有一定交互功能的可视化仿真平台。使用这种方法,可以方便快捷的构建仿真飞行环境,模拟真实的飞行运动。仿真结果表明:基于 Vega 的飞行视景仿真系统功能结构具有通用性,有一定的工程应用价值。

参 考 文 献

- 1 威泽华,杜小菁,廖炎平,等. 基于 VEGA PRIME 的集束火箭仿真训练系统的研究. 系统仿真学报, 2010;22(2): 366—368
- 2 王 乘,李利军,周均清,等. Vega 实时三维视景仿真技术. 武汉:华中科技大学出版社, 2004
- 3 赵银玲,吴成富,陈怀民. 基于 Vega 的无人机视景仿真应用研究. 测控技术, 2008;27(1): 96—99
- 4 杨大光,常 波,孙国庆,等. GL Studio 在飞机仪表板仿真中的应用. 现代电子技术, 2010;(24): 158—160
- 5 王 乘,周均清,李利军,等. Creator 可视化仿真建模技术. 武汉:华中科技大学出版社, 2005
- 6 宋 磊. 船舶操纵模拟器三维视景建模技术研究. 哈尔滨:哈尔滨工程大学, 2007
- 7 王 琳,孟晓风,钟 波,等. 交互式飞行视景仿真系统的设计与实现. 系统仿真学报, 2008;20(1): 73—75

8 刘 伟,郭 雷,于 勇. Vega 中运动模型的研究与仿真. 微处
理机, 2010;(1):53—57

9 MultiGen Paradigm Inc. Vega programmer's Guide. USA: MultiGen
Paradigm Inc, 2001

Visual Simulation Design of Flight Based on Vega

JIA Xin¹, ZHANG Yan-yuan¹, ZHANG Jing², FAN Feng-feng¹

(College of Computer Science, Northwestern Polytechnical University¹, Xi'an 710129, P. R. China;
AVIC the First Aircraft Institute², Xi'an 710089, P. R. China)

[**Abstract**] In the special instrument designed software and Vega instrument tools for instrument modeling of the flight visual simulation, there are disadvantages in development cost, display effect and system scalability. In light of that, a simulation method for complex instrument model based on Multigen Creator & Vega was researched. A general modularized functional structure of the visual simulation flight system was presented, and the key technologies which were used for developing the system were given too. Among the key technologies, the method of designing and driving instrument model with Creator and programming the Visual driver based on Vega was shown. Also the code was attached. The principle of achieving the movement of flight model and the precise display of the instrument modules were proposed. The system can simulate the real-life interactive flight, providing a generic simulation platform.

[**Key words**] flight simulation instrument model visual simulation