

一种虚拟现实建模的新方法*

邵延华, 吴斌, 张红英, 楚红雨, 方艳红
(西南科技大学 信息工程学院, 四川 绵阳 621010)

摘要: 介绍了 OpenGL 和 Matlab2009a 各自的特点, 在结合各自特点基础上, 建立了基于二者的虚拟现实三维建模开发平台; 以建立的虚拟现实建模开发平台为例, 对比已有的使用 Matlab 外部接口的方法, 证明了基于 OpenGL 和 Matlab 开发方法的优越性和有效性。

关键词: OpenGL; Matlab; 三维建模; 虚拟现实

中图分类号: TP391

文献标识码: A

A new method for virtual reality modeling

SHAO Yan Hua, WU Bin, ZHANG Hong Ying, CHU Hong Yu, FANG Yan Hong
(School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China)

Abstract: Firstly, characteristics of OpenGL and Matlab2009a are introduced. Secondly, on the base of OpenGL and Matlab, an experimental platform of three-dimensional modeling is created in virtual reality. At last, compared with existing methods, which use Matlab's external interface, the validity and advantage of the method is demonstrated by creating a three-dimensional modeling platform of virtual reality.

Key words: OpenGL; Matlab; three dimension modeling; virtual reality

VR(Virtual Reality)技术的主要目标之一是允许用户以尽可能自然的方式与虚拟世界直接交互。可以简明定义: 虚拟现实是计算机生成的给人多种感官刺激的虚拟世界(环境), 是一种高级的人机交互系统^[1]。近年来, 虚拟现实的概念已经深入人心, 因此, 作为虚拟现实技术的重要环节——三维物体的建模技术成了近年来的研究热点。

但是虚拟现实中的物体对象的绘制和运动, 涉及到大量的数学运算, 特别是矩阵运算。作为图形硬件的软件开发接口 OpenGL 已经成为事实的工业标准。在选用 OpenGL 的基础上, 现有的建模方法要么单独利用高级语言; 要么需要复杂的设置利用 Matlab 和其他高级语言的接口进行混合编程; 要么算法方面利用 Matlab 进行仿真, 实际构建虚拟环境时再用其他高级语言另起炉灶构建平台, 增加了开发的复杂度^[2-4]。本文提出一种新的基于 OpenGL 和 Matlab 进行建模的方法, 它充分利用了 Matlab 的高速算法验证优势和 OpenGL 的三维建模优势。

在本文创作时, 国内还几乎没有介绍相似方法的文献公开发表。本文提出的开发方法仅在 2009 年 3 月 6 日发布的最新版本 Matlab2009a 或后续版本平台上可以实现。

1 Matlab、OpenGL 和 Tao 介绍

本文实现方法选用 Matlab 和 Tao 类库中的 OpenGL 实现作为系统实现的软件平台。

1.1 OpenGL 简介

OpenGL 是由美国 SGI 公司推出的开放式图形硬件的软件开发接口(目前由 Khronos 小组负责), OpenGL 独立于窗口系统和操作系统, 在各种平台之间方便移植, 能灵活方便地实现二维和三维的高级图形技术。它提供了建模、变换、光照处理、色彩处理、动画以及其他更先进的功能, 例如纹理映射、运动模糊和雾化效果等, 为实现逼真的三维绘制效果及建立交互式三维场景奠定了良好的基础。OpenGL 以其高性能的交互式三维图形建模能力和易于编程开发等优越性, 已经成为一种事实高性能图形和交互性场景处理的工业标准,

* 基金项目: 国家自然科学基金(60802040)

图形、图像及多媒体

Image Processing and Multimedia Technology

是从事三维图形开发工作的必备工具^[4]。

OpenGL 必须依赖于一定的软件开发环境才能进行软件开发，目前几乎所有的应用程序开发工具都支持 OpenGL 的集成，但是绝大部分开发人员都出于执行效率考虑而采用 C/C++ 等语言平台作为开发工具。

1.2 Matlab 简介

Matlab 由美国 MathWorks 开发推出的，作为科学计算应用软件的突出代表，已发展成为应用学科计算机辅助分析、设计、仿真及教学不可缺少的基础软件。在 Matlab 环境中进行动态仿真，用户可借助 Simulink 软件包方 Matlab 作为工程计算及数据分析软件，它的应用范围涵盖了当今所有的工业、电子、医疗、建筑等各个领域。通过 Matlab 可以在短小时内完成复杂的科研难题，能迅速地测试构思，综合评测系统性能，并能快速设计出更多解决方案确保更高的技术要求。Matlab 还有出色的图形处理功能，三维仿真形象逼真，是优秀的动态仿真软件^[5]。

1.3 Tao 类库简介

Tao 框架类库是为多媒体开发而产生的一个跨平台的类库，可以运行在 Windows 和 Linux 平台。它基于著名的 .NET 和 Mono 项目。其最新版本是 2.1.0，内部集成了 Tao.OpenGl 2.1.0.12、Tao.Platform.Windows 1.0.0.5、Tao.FreeGlut 2.4.0.2 等重要的类库，为本文开发方法的实现提供了基础。

2 基于 OpenGL 和 Matlab 的建模过程

2.1 编程环境设置

传统上采用 Visual C++ 结合三维图形开发接口 OpenGL 作为软件开发平台，设置包括添加 OpenGL 的 4 个静态库文件 glaux.lib、glu32.lib、glut32.lib、opengl32.lib；然后在视图类的 cpp 文件中加入以上静态库的头文件：gl\gl.h、gl\glaux.h、gl\glut.h。由于微软提供的 OpenGL 版本较低，可能有部分功能无法使用，还需要手动升级库文件，其设置比较繁琐。下面详细分析在 Matlab2009a 平台上设置 Tao.OpenGl 并进行开发的过程。

在当前路径中建立 TaoLight 文件夹，并将 Tao.OpenGl.dll 和 Tao.Platform.Windows.dll 复制到该文件夹中；建立 Functions 文件夹，放置配置文件及自编的函数。为了提高程序的可移植性，把程序集的动态链接库放入自建文件夹。更一般的方法是到 <http://www.taoframework.com/> 下载 Tao Framework 2.1.0 并安装。

(1) 建立 LoadTaoOpenGl 的 M 文件，写入以下内容^[6]：

```
% 设置 TAO OpenGL 程序集 %
functionname='LoadTaoOpenGl.m';
functiondir=which(functionname);
% 得到当前目录 %
functiondir=functiondir(1:end-length(functionname));
```

```
disp('Ignore next 2 warnings (bug in R2009a)');
NET.addAssembly([functiondir 'TaoLight/Tao.OpenGl.dll']);
NET.addAssembly([functiondir 'TaoLight/Tao.Platform.Windows.dll']);
disp('Tao OpenGL .NET assemblies loaded');
% 添加函数和数据文件路径 %
addpath([functiondir 'Functions']);
addpath([functiondir 'ExampleData']);
disp('function and example data Path added');
```

(2) 在 Functions 文件夹中编制 OpenGL_Window 函数，进行一些环境设置，例如完成窗口绑定及鼠标操作的映射设置。绑定鼠标移动操作的代码如下：

```
addlistener(simpleOpenGLControl1,'MouseMove',eval(['@(src,evt)OpenGL_Window("MouseMove",src,evt,' str_id ')]))
主要参考 Tao 类库的说明文档、微软的 MSDN 和 Matlab 的用户手册，这里不再详述。
```

(3) 在 Functions 文件夹中编制 OpenGL_Matlab_Figure 函数，除进行相关的图形控制设置外还进行以下操作：

添加必须的 Windows 程序集：

```
NET.addAssembly('System');
NET.addAssembly('System.Windows.Forms');
```

创建一个新的窗口(.NET window):

```
Form1=System.Windows.Forms.Form;
Form1.Width=data.Width;
```

```
Form1.Height=data.Height;
```

```
Form1.Visible=false;
```

```
Form1.Text='SWUST- 邵延华 ';
```

// 所创建窗体的标题条

为 .NET 窗口创建一个 OpenGL 控制集：

```
simpleOpenGLControl1 = Tao.Platform.Windows.SimpleOpenGLControl;
```

设置 OpenGL 控制集 simpleOpenGLControl1 的参数：

```
simpleOpenGLControl1.AccumBits = 0;
```

//accumulation buffer depth

```
simpleOpenGLControl1.AutoCheckErrors = false;
```

```
simpleOpenGLControl1.AutoFinish = false;
```

```
simpleOpenGLControl1.AutoMakeCurrent = false;
```

```
simpleOpenGLControl1.AutoSwapBuffers = false;
```

```
simpleOpenGLControl1.BackColor = System.Drawing.Color.Black;
```

```
simpleOpenGLControl1.ColorBits = 32;
```

//color buffer depth

```
simpleOpenGLControl1.DepthBits = 24;
```

//depth buffer (Z-buffer) depth

图形、图像及多媒体

Image Processing and Multimedia Technology

```

simpleOpenGLControl1.Location = System.Drawing.Point(1, 1);
simpleOpenGLControl1.Name = 'simpleOpenGLControl1';
the height and width of the control:
simpleOpenGLControl1.Size = System.Drawing.Size(Form1.
Width-17, Form1.Height-38);
simpleOpenGLControl1.StencilBits = 0; //stencil buffer
the tab order of the control within its container:
simpleOpenGLControl1.TabIndex = 0;
初始化控制集:
simpleOpenGLControl1.InitializeContexts();
添加控制集到图形实例:
Form1.Controls.Add(simpleOpenGLControl1);
等待初始化:
pause(1)
为使用 Tao.Opengl 类库的 Opengl 函数, 载入所需要的
的程序集:
import Tao.OpenGL.*

```

2.2 三维物体显示过程

将所创建的虚拟环境中的三维物体在计算机屏幕上显示出来是一个从三维空间到二维平面的投影过程, 一般都要经历下面一系列步骤^[1]: 视点变换(Viewing Transformation), 即调整视点的位置; 模型变换(Modeling Transformation), 即对模型进行旋转、平移和缩放; 投影变换(Projection Transformation), 即把三维模型投影到二维屏幕上的过程; 视口变换(Viewport Transformation), 即规定了屏幕上显示场景的位置和尺寸。

通过上面的几个步骤, 一个三维空间里的物体就可以在二维的电脑屏幕上正确显示了。三维物体的显示过程如图 1 所示。

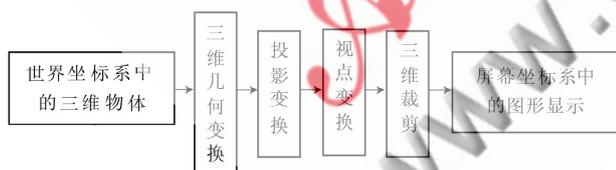


图 1 三维物体的显示过程

2.3 快速绘制技术

在计算机中模拟复杂场景动作时, 会降低渲染速度, 因而造成画面的不连续。原因有很多, 本文系统的延时主要是当仿真机器人运动、碰撞和其他场景更新时, 会出现处理延迟。这是因为机器人运动的过程中, 每一个关节都涉及局部坐标和全局坐标的相互转化, 需要引入大量的矩阵运算。此情况下, 使用有关的快速绘制方法以帮助提高渲染速度显得尤为重要。

目前复杂场景快速绘制主要有下面三大类方法: 可见性判断、层次细节 LOD(Level of Detail)和基于图像

的绘制。

不可见面的剔除是根据场景中几何对象的空间关系, 对物体之间的遮挡关系进行估计。可以简单分为三大类: 背面剔除(Back-Face Culling)、视域剔除(View Frustum Culling)和遮挡剔除(Occlusion Culling), 这个过程在真正的消隐处理之前, 对不可见面的大量删除省去了很多后期处理时间。避免绘制无用的、对于视点而言的不可见区域。本文使用可见性判断来加速绘制场景。

2.4 程序实例

通过上文所述设置, 编写 M 文件调用 OpenGL 相关库函数, M 文件主要功能有: 设置背景色、设置颜色和深度缓存、设置深度测试、读取图像、启用纹理和光照、绘制三维模型。通过图 1 所示变换把现实世界中的三维物体以二维的平面形式显示在计算机屏幕上, 绘制出逼真的三维实体模型, 如图 2 所示。其操作和其他高级语言平台利用 OpenGL 建模基本一致, 在此不再详述。

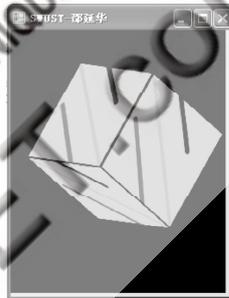


图 2 虚拟环境中所建放的三维图形

三维建模对提高虚拟环境的真实感有至关重要的作用, 是构建虚拟环境最重要环节。本文所提出的一种新的三维建模方法, 可有效地应用于虚拟现实环境的构建和计算机图形学算法的验证, 基于 Matlab 和 OpenGL, 通过运用快速绘制算法, 建立三维仿真场景。仿真结果表明, 本方法能充分利用 Matlab 高速数学运算、便利图形场景绘制和事实的工业标准 OpenGL, 无论从降低开发设置的难度上还是对新算法开发验证效率上都增强了三维建模的效力。所以, 对三维建模而言, 本方法极具应用价值。

参考文献

- [1] 石教英.虚拟现实基础及实用算法[M].北京:科学出版社,2002.
- [2] 石琼,沈春林,谭皓.基于OpenGL的三维建模实现方法[J].计算机工程与应用, 2004,40(18):122-124.
- [3] 翟雪琴,郝矿荣,曹自洋.基于OpenGL的工业机器人动力学仿真的研究[J].机床与液压,2004,17(1):14-15.
- [4] 徐波译.OpenGL编程指南(第六版)[M].北京:机械工业出版社,2008.
- [5] MATLAB builder NE user's guide version 3.1 (Release 2009a). The MathWorks, Inc. 2009:44-47.

(收稿日期: 2009-05-20)