

# 基于 OpenGL 的新型水雷三维仿真训练系统<sup>①</sup>

周 杰, 高洪林, 周文林

(海军潜艇学院, 青岛 266071)

**摘 要:** 为克服我国新型水雷开展训练面临的困难, 介绍了采用 Visual C++ 及 OpenGL 构建某型水雷三维仿真训练系统的方法和过程。重点阐述了如何在 OpenGL 开发环境中实现对三维模型的交互控制。训练系统的建立证明了方法的可行性。

**关键词:** 三维仿真; 交互控制; OpenGL 技术; 开发环境; 模型显示

## 3D Training System for New Mine Based on OpenGL

ZHOU Jie, GAO Hong-Lin, ZHOU Wen-Lin

(Naval Submarine Academy, Qingdao 266071, China)

**Abstract:** In order to overcome the difficulties in new mine training, the method and course of building the 3D training system for new mine based on Visual C++ and OpenGL are introduced. The paper concentrates on how to realize the interactive control of 3D objects in the OpenGL program developing environment. The process of building the training system demonstrates the feasibility of the method.

**Keywords:** 3D simulation; interactive control; OpenGL technology; program developing environment; model showing

## 1 引言

我国某新型水雷具有自动化程度高、战斗准备迅速和性能稳定可靠的特点。但由于装备结构复杂, 使用人员需要经过系统培训, 但训练用水雷数量有限, 致使快速提高训练、使用水平有一定的困难。三维仿真训练系统可以有效的解决训练过程中面临的困难。

本系统是在 Visual C++6.0 环境下, 利用 MFC 平台搭建 OpenGL 工程来实现的。OpenGL 自从 1992 年诞生到现在, 已经成为业内最为广泛使用和支持的 2D 和 3D 图形应用程序接口, 被应用在不同的计算机平台上去开发各种各样的应用程序。OpenGL 具有一套渲染、纹理贴图、特效和其他功能强大的显示函数, 加速了应用程序的开发。功能强大 OpenGL 可以在所有流行的个人计算机和 workstation 平台上开发和使用的, 确保了程序有良好的可移植性。OpenGL 的特点是可靠度高、可扩展、可伸缩、灵活、容易使用等。借助 Windows 编程环境及利用 OpenGL 的变换, 色彩处理, 光线处理, 动画及纹理映射等功能进行某型水雷三维

仿真训练软件的开发可实现短小精悍、性能优异的三维仿真工具, 为武器使用训练提供一个专门的可交互的三维仿真手段。

## 2 OpenGL 简介

OpenGL 是美国高级图形和高性能计算机系统公司 SGI 所开发的三维图形库, 其显著优点是独立于硬件设备、窗口系统和操作系统, 目前已经成为事实上的高性能图形和交互式视景处理的标准。在仿真三维建模, 虚拟现实技术等方面, OpenGL 起了极其重要的作用。用户可以利用 OpenGL 图形库, 创建出接近光线跟踪的高质量静止或活动的三维彩色图像<sup>[1]</sup>。

OpenGL 渲染使用一种流水线结构。OpenGL 使用 4 个主要阶段来处理像素和顶点数据: 顶点操作、像素操作、光栅化和片元操作, 并将结果存储在帧缓存和纹理内存中, 如图 1 所示。

应用程序将两种数据传递给 OpenGL 进行渲染: 顶点数据和像素数据。它们使用顶点数据来渲染 2D

<sup>①</sup> 收稿时间:2010-07-22;收到修改稿时间:2010-10-01

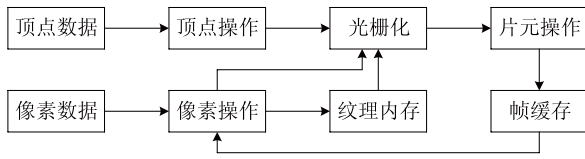


图 1 OpenGL 流水线结构

和 3D 几何图元，如点、直线和填充图元。应用程序以像素数组的方式指定像素数据，并命令 OpenGL 将像素直接显示到帧缓存中，或者将其复制到纹理内存中供以后用作纹理图。另外，应用程序还可以从帧缓存中读取像素数据，或者将部分帧缓存复制到纹理内存中<sup>[2]</sup>。

### 3 系统总体设计与功能

系统在 Windows XP 平台上，使用 VC++6.0 与 OpenGL 结合的编程仿真，通过 3DS MAX 三维软件建立三维模型，再将模型导入仿真系统中，进行人机交互控制。通过对各个三维仿真部件和工具进行鼠标选取、平移、旋转操作，结合系统预先设计的视点转换和安装、拆卸动画，达到对某型水雷的各种技术准备操作进行三维仿真训练的目的。模拟训练过程中，用户按操作步骤使用鼠标对零部件和相关工具进行选择，并进行移动和旋转，使其以规定姿态到达指定位置，完成仿真训练。其系统总框架如图 2 所示。

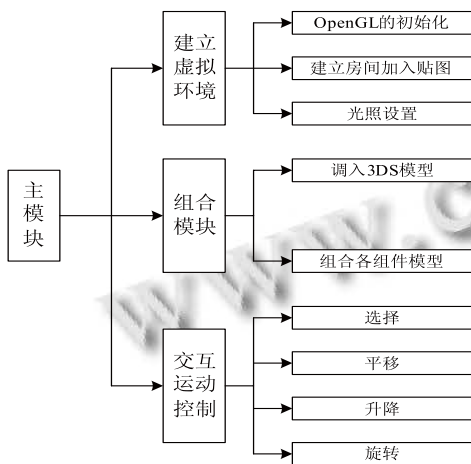


图 2 系统框架图

## 4 系统关键技术

### 4.1 构建 OpenGL 开发环境

在所有的 Windows 应用程序开发工具当中，

VC++6.0 已经成为 OpenGL 图形应用的首选开发环境。由于 OpenGL 类似状态机，它的各种状态（或模式）必须预先设置，然后让这些状态一直生效，直到再次修改它们，所以必须对开发环境进行构建，以初始化相关 OpenGL 状态，并对程序相应的函数进行修改，使 OpenGL 虚拟环境顺利显示。

(1) 创建 MFC 界面，向工程中添加 OpenGL 编译链接库 opengl32.lib、glu32.lib、glaux.lib、glut32.lib，并确保在系统的 system32 目录下有下列动态链接库文件的存在：glu.dll、glu32.dll、glut.dll、glut32.dll、opengl32.dll。

(2) 编写创建函数。因为 OpenGL 仅能在客户窗口中绘图而无法在该窗口的子窗口和兄弟窗口中绘图，所以必须在客户区域中去除<sup>[3]</sup>。本程序在 VIEW 类的 PreCreateWindow() 函数中添加下面语句来实现：  
cs.style=WS\_CLIPCHILDREN|WS\_CLIPSIBLING;

(3) 设置像素格式，创建绘制描述表。这一过程可利用视图类里的 OnCreate() 函数通过用 CreateOpenGL() 来实现。OpenGL 依赖绘制描述表操作硬件，因此调用 OpenGL 命令前，必须创建绘制描述表并使其成为当前的绘制描述表，即：

```

m_hRC=wglCreateContext(hDC);
wglMakeCurrent(hDC, m_hRC);
  
```

(4) 定义当前视区及投影模式等。这些操作在对 WM\_SIZE 消息的处理函数 OnSize() 中体现，在其中调用 glViewport() 函数对创建的窗口变换进行更改以适应窗口大小的改变。还可以通过 gluPerspective() 函数重新设置投影变换，使得当窗口的大小发生改变时，显示在窗口中的场景不会被扭曲。

(5) 填充背景色，进行一些 OpenGL 绘制前的初始化工作。当需要重新设置窗口背景时，产生 WM\_ERASEBKGD 消息。处理该消息的缺省操作是用当前背景色填充整个窗口。OpenGL 用 glClear() 函数来执行类似操作，所以要对 WM\_ERASEBKGD 的消息处理函数做适当改变：注释掉 OnEraseBKGD() 函数中所有的语句，添加 return TRUE 语句，使该函数不执行操作，仅返回 TRUE 值。

(6) 绘制场景，最后删除绘制描述表。所有的

OpenGL 的绘制操作都必须放在 OnDraw () 函数中。在窗口销毁时会触发 WM\_DESTROY 消息的处理函数, 因此我们在该函数中删除绘制描述表。

#### 4.2 对象模型的创建与显示

##### (1) 对象模型的创建

系统模型使用 3DS MAX 建立, 该软件建模精细、功能强大, 但是它不具备人机交互功能。在 Visual C++ 中利用 OpenGL 建立的模型可以进行人机交互, 但用 OpenGL 建模的缺点是无法表达复杂的物体和细节, 仿真效果差, 而且工作量繁杂。我们结合 3DS MAX 和 OpenGL 的建模特点, 用 3DS MAX 制作水雷各个组件和工具的三维模型, 并导出为 3ds 文件, 然后在 Visual C++6.0 环境下通过接口程序读取模型的数据源文件。

在 3DS MAX 中水雷各部分使用扩展几何体建模, 扩展几何体主要是一些较为复杂的几何体, 如倒角立方体、圆柱体等, 利用布尔运算初步建模后, 用 bend 等编辑修改其对模型进行修改和编辑, 实现模型的细节, 使用材质贴图模拟模型金属质感, 并利用 UVW 贴图修改器对材质贴图进行调整, 以求达到最佳效果。

##### (2) 3DS 模型文件的读取与显示

3DS 文件是由许多块(chunk)组成的(大块中镶嵌子块)。由于至今为止, 没有一个官方的文献说明其格式, 所以还有很多未知的块<sup>[4]</sup>。不过这并不影响我们读入 3DS 文件中的模型。因为我们在读入时, 可以根据自己的需要选择性地读入需要的块, 而忽略掉那些不感兴趣或未知的块。这正是块结构给我们带来的好处。一个块由块信息和块数据组成。块信息又由块 ID(两个字节长的标识, 如 4D4D)和块的长度(四个字节, 其实也就是下一个块的偏移字节数)组成。用 VC++ 以十六进制方式打开 3DS 文件可以很清楚的看到其结构。在读入这种块结构(大块中嵌套小块, 而块的结构固定)的文件时, 完全可以用递归的方法实现, 返回上一级(子块读完, 返回父块)的条件则是当前已经读入的块的字节数是否等于块的长度。从父块转向读入其子块, 则可用 switch 语句实现, 通过子块的 ID 判断进入哪个分支<sup>[5]</sup>。程序

流程图如图 3 所示。

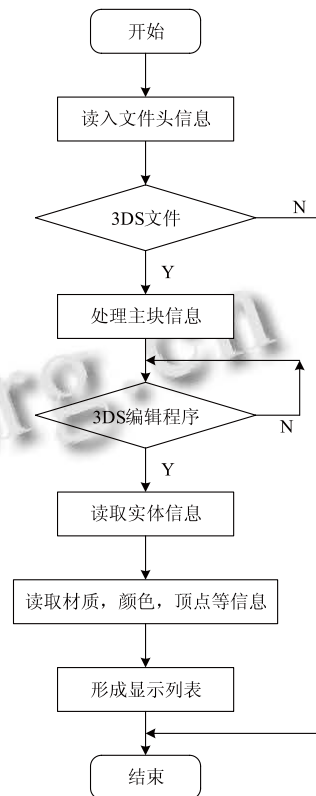


图 3 流程图

读入 3DS 文件通过定义一个 CLoad3DS 类来处理, 其实现的关键代码如下:

```

Class CLoad3DS
{
  CLoad3DS (); //初始化数据成员
  Bool import3DS(t3DModel *pModel, char *strFileName) //装入 3DS 文件到模型结构中
  int GetString(char *) //读一个字符串
  void ReadChunk(tChunk*) //读一个块
  void ProcessNextChunk(t3DModel *pModel, tChunk); //读下一个块
  .....读取模型的材质, 颜色, 顶点等信息
  void ComputeNormals(t3DModel *pModel) //计算对象顶点的法向量
  void CleanUp (); //关闭文件, 释放内存空间
}
  
```

在 OpenGL 中实现图形绘制, 是通过将物体抽象

为笛卡尔坐标系下点，线段，多边形等多种形状的集合，再通过函数 `glBegin()` 与 `glEnd()` 这两个函数中一系列的顶点数据来还原物体的。线的绘制包括了线段，折线和闭和曲线的绘制。对于面的绘制，也是通过 `glBegin()` 与 `glEnd()` 这两个函数来实现的，所不同的是采用不同的状态参数。如三角形片状态参数 `GL_TRIANGLE_STRIP`，体的绘制是通过几个四边形 `GL_QUAD` 来实现的。读入某部件和工具模型如图 4 所示。

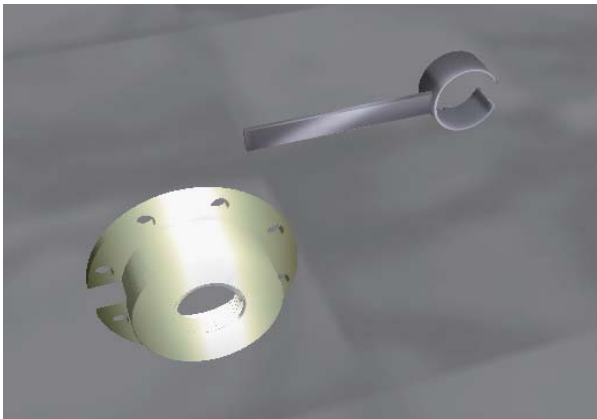


图 4 3DS 模型显示

### 4.3 模型的鼠标选取与操纵

由于水雷仿真训练操作需要用户对屏幕上的部件和工具进行选择，并且移动、旋转这些部件。支持这类交互操作是 OpenGL 的设计目标之一。由于屏幕上绘制的物体一般经历多次旋转、移动和透视变换，因此在三维场景中判断用户所选择的是哪个物体会比较困难。为实现这个目的，我们使用 OpenGL 的选择机制<sup>[6]</sup>。在进入选择模式时，整个场景都将重绘到帧缓冲区，调用绘制图元函数进行重绘时，图元名称将加载到名字堆栈中，完成名字堆栈的创建。然后确定特殊的挑选矩阵，并协同使用投影矩阵，把绘图限制在视口内靠近鼠标双击位置的一个小区域内，当退出选择模式时，OpenGL 以点击记录的形式返回靠近鼠标双击位置的图元列表。点击记录包含的信息，对应名字堆栈中当前的内容。因此，当点击记录返回时，我们可以用它来确定屏幕上的那些图元被用户选择。鼠标选取模型过程中使用

`glSelectBuffer()` 函数制定返回点击记录的数组，`glRenderMode()` 函数进入选择或渲染模式，`glInitNames()` 和 `glPushName()` 函数对名字堆栈进行初始化，`glPickMatrix()` 函数确定挑选矩阵<sup>[7]</sup>。鼠标选取程序流程如图 5 所示。

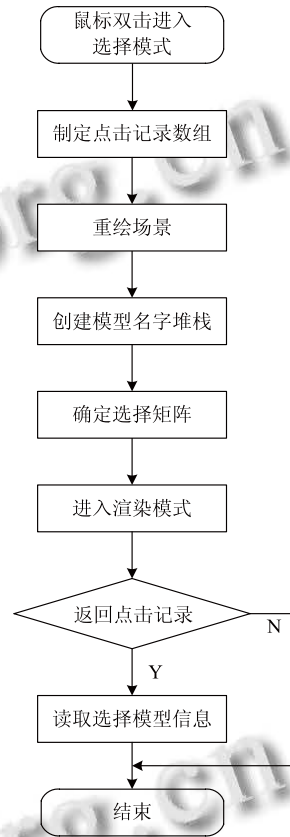


图 5 流程图

选取部件后还必须对所选部件实现平移变换，旋转变换。模型是相对原点来定义绘制，移动坐标原点就能实现平移，其函数为 `glTranslatef(x, y, z)`，该函数用指定对象坐标  $x, y, z$  沿着  $x, y, z$  三轴移动。旋转变换是模型所在的坐标系发生旋转变换，即坐标系的变化将直接导致物体的变化，函数为 `glRotatef(angle, x, y, z)`，该函数中第一个变量 `angle` 为模型旋转的角度，单位为度，后 3 个变量表示以原点  $(0, 0, 0)$  到点  $(x, y, z)$  的连线为轴线逆时针旋转物体。

选取部件后还必须对所选部件实现平移变换，旋转变换。模型是相对原点来定义绘制，移动坐标原点就能实现平移，其函数为 `glTranslatef(x, y, z)`，该函

数用指定对象坐标  $x, y, z$  沿着  $x, y, z$  三轴移动<sup>[8]</sup>。旋转变换是模型所在的坐标系发生旋转变换,即坐标系的变化将直接导致物体的变化,函数为  $glRotatef(\text{angle}, x, y, z)$ ,该函数中第一个变量  $\text{angle}$  为模型旋转的角度,单位为度,后 3 个变量表示以原点  $(0, 0, 0)$  到点  $(x, y, z)$  的连线为轴线逆时针旋转物体。

用户可以通过菜单选择变换操作,如旋转操作、XY 平面和 XZ 平面平移,在旋转操作中,三维模型置于预先定义的虚拟球中,按下鼠标按键,在虚拟球上确定了一点,拖动鼠标就是移动该点,就实现了旋转模型的目的。在平移操作中,将鼠标移动转换为 X 轴、Y 轴或 X 轴、Z 轴的平移量,从而实现模型平移。下面步骤实现了对导入 3ds 模型的旋转和平移变换:

(1) 确定鼠标坐标偏移量转换为三维坐标系平移量和旋转量的转换算法。

(2) 定义变量,如旋转矩阵、平移矩阵及鼠标各个状态的定义,包括按下,点击,拖动等。

(3) 根据各变量的变化来更新旋转矩阵和平移矩阵。

(4) 将变化的结果应用于选定的模型。

## 5 结语

本文给出了一种构建三维仿真系统的基本过程和实现方法。OpenGL 和 3DS MAX 结合实现三维仿真既利用了 3DS MAX 建立模型方便快捷的特点,又利用了 OpenGL 容易实现交互性的特点,将平移、

旋转以及三维模型之间的层次关系和相对运动等运动信息进行集成,在此基础上开发模型的运动仿真功能。三维仿真操作时,模型之间相互独立,省去了大量的实时的逻辑和参数计算,提高了仿真的实时性流畅性。本软件可以方便地在不同平台和系统间移植。利用本文所述方法构建三维仿真系统,简单方便,通用性好,这对于快速开发可视化模拟训练系统具有实际意义。

## 参考文献

- 1 夏艳,过仲阳,章伟伟.基于 OpenGL 的虚拟现实关键技术研究.计算机时代,2008,7:46-50.
- 2 林培炎,冯开平.采用 VC++ 与 OpenGL 的三维场景编辑系统的研究与设计.工程图学学报,2009,5:58-62.
- 3 Dave S, Mason W, Jackie N. OpenGL Programming Guide.北京:机械工业出版社,2009.
- 4 和平鸽工作室.OpenGL 高级编程与可视化系统开发.北京:中国水利出版社,2002.
- 5 董加强,任松.基于 OpenGL 的卫星在轨运行可视化仿真系统设计与实现.计算机测量与控制,2008,16(6):824-826.
- 6 张山山,赵灿辉.OpenGL 在地震动模拟中的应用.矿山测量,2008,8(3):71-74.
- 7 张伟,郑力.数控加工仿真系统的研究现状与发展趋势.机械制造,2007,9(517):7-8.
- 8 邢少葵,伍铁军.基于 VC++ 与 OpenGL 的飞秒激光加工几何仿真系统的研究.机械制造与自动化,2009,38(4):112-115.

(上接第 79 页)

## 6 结束语

本文设计了一种基于 USB2.0 的视频图像处理芯片,通过功能仿真和 FPGA 验证,表明达到了设计要求,完成了设计任务。随着 FPGA 技术不断发展,使用 FPGA 处理图像信息已成为图像处理领域中的一个发展趋势,特别是在对图像处理速度有较高要求的实时图像处理系统中,FPGA 将更能体现其在速度与灵活性方面的优势。

## 参考文献

- 1 阮秋琦.数字图像处理学.北京:电子工业出版社,2001. 212-229.

- 2 袁文波,张皓,唐振中.FPGA 应用开发从实践到提高.北京:北京航空航天大学出版社,2007.88-98.
- 3 柳兰,何娜,黄智伟.基于 USB 接口的数据采集与控制系统设计.电子设计工程,2009,17(6):38-40.
- 4 余永辉,涂巧玲,徐霞,等.基于 USB2.0 和线阵 CCD 的高速数据采集系统设计.压电与声光,2009,31(3):448-450.
- 5 Han JT, Guan Y. Implementation of SoC-PC communication interface based on USB2.0. IEEE International Conference on New Trends in Information and Service Science, 2009:831-834.
- 6 仙云森.基于 FPGA 的图像处理算法研究及硬件设计[硕士学位论文].大连:大连理工大学,2008.