

飞行仿真实验中飞机飞行参数与坐标系变换

刘 通, 谢剑斌, 丁文霞, 程江华, 闫 玮, 李沛秦

(国防科学技术大学 电子科学与工程学院, 湖南 长沙 410073)

摘 要: 飞机姿态控制是飞行仿真试验中的一项重要课题, OpenGL 是飞行仿真试验中常用的图形交互平台。要在 OpenGL 平台上实现飞机飞行参数对飞机飞行姿态的控制, 其主要工作是完成飞机飞行参数与 OpenGL 中坐标系的变换。详述了在飞行仿真试验中飞机飞行参数与坐标系变换的相关问题, 并依据 OpenGL 的一些基本变换函数解决了飞机飞行参数与 OpenGL 中坐标系的变换问题, 实现了飞行仿真试验中飞机飞行参数对飞机姿态的控制。

关键词: 飞行仿真; 飞行参数; 坐标系变换; OpenGL

中图分类号: TP391.9

文献标识码: A

文章编号: 1673-629X(2006)10-0191-03

Transform Between Plane's Flight Parameter and Coordinate on Experiment of Flight Simulation

LIU Tong, XIE Jian-bin, DING Wen-xia, CHENG Jiang-hua, YAN Wei, LI Pei-qin

(School of Electronics Engineering, National University of Defence Technology, Changsha 410073, China)

Abstract: The controlling of plane's pose is a very important problem on experiment of flight simulation. OpenGL is the normal graph alternation flat roof on experiment of flight simulation. In order to use flight parameter to control plane's pose on OpenGL flat roof, the most important thing need to do is complete the transform between flight parameter and coordinate of OpenGL. This paper dilates many correlation problems, and resolves the problem according as some basic transform function in OpenGL. In this way, realize the controlling of plane's pose on experiment of flight simulation.

Key words: flight simulation; flight parameter; coordinate switch; OpenGL

0 引言

在飞行仿真试验中, 需要根据飞机的飞行参数完成大地坐标系与飞机机体坐标系的变换, 实现飞行仿真的漫游控制, 从而让飞行员以第一人称漫游 3D 场景。在三维图形系统开发过程中, 常采用虚拟相机方法来实现三维场景漫游, 便于用户和三维场景的交互, 有效的软件工具是 OpenGL。常用的漫游控制方法是利用 OpenGL 库中的旋转平移等函数实现, 而函数的控制参数大多是飞机的飞行参数。在本试验中, PC 实时接收的飞机飞行参数有飞机所处经度、纬度、高度, 飞机的偏航角、俯仰角、横滚角。如何实时响应这些数据, 完成坐标系的变换? 下面来探讨这个问题。

1 坐标系与飞行参数

首先要了解大地坐标系与飞机机体坐标系的关系, 通

过这一关系来了解飞机的 6 个飞行参数(也即飞机的 6 自由度)的意义^[1,2]。

图 1 中, $EXeYeZe$ 为地面坐标系, 面 $EXeZe$ 为水平面, 线 EYe 为铅直线。 $PXpYpZp$ 为机体坐标系, 点 P 为飞行器质心, PXp 指向飞行器机头方向, 面 $PXpZp$ 为飞行器机体平面, 面 $PXpYp$ 为飞行器对称面。

OQ 为铅直面 OPQ 与水平面 $EXeZe$ 的交线; φ 为偏航角, 即机体坐标轴 PXp 在水平面上的投影与地面坐标轴 EXe 的夹角, 规定飞行器向左偏航时 φ 为正, 也即机体坐标轴绕其 Y 轴旋转的角度, 规定逆时针旋转时为正; θ 为俯仰角, 即 PXp 轴与水平面的夹角, 规定飞行器头向上仰时 θ 为正, 也即机体坐标轴绕其 Z 轴旋转的角度, 规定逆时针旋转时为正; μ 为横滚角, 即 PYp 轴与铅直线的夹角, 规定飞行器向右滚转时 μ 为正, 也即机体坐标轴绕其 X 轴旋转的角度, 规定顺时针旋转时为正(如图 2 所示); 机体坐标系原点在大地坐标系下的坐标值分别按一定关系对应于飞行器所处的经度、纬度和高度值。

在 OpenGL 图形系统中, 绘制和显示地形需要将地形数据对应的大地坐标系转化到 OpenGL 的几何建模坐标系, 即世界坐标系中。在屏幕视图中, 世界坐标系原点在视图中心, x 轴方向由左向右, y 轴方向由下向上, z 轴方向

收稿日期: 2006-02-14

基金项目: 航空科学基金资助项目(01C15001)

作者简介: 刘 通(1982-), 男, 河南南阳人, 硕士研究生, 研究方向为三维图形显示与信息可视化; 谢剑斌, 博士后, 副教授, 硕士生导师, 从事数字视频技术、信息可视化技术和图形图像处理的教学与科研工作。

由里向外,是一个右手系。这样飞行器所处的位置也要经过类似变换而换算到世界坐标系下。

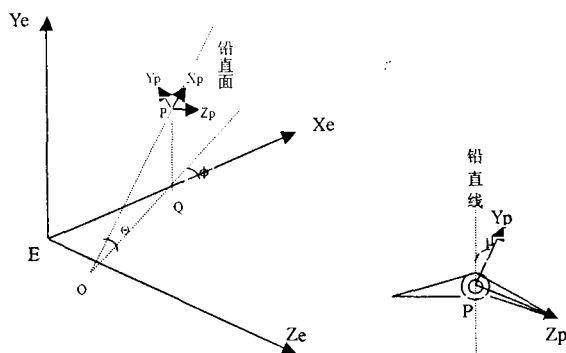


图 1 坐标系与飞行参数

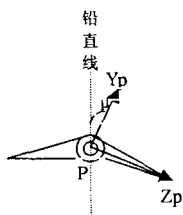


图 2 机体坐标系下的横滚

2 相关图形学及 OpenGL 基础知识

仿真景物是 3D 实物,而计算机屏幕是 2D 画面,要将 3D 实物显示到 2D 画面上而且呈现出立体感,一般可以采用下面的步骤^[3~6]:

a. 视点变换\模型变换(前者是将相机放在合适的地方对准 3D 景物,后者是将 3D 物体放在合适的地方朝向相机,两者本质相同)。

b. 投影变换(调整相机镜头,使 3D 景物投影在 2D 胶片上)。

c. 视口变换(调整胶片大小决定画面尺寸)。

上面的变换在图形学中都是采用矩阵相乘的方法实现的。在 OpenGL 中提供了相应的函数实现这些变换,简要列举如下:

(1) 几何变换。

可用于实现视点变换和模型变换。

`glTranslated(GLdouble x, GLdouble y, GLdouble z);`
`glTranslatef(GLfloat x, GLfloat y, GLfloat z);`

目标沿 X, Y, Z 轴分别平移 x, y, z 。

`glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z);`

`glRotatef(GLfloat angle, GLfloat x, GLfloat y, GLfloat z);`

目标以 X, Y, Z 轴为轴分别逆时针旋转 x, y, z 。

`glScaled(GLdouble x, GLdouble y, GLdouble z);`

`glScalef(GLfloat x, GLfloat y, GLfloat z);`

目标分别在 X, Y, Z 方向缩放,相对应的缩放因子为 x, y, z 。

(2) 投影变换。

a. 正射投影:

`glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far);`

创建一个平行视景体,即投射射线是平行线,把第一个矩形视景投影到第二个矩形视景上。并用这个矩阵乘以当前矩阵,以完成变换。近景第一个矩形的左上角三维空

间坐标(left, bottom, -near),右下角的三维空间坐标(right, top, -near);远景第二个矩形左上角三维空间坐标(left, bottom, -far),右下角的三维空间坐标(right, top, -far)。

b. 透射投影:

`glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble znear, GLdouble zfar);`

创建一个形如棱台的视景体,其近截取面由 left, right, bottom, top, znear 确定,远截取面由从视点投影近截取面到 Z 轴 zfar 位置决定。

(3) 视口变换。

`glViewport(GLint x, GLint y, GLsizei width, GLsizei height);`

这个函数定义一个视口, x 和 y 是视口在屏幕窗口坐标系中左上坐标,缺省是(0,0)。width,height 是宽和高。注意使用中视口长宽比例的调整会导致图像变形。因此 reshape() 中要检测窗口尺寸,修正视口大小,保证图像不变形。

当然,也可以不用这些封装的函数,直接用原始的矩阵乘法进行变换。相关的函数列举如下:

`glLoadMatrixf(fld) (* m);`

它将用数组或指针形式给出的 4×4 矩阵 m 设为当前矩阵,运算后的结果可返回当前矩阵。

`glMultMatrixf(fld) (* m1);`

它将 4×4 矩阵 m_1 与当前矩阵 m 相乘,把结果 $m = m * m_1$ 返还 m 。

3 坐标系变换方法

在飞行仿真中飞行员是主角,所有的景物都要放到飞行员的视野里进行观察,所以有必要进行坐标系变换,把地面坐标系下定义的物体都放到飞行员的观察坐标系下定位,这里取机体坐标系为观察坐标系。把飞机质心假定为相机镜头,也可以说是飞行员的眼睛,现在要做的工作是把飞机相对于景物的运动变换为景物相对于飞机的运动,也就是做模型变换,把地面坐标系下静止的景物变换成机体坐标系下运动的景物,而把地面坐标系下运动的飞机质心变换成机体坐标系下静止的坐标原点,从而实现地面坐标系与机体坐标系的变换。其编程实现如下:

`float tlon, tlat, thei; /* 飞机质心在地面坐标系下的坐标,由经度、纬度、高度三个参数变换得到 */`

`float roll, pitch, heading; /* 分别对应飞机绕地面坐标系三基轴(EX_e, EY_e, EZ_e)旋转的滚转角、俯仰角、偏航角 */`

`glLoadIdentity(); /* 清成单位阵 */`

`glTranslatef(tlon, tlat, thei);`

`glRotatef(heading, 0.0, 0.0, 1.0);`

`glRotatef(pitch, 0.0, -1.0, 0.0);`

`glRotatef(roll, 1.0, 0.0, 0.0);`

为方便使用,可以将上述函数整合到一个函数中:

```
void Tran(float tlon, float tlat, float thei, float roll,
float pitch, float heading)
{
    glLoadIdentity();
    glTranslatef(tlon, tlat, thei);
    glRotatef(heading, 0.0, 0.0, 1.0);
    glRotatef(pitch, 0.0, -1.0, 0.0);
    glRotatef(roll, 1.0, 0.0, 0.0);
}
```

飞机的 6 个飞行参数可以通过网络通信的方式实时传送到程序中,经过必要的变换后按照对应顺序存放数组 plane^[6]中。在仿真程序中,调用函数

```
Tran(plane[0], plane[1], plane[2], plane[3], plane[4], plane[5]);
```

这样就完成了两个坐标系的变换,实现了飞机飞行参数对漫游的控制。

4 实验结论

OpenGL 缺省的坐标系是观察坐标系,所有的景物都要放到观察坐标系下定位。在飞行仿真时,要让飞行员有身临其境的感觉,以第一人称的角色漫游整个 3D 场景,就要建立以飞行员为中心的观察坐标系。飞行仿真中飞

机的飞行参数可以实时获取,可以很方便地由这些可行的飞行参数来制定以飞行员为中心的观察坐标系,在文中就是指机体坐标系。然后利用 OpenGL 中相关的模型变换函数,把地面坐标系下的场景统统搬移到观察坐标系,就完成了坐标系的一个变换过程。

飞机的飞行参数有许多,文中用到的是飞机常用的定位姿态的 6 个自由度,完成坐标变换。根据获取的飞行参数不同,可以采用不同的方法建立观察坐标系,采用不同的 OpenGL 函数实现坐标系的变换。

参考文献:

- [1] 王行仁. 飞行实时仿真系统及技术[M]. 北京:北京航空航天大学出版社,1998.
- [2] 席迎来. 基于 PC 机的飞行器六自由度仿真[D]. 西安:西北工业大学,2003.
- [3] Wright R S, Sweet M. OpenGL 超级宝典[M]. 北京:人民邮电出版社,2001.
- [4] 励映群,王润杰,李松维. Open GL 中实现飞行仿真观察变换的几种方法[J]. 计算机仿真,2003,20(5):71-73.
- [5] 付战平,邸亚洲,尚希良,等. 基于 OpenGL 的飞行过程再现与仿真[J]. 系统仿真学报,2002,14(9):1197-1199.
- [6] 尚 游,陈岩涛. OpenGL 图形程序设计指南[M]. 北京:水利水电出版社,2001.

(上接第 190 页)

除电压稳定时探针间电流随机涨落的影响,在每个扫描点连续测量 1000 次,并将其算术平均值作为最后的测量结果返回上层应用程序。实践表明这种处理获得了良好的效果。

数据处理软件是运行于 Ring3 级的用户态程序,采用 Microsoft Visual C++ 6.0 编写,其主要功能是调用 VxD 驱动程序进行扫描测量,并接受驱动程序返回的测量结果,根据结果作出被测等离子体的伏安特性曲线,最后根据用户的需要计算等离子体的各种特性参数,如电子温度、离子温度等。

为了改善软件与用户的交互速度,数据处理程序采用了多线程技术^[5]。其中主线程创建用户界面,并等待用户的操作请求,响应用户选择菜单、点击按钮和工具条等消息,此外,它还负责测量结果的处理。为了启动一次测量过程,用户首先设置板卡的基地址、扫描的起始电压和终止电压,然后选择相应的菜单项或者按钮启动测量,此时主线程打开设备加载驱动程序,创建一个事件,并调用未公开的 kernel32.dll API 服务 OpenVxdHandle(),将其转换为 ring0 级可用的事件句柄。接着主线程将这些参数通过 BEGIN_SCAN Ioctl 传递给 VxD,并创建一个辅助线程,然后主线程进入消息循环等待处理用户的其他操作。

辅助线程专门负责与驱动程序的数据交换。它从用户设定的起始扫描电压开始,发送 MEASURE Ioctl 请求

后就调用 WaitForSingleObjectEx 挂起;等驱动完成该点的电压电流测量后,线程被唤醒接受结果数据,并发送消息通知主线程刷新显示,将当前的测量结果显示在屏幕上;再进行下一个扫描点的测量,重复此过程直到用户设定的终止电压。

3 小 结

扫描电源是 Langmuir 探针诊断技术的关键设备。文中介绍了一种使用简便、成本低廉的扫描电源的设计,详细阐述了系统硬件和软件设计中的关键技术。目前,该套设备已经成功应用于等离子体物理教学与科研工作中,获得了用户的好评。

参考文献:

- [1] 谭永胜,王景峰,刘支华. Langmuir 探针电源的研究[J]. 长春理工大学学报,2003,26(1):63-64.
- [2] Apex Microtechnology. PA42 Datasheet[EB/OL]. <http://www.apexmicrotech.com>,2001.
- [3] 武安河,周利莉. Windows 设备驱动程序 VxD 与 WDM 开发实务[M]. 北京:电子工业出版社,2001.
- [4] Hazzah K. Windows VxD 与设备驱动程序权威指南[M]. 北京:中国电力出版社,2001.
- [5] Richter J. Windows 高级编程指南[M]. 北京:清华大学出版社,1999.