



图形

图像

动画

专 家

# OpenGL

[ 编程与实例 ]

向世明 编著



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
URL: <http://www.phei.com.cn>

图形图像动画专家

# OpenGL 编程与实例

向世明 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内容提要

本书介绍了应用 OpenGL 开发三维图形的方法。内容包括:怎样利用 Visual C++ 5.0 MFC 中的类建立 OpenGL 应用程序的基本格式,建立和控制 2D 形状和 3D 形体、标准二次曲面、Bezier 曲线及曲面、NURBS 曲线及曲面、点光源、无限光源、聚光、基本材质、贴图材质、Mip 贴图、位图字符、文字、OpenGL 动画等,观察物体,操作像素、位图、图像,融合和反走样物体,雾化场景,利用帧缓存进行颜色屏蔽、深度控制、遮掩、全局场景反走样、景深模拟和场景对焦效果处理,利用对话框、菜单、鼠标选择物体、信息反馈实现 OpenGL 基本的交互图形程序设计。最后,本书通过 6 个实用例程说明如何将 OpenGL 的建模技术、图像处理技术和人机交互技术等 3 大技术进行结合。

本书涉及了所有的 OpenGL 核心函数和实用库函数,注重 OpenGL 的内存联系和实际运用。书中的每个工程文件模式固定且十分简洁,内容循序渐进。本书配有 CD-ROM 一张,包含了书中所有源程序和 18 个工程文件所生成的可执行文件等内容,帮助读者学习。本书适合于具有基本计算机图形学基础知识及 Visual C++ 5.0 MFC 编程基础知识的计算机软件开发人员和工程技术人员等阅读。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,翻版必究。

### 图书在版编目(CIP)数据

OpenGL 编程与实例/向世明编著. - 北京:电子工业出版社,1999.9  
(图形图像动画专家/文宏武策划)  
ISBN 7-5053-5625-9

I .0… II .向… III .图形软件-软件接口 .Opengl IV .TP334

中国版本图书馆 CIP 数据核字(1999)第 31908 号

丛 书 名:图形图像动画专家

书 名:OpenGL 编程与实例

编 著 者:向世明

审 者:贾志刚

责任编辑:文宏武

特约编辑:张激扬

印 刷 者:北京京安达明印刷厂

出版发行:电子工业出版社 URL:<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:30.25 字数:774 千字

版 次:1999 年 9 月第 1 版 1999 年 9 月第 1 次印刷

书 号:ISBN 7-5053-5625-9  
TP·2880

印 数:5000 册 定价:50.00 元(含光盘)

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换。  
若书店售缺,请与本社发行部联系调换。电话 68279077

# 前 言

1998年春,《泰坦尼克号》在全球成功放映,使人们首次注意到3D数字特技效果的新浪潮;3D Studio MAX 2.0的推出,使PC机用户看到了新世纪室内动画大制作的曙光;气象服务数字化地显示了天气的变化等等。玩一玩3D游戏,亦或乘上网巴,我们无不体验到3D世界的全新感觉。我们思索:3D产品的技术支持是什么?

3D的技术支持是3D的API。令人振奋的是,自1992年7月SGI公司发布了作为3D API的第一版OpenGL以来,OpenGL经受了实践的检验,越来越获得包括SGI、Microsoft、SUN、IBM等在内的大型计算机企业的支持。今年,OpenGL 1.2版已经发布,并且Microsoft已将其集成于Windows NT 4.0和Windows 98中。

OpenGL集成了所有曲面造型、图形变换、光照、材质、纹理、像素操作、融合、反走样、雾化等等复杂的计算机图形学算法。应用OpenGL绘制图形就像我们使用Windows GDI的MoveTo()函数一样方便。随着微处理器性能的不不断提高和32位操作系统的出现,使得运用OpenGL开发的三维图形质量正在接近工作站的水平。

## 本书的特点

本书采用循序渐进的方法介绍OpenGL的编程技巧,同时注重OpenGL函数的内在联系,力求做到既介绍基本知识又提炼编程思想。本书的程序全部在Visual C++ 5.0 MFC环境中完成,每个例程均介绍OpenGL的一个主题,每个主题中又有许多知识点,每个OpenGL知识点均通过实例来体现。这些实例是通过视窗类(CView类)来集成,通过菜单来观察和学习。因此,每个视窗类均有一个或多个成员变量,它们是枚举成员中的一个,其选定方法是通过响应菜单来完成。对于每一个枚举成员,均有一个视窗类成员函数与之相对应,该函数具体介绍OpenGL的知识点,并由switch语句来组织。这就是本书所采用的固定编程模式。该模式便于集中精力学习OpenGL,而绝不是借此学习Visual C++的知识。本书大量地涉及到为类添加成员变量或成员函数的操作,凡未经指明时,均针对视窗类。

考虑到辅助库函数的可移植性差,本书并不介绍OpenGL辅助库函数的用法。比如,辅助库函数提供了绘制圆环面的函数,但其精度不能交互地控制,因此本书并不用它,而是提供一个更好的绘制函数。在本书诸多例程中,仅一次应用了一个制作茶壶的函数。

## 读者对象

阅读本书并不需要较深的计算机图形学知识和Visual C++的资深编程经验。但是懂得计算机图形学的基本概念和Visual C++ MFC编程的基本方法无疑对于理解OpenGL和快速运用OpenGL将是十分重要的。

如果读者没有这两方面的基本知识,阅读本书也没有任何障碍,因为本书不仅采用循序渐进的介绍方式,而且对于任何Visual C++编程(哪怕是添加一个头文件等工作)都是严格按步骤来进行的。对于计算机图形学的基本概念,相应的地方也有必要的介绍。

为了便于学习,第16章的6个实用程序代码均被完整地列出。因而,本书的设计面向各

层次的计算机图形图像处理的专业人员和一般计算机爱好者。由于 OpenGL 具有高度的开放性、可移植性、网络透明性等特点,OpenGL 可适用于游戏开发、科学计算可视化(运动仿真、有限元分析、流体计算、医学成像、分子化学、汽车等曲面造型、地理信息、石油勘探、气候模型模拟)、虚拟现实技术、影视广告、CAI 等等领域内的广大工程技术人员和教学科研人员学习使用。

本书基于 Windows 操作系统和 Visual C++ 编程环境,所有与 OpenGL 有关的格式和程序控制均可由读者按步骤亲自编程,从而读者可领会 OpenGL 的强大功能和 OpenGL 的实质。

## 本书附带的 CD-ROM

必须指出,即使没有这张 CD-ROM,阅读本书和按步骤进行编程也没有任何障碍,但无疑这张 CD-ROM 必将大大地加快读者阅读的学习进程。CD-ROM 中不仅包含了本书中所有的源程序代码,而且附有 Win32 Application 样板程序等。具体内容参见附录介绍。在本书的第 3 章至第 16 章中,作者并不希望读者去先运行 CD-ROM 中对应的工程文件,而是希望读者按步骤进行编程,以从中体会。

## 所需环境

需要 Windows 95(98)开发环境和 Visual C++ 2.0 以上的开发工具。建议使用 Visual C++ 5.0 或 6.0。本书中的所有例程均在 Visual C++ 5.0 MFC 中完成。

## 主要内容

第 1 章,进入三维真彩世界。三维图形现实的市场需要为 OpenGL 提供了应用的大舞台。

第 2 章,OpenGL 概述。本章的主要内容有:OpenGL 的发展简史、OpenGL 的特点、OpenGL 工作结构和功能、OpenGL 的工作原理、图形操作步骤、OpenGL 函数库、OpenGL 结构、OpenGL 数据类型和 OpenGL 句法等。

第 3 章,OpenGL 极小化工作。就像我们应用 GDI 函数之前要初始化设备描述表对象一样,为使 OpenGL 获得硬件支持,必须建立 OpenGL 的像素格式、当前化着色描述表。本章介绍了如何利用 Visual C++ MFC 基本类和 Windows 消息来解决这一问题。

第 4 章,绘制二维图形。二维图形包括点、线、面。本章介绍了如何生成二维图形及对二维图形的控制方法,比如,如何生成点画线、如何控制线宽、如何定义一个多边形等等基本造型知识。作为扩展,本章还介绍了如何构造三维形体的基本方法,并由二维的面拼接了一个盒子。

第 5 章,OpenGL 变换。场景是由许多物体有序组合而成。将一个三维场景映射成计算机屏幕区域上的一个二维图像需要一系列的变换,OpenGL 集成了这些算法,使编程者可以模拟架设相机、选择场景、调整拍摄角度、剪切场景及投影成像等照相过程。

第 6 章,显示列表。OpenGL 可以通过所谓的显示列表来优化被多次调用的代码。只要后继时间允许,OpenGL 就会执行它们,这对提高应用程序(特别是网络程序)的整体性能将是十分重要的。

第7章,模拟真实世界。光照和材质是生成真实感三维图形的基本要素。本章讨论材质定义、点光源、平行光源和聚光的生成与控制,光的模拟,颜色缓存、深度缓存的操作与应用,着色模式的选择与应用,以及颜色、材质和光三者的相互作用,多光源的相互作用等。通过光照一个盒子和球来应用这些基本知识,并采用对话框的形式实时地控制颜色、颜色屏蔽、光照、材质等参数。通过本章,还可以学习制作 OpenGL 交互程序的基本方法。

第8章,OpenGL 动画。OpenGL 动画采用双缓存机制。事实上,生成 OpenGL 动画的机制是十分简单的,主要应明确运动对象的运动学条件。

第9章,OpenGL 图像处理基础。本章介绍如何操作像素、位图和图像,如何生成位图字符和输出 Windows 已有字库的字体。

第10章,纹理贴图。纹理为真实造型提供了一个灵活的工具。本章不仅介绍纹理的生成、控制与映射等基本知识,而且还将制作一个专门处理位图的类,通过该类,可以轻松地将一个任意给定的 Windows 24 位位图贴到物体的表面,介绍如何将纹理贴图到曲面物体上,如何运用 Mip 贴图等高级纹理贴图技术。

第11章,增强图像效果。它们均属于 OpenGL 高级图像生成技术。本章通过实例介绍了采用融合将多个物体颜色进行混合生成新物体,制作透明物体,对点、线和面进行反走样,以及通过雾化模拟真实自然界的方法。

第12章,实用库函数。本章介绍了 OpenGL 实用库函数。重点介绍标准二次曲面的生成、图像操作和自动 Mip 贴图方法。

第13章,曲线和曲面。Bezier 曲线和曲面,NURBS 曲线和曲面在工程中有着广泛的应用。本章介绍这两类曲线和曲面、纹理曲面等的生成与控制,还介绍了如何制作有洞曲面(连通曲面)的方法。

第14章,高级制作技术。主要内容有:模板缓存的操作与应用,累积缓存的操作与应用。本章通过模板缓存制作了一块正方形掩膜并将一个球绘制在该掩膜上,利用模板缓存实现对两球交集的制作;利用累积缓存进行场景全局反走样、景深模拟和对焦效果处理。

第15章,交互技术。OpenGL 的 3 个基本模式即绘制、选择与反馈模式构成了利用 OpenGL 开发交互式三维图形的基础。主要内容有:选择模式的建立与操作步骤、反馈模式的建立与应用。本章利用选择模式,将绘图区等分成 4 个区域,当鼠标单击一个区域时,启动选择机制并改变该块区域的背景颜色;利用反馈模式绘制着色了两个点,并将这两个点的坐标和颜色值打印在绘图区指定的位置。

第16章,OpenGL 全面制作技术。本章集中了 6 个实用例程,通过它们重点说明了 OpenGL 的 3 大技术:建立模型技术、图像处理技术和人机交互技术,并将这些技术结合起来。第一个例程说明了建立数学模型是解决问题的一个好途径。该例程演示了兔子和乌龟的追逐轨迹以及如何制作分形图形。物体建模技术是最核心的技术也是最难处理的技术。因此,第二和第三个例程说明建立物体几何模型的理论方法,用实例说明怎样去组织好所需数据(几何顶点、颜色、法线和纹理坐标等)及数据结构并与所需要目标结合起来,怎样为人机交互提供接口;怎样实现由数据驱动动画。第二个例程制作一面飘扬的旗帜,第三个例程则显示一个滚动的西瓜。第四个例程说明怎样将画面着色到内存位图设备,说明当指定画面中所需要的子区域时怎样将其输出到外部文件,说明怎样将简单物体(本例程中由一个线框球、几个圆圈构成)组合起来生成一个艺术处理的画面。第五个例程说明了物体阴影的制作方法。重点制作铅笔及其阴影。“铅笔”作为一类物体的总称,以类的形式给出。本例程还提供了另外两个类,

一个处理颜色,一个处理向量的各种运算、点绕轴和任意向量的旋转、点的光透点计算、光的反射方向计算等等。最后铅笔可以进行各种几何变换(封装于类中),因而三个类不经修改可以直接移植应用。最后一个例程说明牛头刨床急回机构的运动仿真。通过该例程说明较一般的基于运动学的运动观察方法和程序设计思想。

参加本书编写工作的同志还有:华中理工大学的邓爱平、杨克升、曹淮、张玲、王峰、史美康、吴诚德,武汉水利电力大学的王考等。在此作者再一次向他们表示深深的谢意。

作者要深深地感谢华中理工大学的罗晋华和黄飞,感谢他们珍贵的指导、诚挚的关心和硬件等条件的支持;感谢华中理工大学的邓爱平,感谢她富有成效的鼓励、全力的支持和所提供的算法;感谢深圳中信公司的张佳鑫、梁三光,感谢他们的指导和所提供的资料。在此作者还要感谢重庆建筑大学的李开海、重庆邮电大学的潘显彬、重庆工业管理学院的张朝晖、重庆大学的钟佑明,感谢他们的鼓励和支持;感谢华中理工大学的史美康和王峰,感谢他们对本书程序的第一次验证和所提供的宝贵建议。

在此,作者还要感谢电子工业出版社的文宏武先生,感谢他所提供的珍贵思路。总之,如果没有这一切,这本书是不可能写成和出版的。

作者

1999年4月于武昌

# 目 录

<b>第 1 章 走进三维真彩世界</b> .....	( 1 )
1.1 从《泰坦尼克号》说起 .....	( 1 )
1.2 遨游虚拟世界 .....	( 2 )
1.3 科学计算的可视化技术 .....	( 3 )
1.4 对 3D Studio MAX 的思考 .....	( 4 )
1.5 3D 图形加速卡 .....	( 4 )
<b>第 2 章 OpenGL 概述</b> .....	( 6 )
2.1 OpenGL 是什么 .....	( 6 )
2.2 OpenGL 的工作结构 .....	( 7 )
2.3 OpenGL 能做什么 .....	( 7 )
2.4 OpenGL 是怎样工作的 .....	( 9 )
2.5 OpenGL 图形操作步骤 .....	( 10 )
2.6 OpenGL 函数及结构 .....	( 11 )
2.7 OpenGL 数据类型 .....	( 12 )
2.8 OpenGL 句法 .....	( 13 )
2.9 OpenGL 状态变量 .....	( 14 )
2.10 齐次坐标表示 .....	( 21 )
<b>第 3 章 极小化 OpenGL</b> .....	( 22 )
3.1 OpenGL 像素格式 .....	( 22 )
3.2 着色描述表 .....	( 24 )
3.3 设置像素格式 .....	( 24 )
3.3.1 填充 PIXELFORMATDESCRIPTOR 结构 .....	( 24 )
3.3.2 设置像素格式 .....	( 25 )
3.3.3 测试像素格式 .....	( 26 )
3.4 创建着色描述表 .....	( 27 )
3.5 极小化 OpenGL .....	( 30 )
3.5.1 创建用户界面 .....	( 31 )
3.5.2 极小化 OpenGL .....	( 32 )
3.6 OpenGL 颜色 .....	( 38 )
3.7 程序清单 .....	( 40 )
<b>第 4 章 绘制二维图形</b> .....	( 49 )
4.1 绘制点 .....	( 49 )
4.2 绘制直线 .....	( 51 )
4.2.1 绘制直线 .....	( 51 )
4.2.2 线型 .....	( 52 )
4.2.3 绘制折线 .....	( 54 )
4.2.4 绘制闭合折线 .....	( 55 )



4.3	绘制多边形	(56)
4.3.1	绘制多边形	(56)
4.3.2	多边形填充模式	(59)
4.4	绘制三角形	(61)
4.4.1	绘制三角形	(61)
4.4.2	绘制三角形片	(62)
4.4.3	绘制三角形扇	(64)
4.5	绘制四边形	(66)
4.5.1	绘制四边形	(66)
4.5.2	绘制四边形片	(66)
4.6	边的可见性	(68)
4.7	由面创建三维形体	(70)
4.8	建立应用程序 TwoDShape	(72)
4.9	TwoDShape 应用程序是如何运行的	(80)
<b>第 5 章</b>	<b>OpenGL 变换</b>	<b>(82)</b>
5.1	建立 GLTrans 工程	(82)
5.2	投影变换	(85)
5.2.1	平行投影	(85)
5.2.2	透视投影	(86)
5.3	几何变换	(88)
5.3.1	一个问题	(88)
5.3.2	平移变换	(88)
5.3.3	平移盒子	(90)
5.3.4	旋转变换	(94)
5.3.5	旋转盒子	(94)
5.3.6	缩放变换	(95)
5.3.7	放大盒子	(95)
5.4	视口变换	(96)
5.5	剪切变换	(97)
5.5.1	剪切变换	(97)
5.5.2	剪切盒子	(98)
5.6	OpenGL 坐标系	(98)
5.7	集成三维物体	(99)
5.7.1	认识平移	(100)
5.7.2	认识旋转和缩放	(101)
5.7.3	集成 8 个盒子	(101)
5.8	矩阵堆栈	(103)
5.8.1	矩阵堆栈的作用	(103)
5.8.2	绘制一辆卡车的平面模型	(104)
5.9	矩阵操作	(107)

<b>第 6 章 显示列表</b> .....	(110)
6.1 创建和执行显示列表 .....	(110)
6.2 管理显示列表 .....	(113)
<b>第 7 章 模拟真实世界</b> .....	(117)
7.1 生成真实图形 .....	(117)
7.2 OpenGL 的颜色模型 .....	(117)
7.3 建立 Light 工程 .....	(119)
7.4 OpenGL 光照 .....	(125)
7.4.1 光的类型 .....	(126)
7.4.2 定义光的属性 .....	(126)
7.4.3 光照盒子和球 .....	(127)
7.5 物体的材质 .....	(132)
7.5.1 定义材质 .....	(133)
7.5.2 辐射光 .....	(134)
7.5.3 材质和光的相互作用 .....	(134)
7.5.4 制作材质 .....	(134)
7.6 盒子就是盒子——法线的作用 .....	(140)
7.7 明暗着色处理 .....	(144)
7.7.1 明暗着色处理 .....	(144)
7.7.2 平滑盒子 .....	(146)
7.8 深度测试 .....	(149)
7.9 光照模型 .....	(151)
7.9.1 全局环境光 .....	(151)
7.9.2 镜面高光 .....	(151)
7.9.3 双面光照 .....	(152)
7.10 聚光.....	(152)
7.10.1 定义聚光.....	(152)
7.10.2 用聚光照射物体 .....	(153)
7.11 创建逻辑调色盘.....	(162)
7.11.1 逻辑调色盘 .....	(162)
7.11.2 创建逻辑调色盘 .....	(163)
7.12 应用程序 Light 使用导示 .....	(169)
<b>第 8 章 OpenGL 动画</b> .....	(172)
8.1 动画生成技术 .....	(172)
8.2 OpenGL 动画工作原理 .....	(172)
8.3 让盒子动起来 .....	(173)
<b>第 9 章 OpenGL 图像处理基础</b> .....	(180)
9.1 位图和图像 .....	(180)
9.1.1 读取图像 .....	(181)
9.1.2 绘制图像 .....	(181)

9.1.3	当前光栅位置 .....	(182)
9.1.4	拷贝图像 .....	(182)
9.1.5	缩放图像 .....	(182)
9.1.6	图像操作工程 Image .....	(182)
9.2	位图字符和文本 .....	(184)
9.2.1	生成位图字符 .....	(185)
9.2.2	生成文本 .....	(186)
<b>第 10 章</b>	<b>纹理贴图 .....</b>	<b>(188)</b>
10.1	纹理贴图技术 .....	(188)
10.2	OpenGL 纹理贴图 .....	(189)
10.2.1	定义纹理 .....	(190)
10.2.2	控制纹理 .....	(190)
10.2.3	说明纹理贴图方式 .....	(191)
10.2.4	定义纹理坐标 .....	(191)
10.2.5	应用纹理贴图技术 .....	(192)
10.3	位图纹理 .....	(196)
10.3.1	设备无关位图(DIB)的数据结构 .....	(196)
10.3.2	创建位图类 .....	(198)
10.3.3	将位图贴到物体上 .....	(203)
10.4	纹理坐标的自动生成 .....	(206)
10.5	Mip 贴图 .....	(209)
<b>第 11 章</b>	<b>增强图像效果 .....</b>	<b>(216)</b>
11.1	OpenGL 融合 .....	(216)
11.1.1	OpenGL 融合操作 .....	(216)
11.1.2	融合四边形 .....	(218)
11.1.3	制作透明玻璃 .....	(222)
11.2	反走样 .....	(226)
11.2.1	反走样思想 .....	(226)
11.2.2	OpenGL 反走样 .....	(226)
11.2.3	反走样的线框球和三角形 .....	(228)
11.3	OpenGL 雾化 .....	(232)
11.3.1	确定雾的属性 .....	(233)
11.3.2	雾化场景 .....	(234)
<b>第 12 章</b>	<b>实用库函数 .....</b>	<b>(238)</b>
12.1	标准二次曲面的生成 .....	(238)
12.1.1	创建二次曲面对象 .....	(238)
12.1.2	绘制二次曲面 .....	(240)
12.2	图像处理 .....	(244)
12.3	自动 Mip 贴图 .....	(246)
<b>第 13 章</b>	<b>曲线和曲面 .....</b>	<b>(252)</b>

13.1	概述	(252)
13.2	Bezier 曲线	(252)
13.2.1	Bezier 曲线	(252)
13.2.2	绘制一条拟正弦曲线	(254)
13.3	NURBS 曲线	(257)
13.3.1	NURBS 曲线的生成	(257)
13.3.2	绘制一条 NURBS 曲线	(260)
13.4	Bezier 曲面	(262)
13.4.1	Bezier 曲面的生成	(262)
13.4.2	绘制网格曲面	(262)
13.4.3	绘制一个填充曲面	(265)
13.5	将位图贴到曲面上	(268)
13.6	NURBS 曲面	(271)
13.7	曲面的剪切	(276)
<b>第 14 章</b>	<b>高级制作技术</b>	<b>(280)</b>
14.1	模板缓存及其应用	(280)
14.1.1	模板缓存的操作	(280)
14.1.2	模板缓存的应用	(281)
14.2	累积缓存及其应用	(286)
14.2.1	累积缓存的操作	(287)
14.2.2	全局反走样	(287)
14.2.3	景深模拟	(293)
<b>第 15 章</b>	<b>交互技术</b>	<b>(297)</b>
15.1	物体的选择	(297)
15.1.1	建立名称堆栈	(297)
15.1.2	拾取物体	(298)
15.1.3	选择模式的应用	(298)
15.2	信息反馈	(308)
15.2.1	反馈模式	(308)
15.2.2	反馈模式的应用	(309)
<b>第 16 章</b>	<b>制作技术与应用实例</b>	<b>(314)</b>
16.1	建立数学模型是解决问题的重要途径	(314)
16.1.1	解决问题的方案	(314)
16.1.2	乌龟和兔子的追逐、分形	(315)
16.1.3	程序清单	(317)
16.2	数据的组织:飘动的旗帜	(328)
16.2.1	解决问题的方案	(328)
16.2.2	飘动的旗帜	(329)
16.2.3	程序清单	(331)
16.3	数据的组织:滚动的西瓜	(349)

16.3.1	解决问题的方案	(349)
16.3.2	滚动的西瓜	(349)
16.3.3	程序清单	(352)
16.4	将画面输出到外部文件	(367)
16.4.1	将画面着色到位图设备	(368)
16.4.2	有选择地输出画面	(372)
16.4.3	程序清单	(375)
16.5	制作阴影	(390)
16.5.1	解决问题的途径	(390)
16.5.2	绘制铅笔及其阴影	(391)
16.5.3	程序清单	(393)
16.6	牛头刨床急回机构的运动仿真	(437)
16.6.1	程序思想	(437)
16.6.2	运动仿真的实现	(439)
16.6.3	程序清单	(441)
附录	配书 CD-ROM 内容介绍	(470)

# 第 1 章 走进三维真彩世界

## 1.1 从《泰坦尼克号》说起

当前的时代是信息化、数字化的时代。数字技术的广泛应用和产业化将在人类历史上引起一场新的革命。近年来,随着计算机软硬件、超大规模集成电路、大容量光盘存储器、数学信号处理技术以及高速数字通信网络的发展,出现了数字视盘机、数字电视、数码照相机、数字电话、数字式直接印刷机等一系列基于数字技术、革命性的新产品。这些新产品在高保真、高清晰度、超大规模的存储容量、灵活的交互性、计算机信息技术处理的便利性和在网上高速传输等方面的优势,大大开阔了人们的眼界,使人们看到了未来数字化时代的曙光和发展趋势,加速了世界各国发展数字技术的步伐。在数字化革命的大潮中,计算机产生图像(CGI)即数字特技效果技术作为数字化技术的一个重要分支正在加速发展。

1998年3月23日晚,电影《泰坦尼克号》于加州好莱坞电影学院奥斯卡受奖大会上获得了11项大奖。其中的“最佳视觉效果”奖完全是基于计算机图形学发展起来的计算机数字艺术的贡献。

当观众观看《泰坦尼克号》中轮船的宏伟场景时,不仅看到了用摄像机拍摄的图像,看到了上万亿字节数据组成的数字图像,而且也看到了由计算机合成制作的一幅幅美妙绝顶的画面。我们不禁要问这是怎么完成的?比如,影片开始时,Rose面前呈现的模拟泰坦尼克号沉没过程:模拟船如何断裂,船头如何扎入海底,船尾沉入海底的过程,轮船从水面消失的动画;在上层甲板上乘客目睹的漂过轮船边的冰山形状;Jack和Rose向外看,目睹船尾和冰山相撞的镜头;当Jack和Rose爬上船尾舵楼时,一些狂暴的数字人在他们背后的上层甲板上来回奔跑的镜头等等。

Microsoft公司的三维动画创作系统SoftImage被重点应用在这部影片的动画制作上,通过摄像机跟踪后,在电脑中确定实拍镜头中物体的位置和摄像机的位置,然后把所获得的数据用于海洋、轮船、人物、场景的绘制。这些数据使得由三维动画制作的乘客的脚能够接触到甲板,行走在走廊上或依靠在栏杆上;由电脑动画制作的海水能够冲刷船体;烟雾能够从烟筒中冒出。这些工作,需要来自于以不同规格建立的模型的运动参数,并进行运动匹配。海洋的制作数据则来自于已建立的一系列用于时辰、风速、海浪起伏、太阳位置及反射参数,使之接近或用于创建一个完整的数字化海洋。海鸥、旗子、鱼、天空中的星星、撞船时从船上落下的碎片、海水与轮船之间溅起的浪花、绳子、烟雾、波动的帆船、冰山、甲板上的冰块、撞碎的玻璃等除人以外的场景均由计算机制作而成。没有人愿意模拟从230英尺高度跳入大海。当你看到船尾开始倾斜到90度,上千人绝望地惊叫着逃命的场景时,能想到其中85%的动画都是由计算机完成的吗?

《泰坦尼克号》是艺术和计算机数字艺术完美结合的杰作,它使人们首次注意到数字特技效果的新浪潮。作为一名图形软件开发与应用工作者,除了惊叹SoftImage以及INFERNO(在《泰坦尼克号》中用于制作影视后期特技效果)等软件的能力外,更应该去思考它们的底层

是如何实现的。可以肯定,OpenGL(Open Graphics Libaray)正是可以完成这项任务的底层工具。同时,还可以肯定,OpenGL必将在CGI技术中发挥重大的作用。

## 1.2 遨游虚拟世界

谈到虚拟现实(Virtual Reality),我们便联想到 LawnMover Man、Johnny Mnemonic 及 Titanic 等电影中的画面。由此而获得的对虚拟世界的认识,使我们对其产生了极大的兴趣。

虚拟现实技术是一门由应用驱动的涉及众多学科的新的实用技术,是集先进的计算机技术、传感与测量技术、微电子技术等为一体的综合集成技术。在计算机技术中,虚拟现实技术的发展又依赖于人工智能、图形学、网络、面向对象、Client/Server、人机交互和高性能计算机技术。该技术的目的是以三维图形为主,从视觉感知、听觉感知角度出发创造出一个让人身临其境的虚拟世界。

远程操作是早期虚拟现实系统工作的动力,美国 Ft. Pierce 海洋学研究所的 Anlhy Yong 及 Donna Kocak 开发出一个高强度激光探测器的远程可视系统。该系统产生水下机器人前面物体的三维图像,这种几何信息可用于指导远程手臂操作。NASA 的国家超级计算机应用中心长期致力于复杂现象的可视化研究。加拿大 Albert 大学为艺术家研制了暑期虚拟环境工作站。在教育培训上,网络虚拟环境可进行游戏、解决各种数学问题。虚拟现实技术甚至可以用来虚拟奥运会。比如,西班牙在 SGI 工作站上做的多用户 VR(Virtual Reality)项目以奥运会体育竞技作为其研究重点,以数字化虚拟方式制作 1992 年巴塞罗那奥运会的足球决赛 4 人电影剧本。

虚拟现实技术的一个核心问题之一是视景的生成。视景的生成首先应生成视景的内部条件,即,建立视景生成的过程和对象的几何模型。接着要完成视景的基本内容表示,比如图形学表示、图像表示、运动图像表示。然后要考虑视景的高层内部表示,比如,三维物体的几何模型、物体的结构层次表示等。最后完成真实视景的生成。OpenGL 正是一种支持视景生成的程序设计语言。

令我们感到十分高兴的是,虚拟现实技术离我们越来越近了。虚拟世界在 Internet 上不断发展壮大。VRML(Virtual Reality Modeling Language)已将三维世界带入了 Web。VRML 对每个人都有用,它是建立和体验一个新世界的通讯媒介,可以模拟现实世界的特征。VRML 浏览器可以带你浏览 web 上的 3D 实境。VRML 生成器能让你生成自己的虚拟世界,供别人寻访。这样人们能够参观你的工厂,访问你的高尔夫球场,在电子咖啡屋和你会谈。VRML 既可以用来建立真实世界场景的模型,也可以建立虚拟的二维、三维世界,就像许多游戏中那样。VRML 也可以用来将抽象的东西形象化。Microsoft 已经采纳了 VRML 2.0 标准,并已利用 DimensionX 基于 Java 的 Liquid Reality 来生成用于 VRML 2.0 浏览的 ActiveX 组件。可在 Windows NT 4.0 上找到 Microsoft 的优化 OpenGL 1.1。Windows 98 已实现了对 OpenGL 的支持。

在众多的 VRML 浏览器中,V. Realm Browser(免费)是一个强大的,基于 OpenGL 的浏览器。V. Realm 的寻访控制在外表和感觉上仍然具有以前模拟器特点的痕迹。V. Realm 有大量的运行参数,从静止和动态模式的着色方式到地心引力和碰撞检测,不一而足。它使用 OpenGL 加速引擎。目前只有 OpenGL 能够提供如此令人印象深刻的视觉效果。如果不借助 OpenGL,使用 V. Realm 将消耗很多时间。WebSpace Navigator(免费)来自于 Template Graph-

ics Software,是第一个可在 Web 上查看和寻访 3D 实境的浏览器。WebSpace 的寻访控制仍然是最好的。由于使用 OpenGL 作为 API 达到了非常优秀的性能和非常高质量的着色视觉效果。这是非 OpenGL 产品需要努力追赶才能达到的性能和质量。WebSpace 不懈可击地对材质进行着色,并且支持 OpenGL 光照模型,具有镜面高光、透明度、辐射和发光效果等等。GLview(非免费)是供图形专业人员使用的浏览器。这些图形专业人员在将应用程序发布之前,需要将 OpenGL 图形流水线细调。

可以肯定地说,OpenGL 正将人们带入三维世界,且随着高新尖科技发展对图形、图像处理的实际需要,OpenGL 必将发挥更加重要的作用。

### 1.3 科学计算的可视化技术

科学计算可视化(Visualization in Scientific Computing)是 1987 年由 B.H. McCormick 等为美国国家科学基金会所写的一份报告(B.H. McCormick. . Visualization in Scientific Computing . Computer Graphics, 1987, Vol. 6) 中首次提出的。短短 10 年,科学计算可视化发展十分迅速,并已成为计算机图形学的一个十分热门的研究方向,在很多领域,比如,有限元分析、流体计算、医学图像、分子化学等领域有着广泛的应用。

科学计算可视化的目标是把由数值计算或实验获得的大量数据按照其自身的物理背景进行有机地结合,用图像的方式来展示数据所表现的内容及其相互关系,便于把握过程的整体演进,发现其内在规律,丰富科学研究的途径,缩短研究周期。在科学计算可视化技术中,算法的研究是十分重要而又十分基础的。与计算机可视化技术的另一研究方向——运动仿真(Athletics Simulation)一样,面临着如何从数据集构造物理模型,如何建立可视化模型以及绘制技术的实现方法等重大问题。可视化技术首先要将抽象数据表示转化成图形或图像图元表示。这些图元一般为点(Point, Pixel)、线(Line, Polyline, Spline)、面(Mesh, NURBS Surface)、图像(Pixel, Bitmap)、体素(Voxel)等等,由这些图元构成可视化模型。可视化模型的绘制技术是在计算机图形学的基础上,对几何元素的共性进行抽取,实现数据的图元化、图元的参数化、图元的属性化、图元属性的参数化。图元的参数化包括几何造型的参数控制、光照模型的参数控制、视点(View Point)的参数控制、投影类型(Orthographic, Perspective)、视觉体(Frustum)的大小等。图元属性则包括颜色(Color)、透明性(Opacity)、材料的光学性能(Ambient, Diffuse, HightLight, Specular, Shininess)及纹理(Texture)等。图元属性参数与其他参数一起,构成可交互的可视化。可视化绘制技术中面绘制技术是最基本的。它以点、线、面及其表示的几何形体如多边形、球面、柱面等为图元。而体绘制技术则是从体元直接绘制出最后的显示图像,不通过中间曲面表示。

不同的数据场所采用的可视化技术也不尽相同,一般是按切片的思想进行降维处理。以体数据场为例,它们分布在研究系统的三维空间,并且随时间的变化而变化,如结构的应力场、人体的温度场等。体数据场的可视化方法就是要把数据映射到几何形状的属性上。因研究的需要,这些时间和空间相关的体数据需要在可视化中显示出来,同时也需要将不同变量类型的数据在显示中分离出来。

可以肯定,由于 OpenGL 是处理专用图形硬件的软件接口,支持可视化实现的语言,必将在计算机可视化技术中发挥其独特的优势。



## 1.4 对 3D Studio MAX 的思考

强有力的关键帧动画制作软件 3D Studio MAX 是基于 OpenGL 标准的(但它所用的 API 为 Autodesk 公司的 Heidi)。该软件在微机上制作的三维场景及动画能够与图形工作站制作的三维场景及动画相媲美。如果读者使用过这个软件,请从图形学的角度思考下面的基本问题。它的点、线、面是如何着色的,球怎么就是球,立方体怎么就是立方体,等等;Bend、Taper、Bever、Twist、Extrude、Lathe 等等变形功能是如何实现的;放样是怎么实现的;光照是怎么实现的;可不可以自己生成标准材质和贴图材质;材质编辑器中双面材质、混合材质、次物体材质等是怎样制成的。其堆栈修改功能给高级三维造型带来了很大的灵活性,它们是怎么对建模进行控制的。运动面板中的诸多运动控制参数是如何控制运动的;动画到底是怎么实现的。View、World、Parent、Local 等坐标系统是怎样进行变换的;物体的变换是怎么实现的。环境效果是怎么实现的。等等。

3D Studio MAX R2 的问世,更使得该软件成为各行各业辅助设计中不可缺少的工具。我们完全可以说它是一个全新设计的超世纪的三维动画及视觉设计软件。在模型制作方面,可以应用 NURBS 顶点及曲线来自由创建复杂的曲面,再也没有点数一定要相同的限制;在动态模拟方面,可以对空气阻力、摩擦力进行参数控制;在分子运动方面,将分子转变为几何物体或指定物体,创造出逼真的爆炸碎片;在电光特效方面,可制作上百种光及镜头相关的特效,如物体发光等;在着色品质方面,可指定场景内的某些物体,以光迹法着色。3D Studio MAX R2 支持 OpenGL 图形加速的硬件。

以上所列问题及其功能,借助 OpenGL 及图形生成算法,都能一一解决(注意,3D Studio MAX 中的很多用语和 OpenGL 是一致的)。但是必须注意,现在的 3D Studio MAX 的动画是关键帧动画,并不是人机交互的,若借助于 Windows 编程环境及 OpenGL 可以做到这一点。

## 1.5 3D 图形加速卡

迄今为止,微处理器的性能几乎每 18 个月翻一番,但与 3D 图形处理芯片的发展速度相比,它就显得黯然失色了。图形处理芯片的性能每 6 个月左右就翻一番,这使得高档的 3D 图形加速卡层出不穷。3D 图形处理加速卡的市场竞争也十分激烈。

专业级 3D 图形加速卡大多采用专门的 3D 图形加速芯片,显示卡有 8M、16M 甚至更大的显示内存。一个或一组 3D 图形加速芯片,在 Windows NT 及 OpenGL 环境下具有异常强大的 3D 图形加速功能,广泛应用于 AutoCAD、3D Studio MAX 及游戏开发中。

OpenGL 作为一个 3D 的 API,尽管在 Windows 95(98)平台、640'480 分辨率、16 位增强色模式、奔腾 133、16M 内存、普通显卡等软件环境和硬件支持下,可以进行绘画,但其绘画速度必然会受到严重影响。如果作为 3D 图形应用开发(例如游戏开发),配置支持 OpenGL 的 3D 图形加速卡将是必要的。这不仅节省大量的着色时间,更重要的是,可以应用高质量的图形性能(如:点抽样,透视校正,双线性映射,雾,镜面反射及 Mip 映射等)来开发出逼真的三维场景。下面简单介绍几种支持 OpenGL 的 3D 图形加速卡。

中国江西科环公司的 GraphExpress MD 三维图形加速卡,使用 GLINT Delta 作为几何流水线加速处理器,一块 GLINT MX 作为渲染处理器。其存储器上有 8M VRAM (Video RAM)