

基于 OpenGL 的卫星跟踪仿真

黄 权 徐学军

(华中科技大学 水电与数字化工程中心 湖北 武汉 430074)

摘 要 基于开放式三维图形程序标准 OpenGL 的卫星三维仿真实现, 论述了在 VC++ 6.0 中利用 OpenGL 进行三维仿真的优点, 以及卫星几何模型的构建、卫星运动轨迹的确定、仿真过程的动态显示以及系统软件的主要功能和实现。由此得到了效果比较好的卫星跟踪仿真图。因受限于目前的计算机技术尚不能做到对卫星上所有设备进行图形显示监测, 但是卫星三维视觉仿真对于卫星检测系统和卫星管理系统依然具有很大的实用价值。

关键词 卫星仿真系统; OpenGL; 建模; 三维仿真

中图分类号: TP391.41

文献标识码: A

文章编号: 1673-629X(2007)02-0131-04

A Simulation of Tracking a Satellite Based on OpenGL

HUANG Quan, XU Xue-jun

(Hydropower & Numerical Project Center, Huazhong
University of Science and Technology, Wuhan 430074, China)

Abstract Discuss a three-dimensions simulation and realization of a satellite based on OpenGL. The advantages of the three-dimensions simulation with OpenGL running in VC++ 6.0, the geometry modeling of a satellite, the trajectory obtaining of a satellite, the dynamic display of the simulation processes and the functions & realization of this system. What's more, a lot of good stimulation pictures of tracking the satellite are gotten, following the steps above. Though parts of the equipments in the satellite can't be supervised, without more advanced computer technologies, the three-dimensions simulation of a satellite still has great benefits to the checking and managing systems of a satellite.

Key words satellite simulation system; OpenGL; modeling; 3DS

0 引 言

计算机图形学的发展使得计算机三维表现技术愈来愈完善, 科学可视化、计算机动画和虚拟现实已成为计算机图形学的三大热门话题。三维图形的一项主要应用技术是可视化技术(Visualization), 使人能够在三维仿真(3-Dsimulation, 3DS) 世界中直接对具有形体的信息进行操作, 赋予人们对物体进行仿真并且实时交互的能力, 因此也得到越来越广泛的运用。

1 OpenGL 简介

SGI 公司推出的 OpenGL 三维图形库因其易于使用而且功能强大而成为高性能图形和交互式视景处理的标准。OpenGL 使用简便, 功能强大, 具有建模、变换、颜色模式设置、光照和材质设置、纹理映射、双缓存

动画等基本功能, 并提供了线框绘制、深度优先线框绘制、反走样线框绘制、平面明暗处理、光滑明暗处理、加阴影和纹理、运动模糊绘制、大气环境效果、深度域效果等丰富的绘图方式。

与其他三维图形开发工具相比, OpenGL 具有以下鲜明的特点:

(1) 与 3DS 相比, OpenGL 没有现成的对象模型可用, 必须从几何图元的点、线和多边形来构造你希望的对象, 但其具有 3DS 不可比拟的灵活性、可控性, 与其他编程工具结合, 如 Visual C++ , 能产生 3DS 无法获得的动态效果。

(2) OpenGL 直接受 Windows 支持, 利用动态连接库 opengl32.dll 直接提供应用程序与 OpenGL 成员函数的连接, 不需要其他的运行环境(而不象其他一些工具如 ARX 等, 需基于 AutoCAD 环境才行)。

(3) OpenGL 独立于硬件与操作系统, 流水线工作方式, 可以在许多平台上实现, 并可以工作于网络环境下, 采用客户端(Client) 和服务端(Server) 模式, 可在不同类型的机器上运行。目前, 应用 OpenGL 的工具

收稿日期: 2006-04-25

作者简介: 黄 权(1979-), 男, 湖北利川人, 硕士研究生, 研究方向为中国电力市场与水电上网竞价策略; 徐学军, 教授, 研究方向为地理信息系统和流体力学。

主要是 C++ 语言,特别是 Visual C++,其 2.0 以上版本均可直接支持 OpenGL,再加上其强大的功能、丰富的资源,已成为 OpenGL 的首选开发工具^[1]。

2 系统设计

2.1 卫星飞行原理和轨道计算

(1)就人造地球卫星来说,其轨道按高度分低轨道和高轨道,按地球自转方向分顺行轨道和逆行轨道。这中间有一些特殊意义的轨道,如赤道轨道、地球同步轨道、对地静止轨道、极地轨道和太阳同步轨道等。

(2)卫星运行轨道的理论,即开普勒太阳系中行星运动的三大定律:a.圆轨道上环绕太阳运动,而太阳在一个焦点上;b.太阳和行星的矢径在相等的时间间隔中扫过相等的面积;c.行星的轨道周期的平方与它的轨道的长轴的三次方成正比。

(3)卫星运行轨道即开普勒轨道是圆锥曲线,当极坐标原点在实焦点时的方程为:

$$r = \frac{P}{1 + e \cos \theta}$$

其中 P 为半参量,而 e 为偏心率。

应用坐标原点置于中心体上的绝对笛卡儿坐标系,由牛顿第二定律 $F = Mr''$,可得到 3 个偶合的微分方程:

$$\ddot{x} = -\frac{\mu x}{(x^2 + y^2 + z^2)^{3/2}}$$

$$\ddot{y} = -\frac{\mu y}{(x^2 + y^2 + z^2)^{3/2}}$$

$$\ddot{z} = -\frac{\mu z}{(x^2 + y^2 + z^2)^{3/2}}$$

这组方程表明积分该方程组将有 6 个积分常数,也就是最常用的经典轨道常数,即开普勒轨道常数,用来描述在空间中的卫星的轨道。可以用以下这些常数递推出卫星在过去或将来的位置:

- a : 半长轴,确定轨道的大小。
- e : 偏心率,定义轨道的形状。
- i : 倾角,赤道平面与卫星轨道平面间的夹角。

Ω : 升交点赤经,从春分点到卫星轨道由南往北穿过赤道的那一点的角度。

ω : 近地点幅角,在轨道平面内升交点到近地点的角度。

ν_0 : 在历元时刻的真近点角,在指定时间(历元)由近地点到卫星所在点的角度。

前 5 个是几何要素,在理想状况下是不变的,第 6 个是时间要素,它总是在不停地变化着。同时人造地球卫星在空间绕地运动通常受到地球引力、日月引力、潮汐力、太阳辐射压力和大气阻力等力学作用,实际轨

道与理论计算常有偏差,故除考虑其他摄动力影响外,定期更新卫星数据库中的轨道参数也显得尤为重要。基于以上分析,这里主要采用解微分方程的方法由已知的 6 个参数进行卫星运行轨道的初轨计算。

2.2 卫星运行仿真系统设计

中国航天技术随着计算机的发展,在航天产品设计与应用中也开始大量地引用了计算机图形技术,如 CAD 辅助设计。目前,随着我校空间信息科学技术研究的深入,在卫星监测中引入三维可视化图形技术已是必然的趋势。因此,结合我校对卫星监测的要求,开发具有良好界面的卫星监测图形软件,对提高卫星监测和遥感信息科学研究的深入开展具有重要的意义。

卫星监测图形软件拟有以下几方面的功能:

- (1)实现卫星太空位置的可视化、姿态可视化,提供全视景漫游;
- (2)实现卫星数据和地形数据分析可视化显示;
- (3)实现卫星信号覆盖预报,及覆盖区地理信息可视化;
- (4)实现太空已知卫星信息的计算机管理,参数和数据借助数据库(SQLServer2000)管理。

卫星监测图形软件能显示卫星的整体面貌,即卫星在太空的位置、卫星的姿态、卫星外部可见的其他形状,提供卫星数据参数查询、分析以及部分信息的解密处理等,提供地球地理数据查询、显示的支持。

3 卫星模型设计

三维图形的建立 OpenGL 的基本元素是称之为 Vertex 的点,以真实世界的三维坐标表示,由夹在 glBegin() 和 glEnd() 之间一系列 Vertex * 的点构成线和多边形,再由多边形去模拟物体表面、构造物体对象,最终显示对象的三维图形。在这过程中,现实世界中的对象经过取景变换、投影变换、视区变换,在屏幕上形成了三维图形。

常用 OpenGL 生成三维实体模型有 3 种方法:

一是利用简单三维实体进行剪裁、拼装,常用于简单实体建模;

二是通过接口文件使用用户输入的物体型值点数据自编算法精确构造三维实体;

三是借助三维建模软件建造物体模型,然后通过读取标准文件如 DXF、ASE 等在程序中重建实体,或者直接使用转换工具将 3DS 文件转换为 OpenGL 显示列表。

由于 OpenGL 没有提供高级命令函数来定义复杂的三维模型,而只提供上述点、线、多边形的方式来构建三维模型,提供了画复杂几何物体的机制,而未提供

描述复杂的几何物体及建立复杂几何物体模型的手段,所以描述卫星的三维几何图形,即建立卫星三维模型是一项艰巨的工作。综合比较,采用借助三维建模软件(3DS MAX)建造卫星模型和地球模型既有利于三维仿真,也利于模型运动的计算和控制,可减少大量工作量。

卫星用 3DS MAX 建模,模型建立后保存为 3DS 数据格式,需要注意的是 3DS 数据格式只能保存和转换几何信息,色彩、材质等信息可能会在转换时丢失,在系统的开发中要对色彩、材质等进行处理。对于数据格式的转换,主要的核心是对模型的构造数据进行转换,形成 OpenGL 的模型构造列表,作为 OpenGL 的显示列表进行模型的重构。

在数据的转换过程中,模型的数据主要是将以下 3 个关键的数据集进行处理:

(1) 顶点集 (Vertices), 顶点的坐标及其法向量的列表。

(2) 多边形列表 (PolyList), 对应的是一组三角形,与材质等显示的属性有关,是在 OpenGL 中进行模型重构的重要部分。

(3) 相关三角形列表 (Common TriangleList), 一个指向多边形列表 (PolyList) 中每一个包含该定点的三角形链表^[2]。

数据的转换作为一个单独的模块 (C3dsReader 类) 进行编程,模型在建立时需注意模型的基准点,色彩、材质、纹理、光照和背景的处理考虑到模型的建立和数据的转换,在模型建立时,并不对模型进行色彩、材质、纹理的赋予,在完成数据转换后,进入 OpenGL 中再对其进行处理,使模型的数据简洁,转换方便,提高开发的效率和系统的通用性。运动方式的和控制对仿真显示的动画效果来说,物体的运动实际上是重绘,由于模型在 OpenGL 中是以显示列表的方式存在,不需要重构,所以只需要知道物体的位置,结合显示技术的处理,则可以得到很好的运动仿真效果^[3]。

在系统的开发中,卫星运动的处理和控制是作为一个单独的模块来开发的,传入运动的方式参数、姿态参数,输出物体的位置参数来实现运动的仿真。

4 三维仿真实现

4.1 运用 OpenGL 三维图形库编写实时动画

Visual C++ 灵活的应用向导 AppWizard 和完善的基本类库 MFC 加上强大 OpenGL 库,使得开发复杂的三维图形应用程序变得比较容易。由于 OpenGL 函数库和操作系统无关,与 Windows 的图形设备接口

(GDI) 模型以及多数 MFC 应用程序的建立方法不太一致,因此必须建立 VC 与 OpenGL 的应用接口。

首先,必须重新设置画图窗口的像素格式,使其符合 OpenGL 对像素格式的需要。为此需声明一个 PIXELFORMATDESCRIPTOR 结构的变量,并适当地设置某些结构成员的值,使其支持 OpenGL 及其颜色模式。再以此变量为参数调用 ChoosePixelFormat() 函数分配一个像素格式号,然后调用 SetPixelFormat() 将其设置为当前像素格式。然后为 OpenGL 建立着色场景,其作用类似于 Windows 的设备场景 DC,建立了着色场景后,OpenGL 才能调用自己的绘图语句在窗口中绘出图形。Win32API 提供了几个操作着色场景的函数,包括建立、复制、使用、删除、查询等,它们都以 wgl 为词头。着色场景是以线程为单位的,每一个线程必须使用一个着色场景作为当前着色场景才能执行 OpenGL 的绘图函数语句。wglCreateContext() 是建立着色场景的函数,它以一个设备场景句柄为参数,返回一个与此设备场景相联的着色场景句柄。再以此二句柄为参数调用函数 wglMakeCurrent() 使着色场景成为线程当前使用的着色场景,这样便建立了 Windows 下 VC 与 OpenGL 应用接口^[4]。

最后还需在 CMySDOpenGLView 类中初始化着色场景,如设置背景色、深度测试、比例因子、投影方式、光源属性、材料属性及最初的场景变换等,在 RenderScene 函数中利用 OpenGL 函数进行三维仿真绘图,三维场景中的运动画面利用定时器 OnTimer 函数控制时间间隔重绘场景。

4.2 实现卫星三维仿真显示的步骤

1) 调用显示函数 Draw3ds,通过 CTriList 类读取卫星 3DS 模型文件,构建并显示仿真卫星三维造型;

2) 提取卫星参数,调用卫星运动轨迹计算函数 CaculateSatellitePosition,计算并返回卫星的运动位置空间位置 (x, y, z),通过 OpenGL 矩阵变换来实现位置变换;

3) 调用 OpenGL 的帧缓存操作函数 glxSwapBuffers,交换绘图计算缓存和显示缓存。

为了获得仿真过程的高度真实感显示,可以采用 OpenGL 所提供的其它相关命令与函数来实现:

(1) 绘制地球、月球及卫星轨道,调整卫星空间姿态;

(2) 为造型体各表面赋予接近真实物体的色彩、材质、贴图和纹理;

(3) 使用定向光源、定位光源或聚光灯,调整明暗和阴影;

(4) 利用投影变换、透视变换、景深、混合、透明、场

景反走样、深度检测等功能，实现对造型体运动以及对场景的真实模拟。

4.3 导航窗口和可变视点设计

导航窗口及多视图，通过在同一个窗口里划分多个子窗口，为视图创造分频效果(见图 1 下)。具体实现是运用 OpenGL 的 glViewport 函数实现分频，目的是在具体观察地球、月球或卫星的时候能对卫星空间态势同时有一个了解^[5]。

视点变化有两方面的含义：一是视点与场景相对位置的变化，即从不同方向观察卫星。本系统设计了用鼠标和键盘分别实现视点的任意变化，用以观察整个空间态势；二是视点随动，因为地球处于自转状态，若要观察卫星扫描区域必须使视点随卫星运动。为解决这一问题可将视点设计成追踪卫星，由 GluLookAt 函数实现。系统仿真设计结果见图 1。

5 结 论

卫星在发射及在轨道运行时需要监测的数据量很大，理论上卫星监测图形软件可以做到对所有星上设备进行图形显示监测，但考虑到软件工作量、目前计算机的运行速度及监测范围的要求，目前的软件显示部分只包含卫星的太空位置、卫星的姿态、天线和帆板状态及部分关键部件的可视化。要开发逼真的三维仿真程序，还需研究许多具体问题，如对象的数据结构、模型数据库、可视化建模方法、渲染效率、仿真视觉效果等。卫星三维视觉仿真对于卫星检测系统和卫星管理系统具有极大的实用价值。

参考文献：

[1] 吴 斌,段海波,薛凤武. OpenGL 编程权威指南[M]. 北京:中国电力出版社,2003.

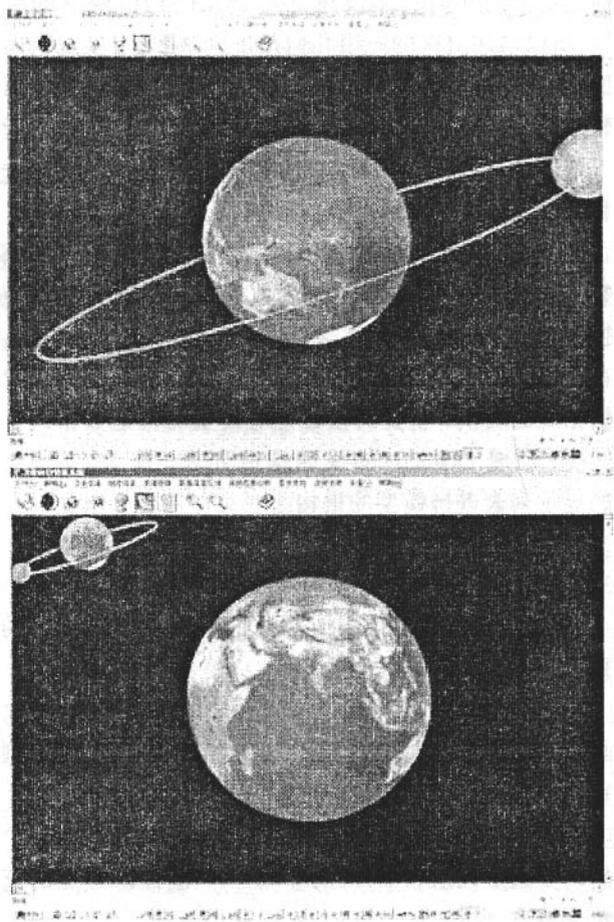


图 1 卫星跟踪仿真结果

[2] 惠晓钟. 3DStudioMAX 新创意[M]. 西安:西安电子科技大学出版社,1999.
[3] 廖朵朵,张华军. OpenGL 三维图形程序设计[M]. 北京:星球地图出版社,1996.
[4] 李 颖,薛海斌. OpenGL 技术应用实例精粹[M]. 北京:国防工业出版社,2001:127-129.
[5] Neider J. OpenGL Programming Guide[M]. [s. l.]:Addison Welsey,1993:42-56.

(上接第 49 页)

参考文献：

[1] Muller G, Moura B, Bellard F et al. Harissa: A flexible and efficient Java environment mixing bytecode and compiled code [C]//in Proceedings of the 3rd Conference on Object-Oriented Technologies and Systems. Berkeley, France: University of Rennes, 1997: 1-20.
[2] Andoh Y. j2c/CafeBab[EB/OL]. 1998. http://www.webcity.co.jp/info/andoh/java/j2c.html.
[3] Shaylor N. JCC - A Java to C converter[EB/OL]. 1997. http://compilers.iecc.com/comparch/article/97-05-149.
[4] Proebsting T A, Townsend G, Bridges P et al. Toba: A Java-to-C translator[EB/OL]. 1999. http://www.cs.arizona.edu/sumatra/toba/.

[5] Laffra C. c2j. A C++ to Java translator[EB/OL]. 2001. http://www.novosoft.us/solutions/tools.shtml.
[6] Tilevich I. c2j++[EB/OL]. 1999. http://www.vbasicmaster.com/html/c2j++.html.
[7] Fox G, Li Xiaoming, Zheng Qiang et al. A prototype of FORTRAN-to-Java converter[C]//in Proceedings of the ACM 1997 Workshop on Java for Science and Engineering Computation. Las Vegas, NV: Syracuse University, 1997.
[8] The UQBT team. Java Backend for GCC[EB/OL]. 2001. http://www.itee.uq.edu.au/cristina/uqbt.html#gcc-jvm.
[9] Harsu M. Re-Engineering legacy software through language conversion[R]. A-2000-8. Tampere: Department of Computer and Information Sciences, University of Tampere, 2000.