

基于 OpenGL 三维飞行模拟的实现

马存宝,朱 超

(西北工业大学 航空学院,西安 710072)

摘要:实现操作面、起落架的运动能够真实的模拟飞机飞行过程。利用 3DSMAX 软件构建民机的各个部件的三维模型,并将其直接导入到 OpenGL 中。针对动画效果的逼真性,通过使用 OpenGL 建立地形和天空的三维模型,在实现地形时运用了颜色缓冲器、深度缓冲器,在绘制天空时运用了纹理贴图技术。使用鼠标按键输入操作实现画面的放大缩小可以更加清楚的了解操作面的状态。在此基础上调用飞参数据和使用双缓冲技术实现了飞机的三维飞行模拟。

关键词:三维建模;OpenGL;3DSMAX;飞行再现;双缓冲技术

中图分类号:TP391.9 **文献标志码:**A **文章编号:**1000-0682(2014)03-0033-04

The implementation of 3D flight simulation based on the OpenGL

MA Cunbao, ZHU Chao

(School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract: Achieving operating surfaces and landing gear movement can realistic simulation of aircraft flight. Using 3DSMAX software to build a three-dimensional model of the various components of civil aircraft, and directly imported into OpenGL. A more realistic animation by using OpenGL to create a three-dimensional model of the terrain and sky. Terrain use the color buffer, depth buffer and the use of texture mapping technology in the draw of the sky. Using the state of the mouse button input operation screen zoom can be a more clear understanding of the operating surface. On this basis, by calling flight parameter and using double buffering technology to achieve a three-dimensional aircraft flight simulation.

Key words: three-dimensional modeling; OpenGL; 3DSMAX; flight reproduction; double buffering technology

0 引言

随着计算机技术和网络技术的飞速发展,计算机仿真技术和虚拟现实仿真在各行各业中得到了广泛的应用。对于飞行器运动的仿真,需要形象直观的表达。而虚拟现实技术恰好能很好的解决这个问题。它可以非常直观的连续实时地再现飞行的场面,同时给一种身临其境的感觉^[1]。目前民航同样也利用虚拟现实技术来实现仿真。当飞机在完成某次飞行或出现事故后,可以利用飞参数据,实现飞行过程的分析,这样使飞机的飞行再现能够更加真实的反映飞行过程,对航空公司航班的飞行品质监控或事后处理分析有着重要意义。该文主要以计算

机图形学中坐标变换、三维投影变换、纹理映射等为理论基础,运用 OpenGL 这个功能强大的开放式图形软件包,借助 VC 为平台,从而模拟了飞机在三维空间飞行的全过程。

1 开发平台介绍

OpenGL (Open graphic library) 是一个三维的计算机图形和模型库。它是一种高性能的开放式且功能强大的 3D 图形库,具有几百个指令和函数。使用这些指令和函数可以构造出高质量的静止物体模型和动态图像,并且可以实现对模型的实时交互操作控制^[2]。

3DSMAX 是 Autodesk 公司开发的三维建模、动画制作软件,利用该软件能够快速构造复杂的三维模型,并加以材质、贴图、纹理、灯光等效果。此外,3DSMAX 软件方便易学,不需要编程便可很直观地构造模型,模型的外观更精细^[3]。该文结合这两种

收稿日期:2013-08-28

作者简介:马存宝(1963),男,陕西三原人,博导,副院长,教授,研究方向为机载系统仿真技术,测控技术。

软件,先用三维建模软件建立模型,再把这些模型导入到 OpenGL 程序,对其进行交互控制。

2 三维建模

2.1 飞机三维模型的建立

2.1.1 飞机的三维建模

该文应用飞行记录器记录的飞参数据来实现三维飞行的模拟,概括来说飞行的模拟分为两种运动:一个是飞机整体的飞行运动,另一个是飞机起落架、操纵面等的运动过程。前者需要建立飞机的机体模型,后者的运动是与机身分开的,是单独运动的,所以在建立模型过程中需单独制作起落架和操作面等的三维模型,并按系统约定在配置文件中指出起落架和操作面等模型的文件名。

再现所用的飞机模型为 3DS 格式的三维模型,飞机的 3DS 模型需按如下标准制作:要求飞机的模型大小比例为 1:1,即每个几何单位代表 1 m,机身主轴线平行于 Y 轴,机头朝向 Y 轴正向,飞机纵对称面平行 Z 轴,飞机面向 Z 轴正方向。图 1 所示即是运用 3DSMAX 软件制作的 B737 的飞机模型。建立模型后,给模型加上材质和贴上相应的纹理,最后将所建模型输出为 3DS 模型格式文件^[4]。



图 1 3DSMAX 制作的三维模型

2.1.2 三维模型的导入显示

3DS 文件是由许多块(chunk)组成的(大块中镶嵌子块)。一个完整的块由块信息和块数据组成。块信息又由块的 ID(2 个字节长的标识如 4D4D)和块的长度(4 个字节其实也就是下一个块的偏移字节数)组成。由于在网上有很多现成的导入 3DS 程序,所以完全可以找一个类封装得比较好的程序,将其移植到自己的工程中就行了^[5]。系统中从网上下载了 3ds.cpp 和 3ds.h 两个文件,加载到程序中。系统中在窗口的初始化中调用自己定义的函数 load3dobj(char * dir, char * cn, int a) 来导入 3DS 文件。其中 dir 是 3DS 文件保存路径, cn 是 3DS 文件的名称, a 表示导入模型的标号。系统导入“前轮.3DS”、“后轮.3DS”、“机身.3DS”、“左水平尾.3DS”和“右水平尾.3DS”等

飞机各个能单独活动的部件

读入 3DS 文件后,就要在视口中将模型绘制出来。使用显示列表技术可以方便、高效地组织 OpenGL 函数,存储模型的绘制方法。特别是对于要不断绘制飞机模型的情况,显示列表可省去大量的重复性工作,以提高系统的快速性。光照的运用能反映出模型的材质和纹理,生动地再现飞机模型。在程序中需要显示模型的地方调用 show3ds(int j0, float tx, float ty, float tz, float size) 来显示 3DS 模型。其中 j0 表示模型的标号, tx, ty, tz 表示飞机的位置, size 表示飞机模型的大小。

2.2 三维场景的建模

飞行三维动画仿真不仅要绘制飞机的三维模型,还应当有飞机飞行的环境才能更形象地衬托出飞机飞行状态。这个环境至少应当包括地面和天空。系统摒弃使用天空盒固定范围场景,而采用随机地形与圆形天空结合的方法以便使得场景可以无限延伸。绘制场景时,OpenGL 的纹理映射技术得到大量运用。它可以使得枯燥的多边形变成生动的自然物体^[6]。

2.2.1 三维地形的建模

地面是一个复杂的曲面,要在虚拟世界中构造凹凸错落的地面,方法有多种。系统采用网格地形来模拟真实地面,具体方法为:先生成网格,再对每个网格点赋上随机高程值,然后对过于突兀的地形高程数据进行平滑处理,最后在它的外表面贴上适当的纹理。此过程具体步骤如图 2 所示。

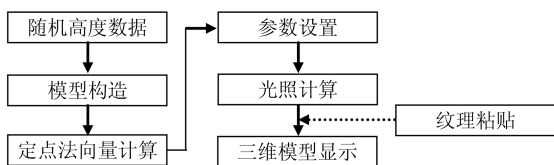
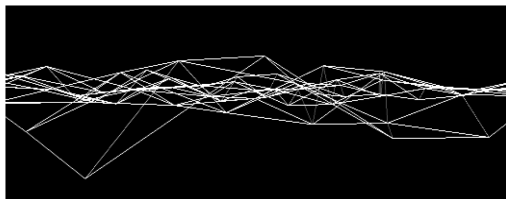
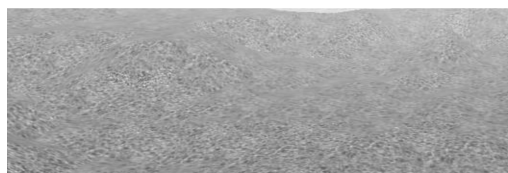


图 2 三维地形的产生基本步骤

网格点高度的算法:假设地形网格矩阵由二维数组 field[m][n] 表示,则网格上某一点 field[i][j] 的高程值 field[i][j].y 是随机产生的。结果如图 3a 所示。粘贴纹理:用三角形近似出地形曲面,并获得各三角形面的法向量。法向量的方向决定了纹理粘贴到平面的内侧还是外侧。根据需要,应在地形曲面的外侧粘贴纹理^[7]。结果如图 3b 所示。



a 随机高度产生的地形框架



b 粘贴纹理后的地形

图3 地形的生成过程

OpenGL 函数进行纹理贴图的要求:

①纹理定义 纹理图像的尺寸必须是2的幂,且不能大于1 024,用 `glTexImage2D()` 函数说明所映射的纹理内容。

②纹理控制 说明纹理以何种方式映射到三维模型表面上。OpenGL 提供了多种映射方式,其中包括纹理滤波、重复与伸缩,其函数为 `glTexParameterf()`。

2.2.2 三维圆形天空的建模

天空的绘制主要是绘制一个半径比较大的球,而其余所有的场景和模型均置于球的内部,这样在球的内部看,球就类似于一个无穷大的天空。完成后,还需要经过纹理映射使场景看起来贴近现实^[7]。

`void gluSphere (quadratic, GLdouble radius, Glint slices, Glint stacks)` 绘制一个球形。`quadratic` 指明是哪个二次对象、`radius` 指球体半径、`slices` 指围绕 z 轴分片的个数、`Stacks` 指顺着 z 轴分片的个数。

`GLUquadricObj * quadratic = gluNewQuadric ();`

`gluNewQuadric` 创建一个二次对象。这个函数创建并返回一个指向新的二次对象的指针。当调用二次描述和控制函数时指向这个对象。如果返回值为0则表明没有足够的空间分配给该对象。完成绘制后调用函数 `glBindTexture (GL_TEXTURE_2D, texture[1])`, 这样将预先存储在 `texture[1]` 的纹理贴图加载到绘制出来的天空框架中,最后完成了天空的建模。为了使天空更加真实可以添加一些雾,通过调用函数

`glFogf (GL_FOG_START, MAX * 2);` //雾的开始深度

`glFogf (GL_FOG_END, MAX * 15);` //雾的结束深度

3 三维飞行模拟的实现

利用 VC + + 6.0 搭建 MFC 单文档应用程序,并按上文提到的方法构建系统框架。绘图工作分为两部分:非实时的和实时的。非实时部分是指绘图

的初始化,如:场景的初始设置、读入飞机模型等;实时部分是指当外部实时数据到来时触发窗口的重绘,包括绘制飞机模型、地面、天空。具体的工作框图如图4所示。

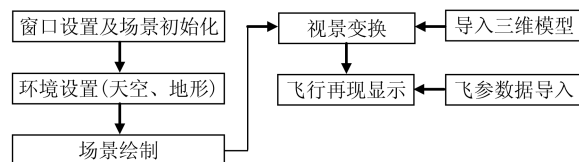


图4 三维飞行模拟实现框图

动画显示采用双缓冲技术,它能够提供前后两个缓存,在显示前台缓存内容的一帧图像时,后台正在绘制下一帧的图像;后台绘制完成后,交换前后台缓存,不断地循环,实现地形的实时动态显示。

3.1 视景变换

创建完模型后,只有放在三维空间中的适当位置,并选择适当的视点及观察方向才能更有利于用户观看;为了观察场景中的模型,需要进行视景转换,包括视点转换、模型转换、投影转换和视窗转换^[8]。主要有以下几个步骤:

1) 执行视点变换,在三维场景中定位观察物体的视点位置和方向。调用函数 `gluLookAt (eyex, eyey, eyez, centerx, centery, centerz, upx, upy, upz)` 来指定视点变化,以照相机为例,参数 `eyex, eyey, eyez` 指定观察点,也就是照相机放置的位置。参数 `centerx, centery, centerz` 指定观测场景的参考点,表示照相机准备拍摄的物体的位置。参数 `upx, upy, upz` 指定视图体自下而上的方向,通常将该方向设置为指向 Y 轴的正方向,表示照相机怎么放置。

2) 利用模型变化命令 `glTranslatef (x, y, z)` 和 `glRotatef (angle, x, y, z)` 进行平移和旋转两种操作,确定物体在场景中的位置和方向。

3) 把三维模型投影到二维屏幕上,即投影转换。一是确定物体投影到屏幕上的方式,是透视投影还是正投影。二是确定场景中有哪些物体要显示在最终的屏幕区域内。利用函数 `gluPerspective ()` 和 `glOrtho ()` 定义投影方式:透视投影和正投影。透视投影同现实生活中人们看到的景物效果一样,距离视点越远的物体看起来越小,距离视点越近的物体看起来越大,符合人们的视觉习惯。透视投影的应用比较广泛,通常用到视景仿真和模拟真实场景的应用程序中。正投影中无论视点到视景体的距离多远,经过投影后,物体的大小总是不变的。因此在三维地形可视化中采用透视投影。

4) 利用 `glViewport ()` 来确定视窗的大小,规定

屏幕上显示场景的范围及尺寸,通过调整这个可以调整视窗的大小和位置。

3.2 飞机整体运动

文中三维飞行动态显示中采用目标固定也就是飞机固定,场景在动。飞机在机体坐标系中实现绕三轴的角运动和飞机线运动则由地面坐标系实现,即用场景的移动来衬托飞机的移动。负责角运动的核心代码为:glRotatef(angle, x,y,z);负责线运动的核心代码为:glTranslatef(x,y,z);飞机运动如图 5 所示。

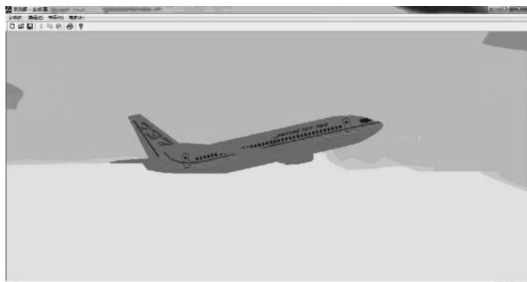


图 5 飞机的三维飞行状态

飞机在飞行过程中,飞行范围相当大,但如果制作如此大的地形就会大大耗费计算机资源而影响系统的实时性。为了解决这个矛盾,可将飞机置于方形区域中心,获得并判断它做线运动时的位移量,一旦超出地形的范围,则将位移置零。这样从视觉效果上看,地面得以无限延伸,而实际上飞机无论怎样“飞”都没有飞出这个地形区域^[9]。

3.3 飞机操纵面的运动

系统不但实现飞机整体的运动包括俯仰、滚转和偏航,而且完成了起落架、襟副翼、扰流板、缝翼、升降舵和水平舵的运动。而相比这两种运动,后者的运动实现难度更大。系统采用的方法是分别对其进行建模后导入到 OpenGL 中。前面已经实现模型建立与导入,但是由于它们旋转轴不一样,所以先找出它们旋转轴,然后通过调用 glRotatef() 来实现旋转。具体方法是:假设绕 AB 轴旋转,A 点在世界系的坐标为(a,b,c),B 点在世界坐标系的坐标为(d,e,f),先用 glTranslatef(a,b,c)将原点移到 A 点,用 glRotatef(angle, d - a, e - b, f - c) 旋转,再用 glTranslatef(- a, - b, - c) 平移回原来世界坐标系的原点即可,也可以移动 B 点。这样就实现了起落架、襟副翼、扰流板等飞行过程中的旋转,使飞行再现更加真实,更能从一定程度上观察出飞行中或许出现的故障。图 6 表示飞机襟副翼等的运动。



图 6 飞机襟副翼等旋转

3.4 人机交互的实现

交互式不仅允许用户指定屏幕上的物体,而且还可以通过鼠标或键盘移动,选择或用其他方式操作这些物体,以实现不同角度、不同方位、不同距离的观察,方便用户变化角度进行观察。能够交互式地从各个不同的角度形象、直观地展示三维地形,进行三维地形可视化,具有很大的实用价值^[10]。系统中用 m_Keys.IsPressed() 来定义键盘响应的交互事件,J、K 分别定义抬高、降低视点,W 前移视点、S 后移视点,A 左移视点、D 右移视点。SHIFT 加速视点变换的速度。通过调用 Camera 类中的 setViewByMouse() 函数来定义鼠标移动来改变视点的旋转。通过这些操作可以清楚地观察飞机各个时刻襟副翼、起落架等的状态。

4 结束语

该文建立了包括飞机机体、起落架、操作面等完善的三维模型,在 Windows 环境下利用 VC++ 与 OpenGL 实现了飞机以及飞行场景(地形、天空)的三维模型及飞行模拟程序,并利用记录的飞参数据实现了飞机的三维飞行模拟。该程序对航空公司航班的飞行品质监控与事故分析有重要的意义。

参考文献:

- [1] 夏辉. 基于 OpenGL 模拟飞机飞行动画的研究[D]. 西安:西安电子科技大学,2006.
- [2] 李丽荣,沈春林,陆宇平,等. 基于 OpenGL 的飞行器超低空追击/拦截三维可视化仿真系统[J]. 南京航空航天大学学报, 2005, 37(3):269-273.
- [3] 王珂. 3DSMAX 2012 中文版从入门到精通[M]. 北京:人民邮电出版社,2012.
- [4] 于莹潇,李嘉林. 基于 OpenGL 的空战飞机三维飞行轨迹显示[J]. 电脑开发与应用, 2004(9):35-37.
- [5] 刘芳,刘贤梅. 3ds 文件读取、绘制与控制方法研究和应用[J]. 计算机工程与设计, 2009(30):4575-4578.
- [6] 和平鸽工作室. OpenGL 高级编程与可视化系统开发. 高级编程篇[M]. 北京:中国水利水电出版社,2006.
- [7] 王新波,朱维杰. 基于 OpenGL 与 3DSMAX 的三维场景建模[J]. 电子科技, 2012(25):79-80.
- [8] 杨柏林,陈根浪,徐静. OpenGL 编程精粹[M]. 北京:机械工业出版社,2010.
- [9] 杨海林,郭爱芳,董伟凡,等. 基于 3DS 模型的飞行再现设计[J]. 计算机仿真, 2004,21(3):71-73.
- [10] 和克智. OpenGL 编程技术详解[M]. 北京:化学工业出版社,2010.