

第 1 章 图形图像概论

章节概述

计算机技术的发展已经将交互式计算机图形、图像变成了一种实用的工具。现在,我们可以看到计算机图形、图像知识被频繁的应用到各种领域,如科学、艺术、工程、商务、教学等方面。本章主要对计算机图形、图像内容进行简单概述,介绍计算机图形、图像应用领域以及发展方向。为了简单、快捷的完成图形、图像编程,我们引入 OpenGL 函数库,在实践环节中主要讲解 OpenGL 函数库与 VS2012 平台链接方法。通过本章的学习,希望大家能够能对计算机图形、图像有一个宏观认识,迅速走进 OpenGL 编程世界。

本章重点与难点

重点:

- (1)了解计算机图形、图像基本概念
- (2)了解计算机图形、图像应用领域

难点:

- (1)OpenGL 开发环境配置
- (2)创建一个简单的 OpenGL 程序

知识正文

1.1 基本概念

计算机图形是利用计算机研究图形的表示、生成、处理和显示的学科。图形通常由点、线、面、体等几何元素和灰度、色彩、线型、线宽等非几何属性组成。从处理技术上来看,图形主要分为两类:一类是基于线条信息表示,如工程图、等高线地图、曲面的线框图等;

另一类是明暗图,也就是通常所说的真实感图形,如图 1.1 所示。

计算机图像有多种含义,其中最常见的定义是指各种图形和影像的总称。在理科的学习以及日常的学习或统计中,图像都是必不可少的组成部分,它为人类构建了一个形象的思维模式,有助于我们学习、思考问题。

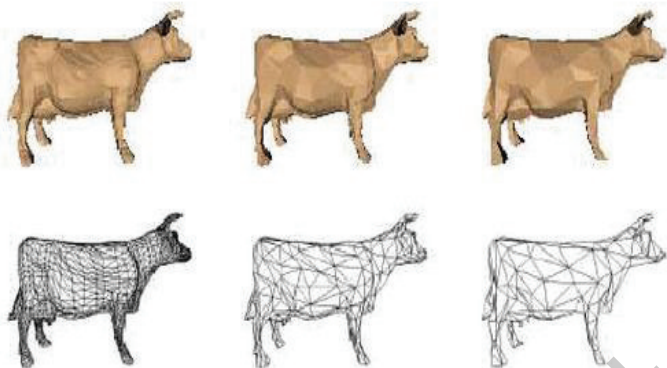


图 1.1 真实感图形以及线框图形

1.2 发展历史

20 世纪 50 年代只有电子管计算机,采用机器编程语言,计算机的功能主要用于科学计算,此时的图形设备仅仅具有输出功能。此时,处于准备和酝酿时期。直到 20 世纪 70 年代,由于光栅显示器的诞生,计算机图形、图像发展才进入兴盛时期,此时区域填充、裁剪、消隐等基本图形概念及其算法纷纷诞生,为图形、图像的发展起到重要推动作用。20 世纪 80 年代后,超大规模集成电路的发展为图形、图像的飞速发展奠定了物质基础。近年来,计算机运算能力的提高,计算机处理速度的加快使得图形、图像的各个研究方向得到了充分发展,现在计算机图形、图像已经广泛应用于动画、科学计算可视化、计算机辅助设计制造、影视娱乐等各个领域。

1.3 应用领域

1.3.1 计算机辅助设计

计算机辅助设计(computer-aided design),比如工程和建筑系统,简称 CAD,现在已经频繁地应用于大楼、汽车、飞机、轮船等产品设计中。

在某些设计中,对象首先是以线框轮廓的形式显示出来,从而展现其整个外形以及该对象的内部特征。线框图可以让设计者很快地看到设计外形以及调整的结果,如图 1.2 所示。建筑设计师使用交互式图形技术来设计平面布局,如图 1.3 所示。在图中标识出了房间、门、窗户及其他建筑特征。

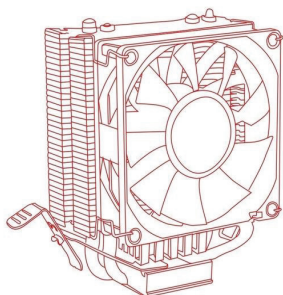


图 1.2 线框图

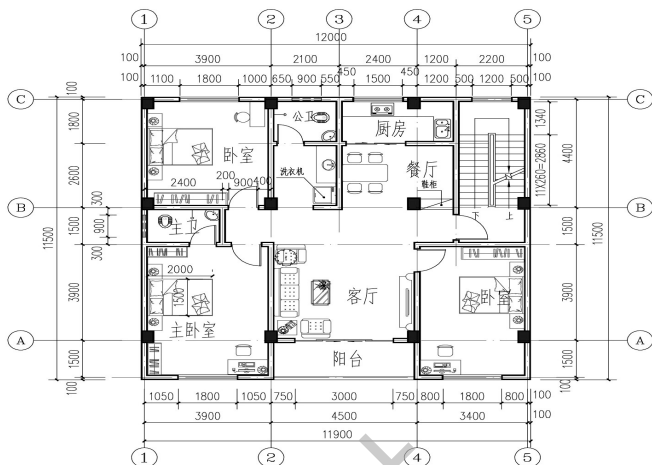


图 1.3 房屋线框图

1.3.2 虚拟现实环境

虚拟现实环境(virtual-reality environment),在虚拟环境中用户可以与三维场景的对象进行交互。该环境中有专门的硬件设备为用户提供三维观察效果,并允许用户在场景中拾取对象,如图 1.4 所示。

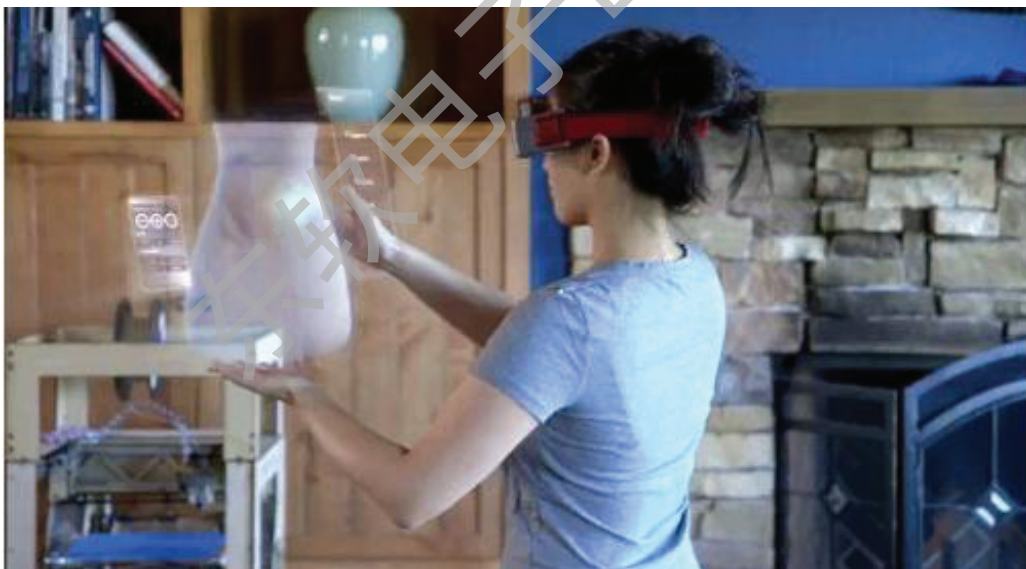


图 1.4 虚拟现实交互图

1.3.3 数据可视化

数据可视化(scientific visualiz)是指为科学计算,工程和医药的数据集生成图形表示。研究分析人员经常要分析大量的信息或研究高度复杂过程的行为。例如,卫星摄像机可以快速的积聚大量数据文件,用户扫描这些数据以确定趋势及相互关系是一个乏味低效的过程,如果将这些数据转换成可视形式,则趋势和模式就可以立刻呈现出来,如图 1.5 所示。



图 1.5 地理信息数据可视化

1.3.4 教育培训

计算机生成的物理模型、财政模型和经济系统模型常用作教学的辅助工具。有些方面的培训要设计专门的硬件系统。例如飞行员、船长大型设备操作员的实习与培训模拟系统，如图 1.6 所示。



图 1.6 飞行员飞行模拟培训

1.3.5 计算机艺术

计算机图形、图像广泛应用于美术和商务艺术应用中，艺术家使用各种计算机方法如商

业软件包、CAD 软件包和动画软件来设计物体的外形并描述物体的运动,如图 1.7 所示。



图 1.7 花瓣艺术设计图

1.3.6 图形用户界面

图形用户界面(graphical user interface, GUI)是目前应用软件必不可少的一部分,它允许用户显示多个矩形屏幕区域窗口,每一个屏幕显示区域可以进行不同的处理,展示各种图形信息,并可以使用鼠标与屏幕进行交互。如图 1.8 所示,展示了典型的图形用户界面,其中包括多个显示窗口,菜单和图符。

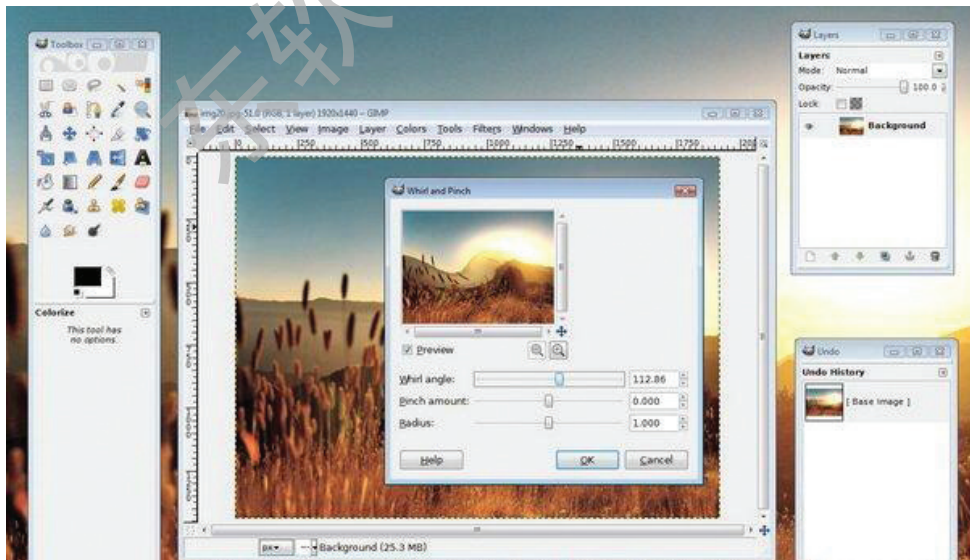


图 1.8 图形用户界面图

1.4 编程软件

图形、图像编程软件有两大类:通用编程软件包和专用应用软件包。专用应用图形软件包是为非程序员设计的,通常是由一组菜单组成,用户通过菜单和程序进行通信。例如工程 CAD 系统,这类应用图形软件包使得们在某些应用中能生成图形、表格而不用关心显示所需的图形函数。相反,通用图形编程软件包提供一个可用于 C、C++、Java 等高级程序语言的图形函数库。通用图形程序设计软件包有 GL(Graphics Library)、OpenGL、VRML (Virtual Reality Modeling Language)等。由于函数库提供了程序设计语言和硬件之间的软件接口,所以这些图形函数被称为计算机图形应用编程接口 (computer - graphics application program interface,CG API)。已开发的计算机图形程序库有很多,有些提供通用的图形函数,有些则以动画、虚拟现实等应用为特定目标。

Open Inventor 软件包采用面向对象函数描述场景,其描述的场景通过 OpenG 来显示。虚拟现实建模语言 (Virtual - Reality Modeling Language,VRML)最初是 Open Inventor 的一个子集,用来建立因特网上虚拟的三维模型。

OpenGL(全写 Open Graphics Library)由 SGI 公司开发,是个定义了跨编程语言、跨平台的编程接口。它用于三维图像(二维的亦可),是一个功能强大,调用方便的底层函数库,其自诞生至今已催生了各种计算机平台及设备上的数千优秀应用程序。常被应用在 CAD、内容创作、能源、娱乐、游戏开发、制造业、制药业及虚拟现实等行业领域中,因为 OpenGL 实现了高性能、极具冲击力的高视觉表现力。因此,本教材中采用 OpengGL 函数来进行图形、图像开发。

1.5 OpenGL 概述

1.5.1 OpenGL 三维图形世界

我们生活在一个充满三维物体的三维世界中,为了使计算机能精确地再现这些物体,我们必须能在三维空间描绘这些物体。我们又生活在一个充满信息的世界中,能否尽快地理解并运用这些信息将直接影响事业的成败,所以我们需要用一种最直接的形式来表示这些信息。最近几年计算机图形、图像的发展使得三维表现技术得以形成,这些三维表现技术使我们能够再现三维世界中的物体。使用三维形体来表示复杂的信息,这种技术就是可视化 (Visualization) 技术。可视化技术使人能够在三维图形世界中直接对具有形体的信息进行操作,和计算机直接交流。这种技术已经把人和机器的力量以一种自然的方式加以统一,这种革命性的变化无疑将极大地提高人们的工作效率。可视化技术赋予人们一种仿真的、三维的并且具有实时交互的能力,这样人们可以在三维图形世界中用以前不可想象的手段来获取信息或发挥自己创造性的思维。机械工程师可以从二维平面图中得以解放直接进入

三维世界,从而很快得到自己设计的三维机械零件模型。医生可以从病人的三维扫描图象分析病人的病灶。军事指挥员可以面对用三维图形技术生成的战场地形,指挥具有真实感的三维飞机、军舰、坦克向目标开进并分析战斗方案的效果。

更令人惊奇的是目前正在发展的虚拟现实技术,它能使人们进入一个三维的、多媒体的虚拟世界,人们可以游历远古时代的城堡,也可以遨游浩瀚的太空。所有这些都依赖于计算机图形学、计算机可视化技术的发展。人们对计算机可视化技术的研究已经历了一个很长的历程,而且形成了许多可视化工具,其中 SGI 公司推出的 GL 三维图形库表现突出,易于使用而且功能强大。利用 GL 开发出来的三维应用软件颇受许多专业技术人员的喜爱,这些三维应用软件已涉及建筑、产品设计、医学、地球科学、流体力学等领域。随着计算机技术的继续发展,GL 已经进一步发展成为 OpenGL,OpenGL 已被认为是高性能图形和交互式视景图像处理的标准。

1.5.2 OpenGL 三维开发环境

OpenGL 包括了 120 个图形函数,开发者可以用这些函数来建立三维模型和进行三维实时交互。OpenGL 经过对 GL 的进一步发展,实现二维和三维的编程技术,在性能上表现得异常优越。它包括建模、变换、光线处理、色彩处理、动画以及更先进的能力,如纹理影射、物体运动模糊等。OpenGL 的这些能力为实现逼真的三维渲染效果、建立交互的三维景观提供了优秀的软件工具。OpenGL 作为一个性能优越的图形、图像应用程序设计界面(API)而适合于广泛的计算环境,从个人计算机到工作站和超级计算机,OpenGL 都能实现高性能的三维图形编程功能。由于许多在计算机界具有领导地位的计算机公司纷纷采用 OpenGL 作为三维图形编程应用程序设计界面,OpenGL 应用程序具有广泛的移植性。因此,OpenGL 已成为目前的三维图形、图像开发标准,是从事三维图形、图像开发工作的技术人员所必须掌握的开发工具。

本教材在编译过程中采用目前流行的 VS2012 平台,将 OpenGL 库链接到 VS2012 平台后进行二维平面及三维场景开发。

1.5.3 OpenGL 库文件

OpenGL 同样包含 100 多个库函数,这些函数都按一定的格式来命名,即每个函数都以 gl 开头。Windows NT 下的 OpenGL 除了具有基本的 OpenGL 函数外,还支持其他四类函数:

- (1)OpenGL 实用库:43 个函数,每个函数以 glu 开头;
- (2)OpenGL 辅助库:31 个函数,每个函数以 aux 开头;
- (3)Windows 专用库函数(WGL):6 个函数,每个函数以 wgl 开头;
- (4)Win32 API 函数:5 个函数,函数前面没有专用前缀。

在 OpenGL 中有 115 个核心函数,这些函数是最基本的,它们可以在任何 OpenGL 的工作平台上应用。函数用于建立各种各样的形体,产生光照效果,进行反走样以及进行纹理映射,进行投影变换等等。由于这些核心函数有许多种形式并能够接受不同类型的参数,实际上这些函数可以派生出 300 多个函数。

OpenGL 的实用函数是比 OpenGL 核心函数更高一层的函数,这些函数是通过调用核心函数来起作用的。这些函数提供了十分简单的用法,从而减轻了开发者的编程负担。OpenGL 的实用函数包括纹理映射、坐标变换、多边形分化、绘制一些如椭球、圆柱、茶壶等简单多边形实体等。这部分函数像核心函数一样在任何 OpenGL 平台都可以应用。

OpenGL 的辅助库是一些特殊的函数,这些函数本来是用于初学者做简单的练习之用,因此这些函数不能在所有的 OpenGL 平台上使用,在 Windows NT 环境下可以使用这些函数。这些函数使用简单,它们可以用于窗口管理、输入输出处理以及绘制一些简单的三维形体。

6 个 WGL 函数是用于连接 OpenGL 和 WindowsNT,这些函数用于在 WindowsNT 环境下的 OpenGL 窗口能够进行渲染着色,在窗口内绘制位图字体以及把文本放在窗口的某一位置等。

最后的 5 个 Win32 函数用于处理象素存储格式和双缓冲区。

1.5.4 OpenGL 绘图方式

OpenGL 是一个高性能的开发软件包,因此 OpenGL 的操作函数十分基本、灵活。例如 OpenGL 中的模型绘制过程就多种多样,内容十分丰富,OpenGL 提供了以下的对三维物体的绘制方式:

- (1) 网格线绘图方式(wireframe)——这种方式仅绘制三维物体的网格轮廓线。
- (2) 深度优先网格线绘图方式(depth_cued)——用网格线方式绘图,增加模拟人眼看物体一样,远处的物体比近处的物体要暗些。
- (3) 反走样网格线绘图方式(antialiased)——用网格线方式绘图,绘图时采用反走样技术以减少图形线条的参差不齐。
- (4) 平面消隐绘图方式(flat_shade)——对模型的隐藏面进行消隐,对模型的平面单元按光照程度进行着色但不进行光滑处理。
- (5) 光滑消隐绘图方式(smooth_shade)——对模型进行消隐按光照渲染着色的过程中再进行光滑处理,这种方式更接近于现实。
- (6) 加阴影和纹理的绘图方式(shadows, textures)——在模型表面贴上纹理甚至于加上光照阴影,使得三维景观呈现出照片的效果。
- (7) 运动模糊的绘图方式(motion-blurred)——模拟物体运动时人眼观察所感觉的动感现象。
- (8) 大气环境效果(atmosphere-effects)——在三维景观中加入如雾等大气环境效果,使人身临其境。
- (9) 深度域效果(depth-of-effects)——类似于照相机镜头效果,模型在聚焦点处清晰,反之则模糊。

这些三维物体绘图和特殊效果处理方式,说明 OpenGL 已经能够模拟比较复杂的三维物体或自然景观。

1.5.5 OpenGL 绘制步骤

在 OpenGL 中进行主要的图形操作直至计算机屏幕上渲染绘制出三维景观的基本

步骤:

(1)根据基本单元建立景物模型,并且对所建立的模型进行数学描述;

(2)把景物模型放在三维空间中的合适的位置,并且设置视点(viewpoint)以观察所感兴趣的景观;

(3)计算模型中所有物体的色彩,其中的色彩根据应用要求来确定,同时确定光照条件、纹理粘贴方式等;

(4)把景物模型的数学描述及其色彩信息转换至计算机屏幕上的像素,这个过程也就是光栅化(rasterization)。

在这些步骤的执行过程中,OpenGL 可能执行其他的一些操作,例如自动消隐处理等。

1.5.6 OpenGL 基本功能

OpenGL 能够对整个三维模型进行渲染着色,从而绘制出与客观世界十分类似的三维景象。另外 OpenGL 还可以进行三维交互、动作模拟等。具体的功能主要有以下这些内容。

(1)模型绘制

OpenGL 能够绘制点、线和多边形。应用这些基本的形体,可以构造出几乎所有的三维模型,OpenGL 通常用模型的多边形的顶点来描述三维模型。

(2)模型观察

在建立了三维景物模型后,就需要用 OpenGL 描述如何观察所建立的三维模型。观察三维模型是通过一系列的坐标变换进行的。模型的坐标变换在使观察者能够在视点位置观察与视点相适应的三维模型景观。在整个三维模型的观察过程中,投影变换的类型决定观察三维模型的观察方式,不同的投影变换得到的三维模型的景象也是不同的。最后的视窗变换则对模型的景象进行裁剪缩放,即决定整个三维模型在屏幕上的图像。

(3)颜色模式的指定

OpenGL 应用了一些专门的函数来指定三维模型的颜色。程序开发者可以选择二个颜色模式,即 RGBA 模式和颜色表模式。在 RGBA 模式中,颜色直接由 RGB 值来指定;在颜色表的模式选择中,颜色值则由颜色表中的一个颜色索引值来指定。开发者还可以选择平面着色和光滑着色二种着色方式对整个三维景观进行着色。

(4)光照应用

用 OpenGL 绘制的三维模型必须加上光照才能更加与客观物体相似。OpenGL 提供了管理四种光(辐射光、环境光、镜面光和漫反射光)的方法,另外还可以指定模型表面的反射特性。

(5)图像效果增强

OpenGL 提供了一系列的增强三维景观的图像效果的函数,这些函数通过反走样、混合和雾化来增强图像的效果。反走样用于改善图像中线段图形的锯齿而更平滑,混合用于处理模型的半透明效果,雾使得影像从视点到远处逐渐褪色,更接近于真实。

(6)位图和图象处理

OpenGL 还提供了专门对位图和图像进行操作的函数。

(7)纹理映射

三维景物因缺少景物的具体细节而显得不够真实,为了更加逼真地表现三维景物,

OpenGL 提供了纹理映射的功能。OpenGL 提供的一系列纹理映射函数使得开发者可以十分方便地把真实图像贴到景物的多边形上,从而可以在视窗内绘制逼真的三维景观。

(8)实时动画

为了获得平滑的动画效果,需要先在内存中生成下一幅图像,然后把已经生成的图像从内存拷贝到屏幕上,这就是 OpenGL 的双缓存技术(double buffer),OpenGL 提供了双缓存技术的一系列函数。

(9)交互技术

目前有许多图形应用需要人机交互,OpenGL 提供了方便的三维图形人机交互接口,用户可以选择修改三维景观中的物体。

1.5.7 OpenGL 库连接

绘制图形前需要将 OpenGL 库链接到 VS2012 平台上,具体链接方法是找到 OpenGL 库中三命名分别为 .h / .lib / .dll 的文件夹将其里面的库文件分别复制到下列地址中,即完成了 OpenGL 库与 VS2012 平台的链接工作。

```
.h C:\Program Files (x86)\Microsoft Visual Studio 11.0\VC\include  
.lib C:\Program Files (x86)\Microsoft Visual Studio 11.0\VC\lib  
.dll C:\Windows\SysWOW64
```

1.6 实践项目——茶壶绘制

1.6.1 茶壶绘制项目构思

该项目作为第一个 OpenGL 测试程序,检测 OpenGL 库与 VS2012 平台链接情况,预想在屏幕中输出一些简单图形。OpenGL 提供了茶壶、圆柱、圆锥等基本几何体的接口函数,我们希望通过直接调用接口函数简化程序从而达到测试链接的目的。

1.6.2 茶壶绘制项目设计

项目设计在屏幕输出一个基本茶壶形状,直接调用函数 `glutWireTeapot()` 实现,如图 1.9 所示。在 OpenGL 中进行主要的图形操作直至计算机屏幕上渲染绘制出三维图形景观的基本步骤如下:

(1)根据基本图形单元建立景物模型,并且对所建立的模型进行数学描述(OpenGL 中把点、线、多边形、图像和位图都作为基本图形单元);

(2)把景物模型放在三维空间中的合适的位置,并且设置视点(viewpoint)以观察所感兴趣的景观;

(3)计算模型中所有物体的色彩,其中的色彩根据应用要求来确定,同时确定光照条件、

纹理粘贴方式等；

(4)把景物模型的数学描述及其色彩信息转换至计算机屏幕上的象素,即光栅化(rasterization)。

1.6.3 茶壶绘制项目实施

OpenGL 绘制流程如图 1.10 所示。



图 1.9 茶壶素描

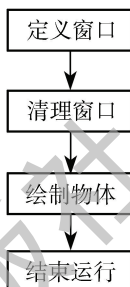


图 1.10 OpenGL 绘制流程图

Step1: 点击文件→新建→项目。

Step2: 选择 win32 控制台项目,如图 1.11 所示。

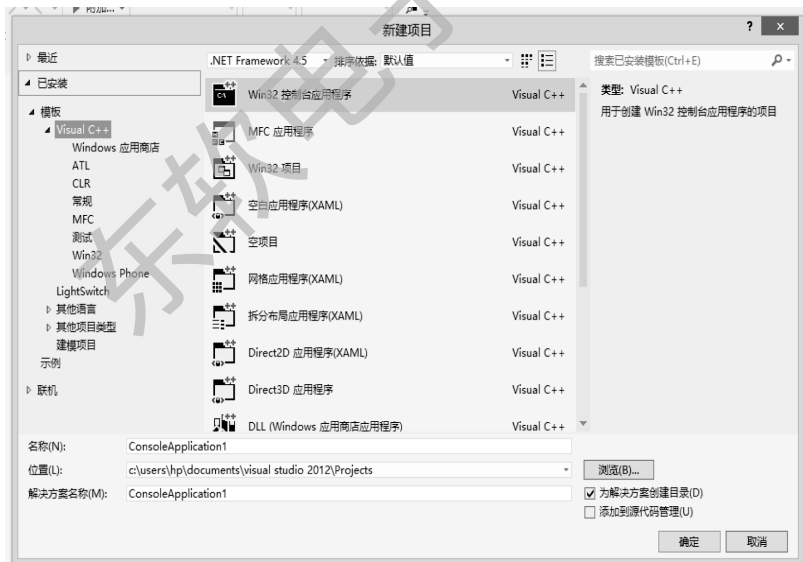


图 1.11 新建 win32 控制台项目

Step3: 点击下一步,如图 1.12 所示。

Step4: 设置链接库,在项目一属性中输入 OpenGL32.lib, GLu32.lib, GLaux.lib, 三个 lib 文件,如图 1.13 所示。



图 1.12 新建项目设置图

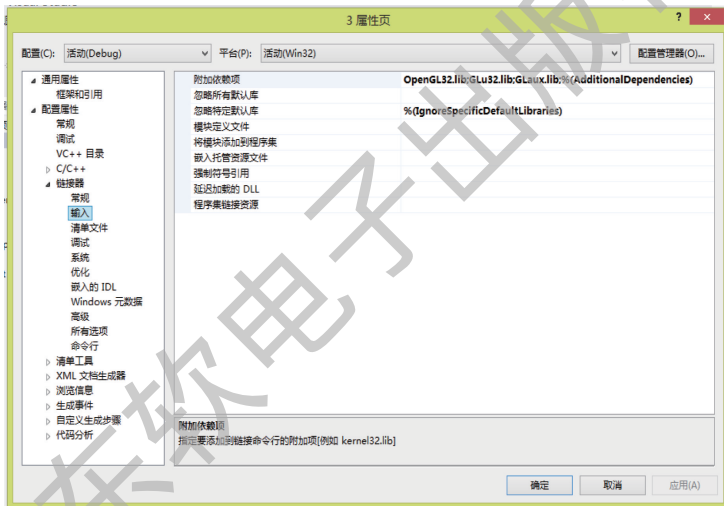


图 1.13 OpenGL 链接库配置

Step5: 在程序中输入代码。

(1) 首先需要包含需要的头文件

```
#include "stdafx.h"
#include "windows.h" // Windows 的头文件
#include <GL\glut.h> // 包含 OpenGL 实用库
#include <stdlib.h>
```

(2) 接着我们对 OpenGL 进行初始设置, 包括清除屏幕所用的颜色, 使用 glColor3f(1.0, 0.0, 0.0) 设置茶壶的颜色为红色, 在颜色设置中, 色彩值的范围从 0.0 到 1.0, 0.0f 表示是最黑情况, 1.0f 表示最亮情况。glClearColor 后的第一个参数是 Red Intensity(红色分量), 第二个是绿色分量, 第三个是蓝色分量, 最后一个是 Alpha 值代表透明度。因此我们使用 glColor3f(0.0, 0.0, 0.0, 0.0) 时, 系统将用黑色清除屏幕。

```
void init() // 此处开始对 OpenGL 进行所有设置
```

```
glClearColor(0.0,0.0,0.0,0.0); // 黑色背景
glColor3f(1.0,0.0,0.0); //设置茶壶为红色
```

(3) 以下是所有的绘制代码,任何你想在屏幕上显示的内容都可以通过此段代码来实现,我们使用 `glClear(GL_COLOR_BUFFER_BIT)` 清除颜色缓存,然后使用 `glut` 辅助函数库 `glutWireTeapot` 绘制一个线框茶壶。

```
void display() // 从这里开始进行所有的绘制
{
    glClear(GL_COLOR_BUFFER_BIT); //清除屏幕中内容
    glutWireTeapot(0.8); //绘制茶壶
    glFlush();}

```

(4) 重置 OpenGL 窗口大小,不管窗口的大小是否已经改变,它都将以透视图的方式运行,场景尺寸被设置成它显示时所在窗口的大小。

```
void myReshape(GLsizei w, GLsizei h) // 重置 OpenGL 窗口大小
{
    glViewport(0, 0, w, h); // 重置当前的视口
    glMatrixMode(GL_PROJECTION); // 选择投影矩阵
    glLoadIdentity(); // 重置投影矩阵
    if (w <= h)
        glOrtho(-1.5, 1.5, -1.5 * (GLfloat)h/(GLfloat)w, 1.5 * (GLfloat)h/(GLfloat)w, -10.0, 10.0);
    else
        glOrtho(-1.5 * (GLfloat)w/(GLfloat)h, 1.5 * (GLfloat)w/(GLfloat)h, -1.5, 1.5, -10.0, 10.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();}

```

(5) 下面是 `main()` 函数是整个程序的入口函数,通过函数调用的方式进行程序初始化后,设置单缓存,屏幕显示色彩方式为 RGB 模式,窗口大小为 (500, 500) 并定义窗口的初始位置,最后调用显示函数 `glutDisplayFunc(display)` 将茶壶显示出来。

```
int main(int argc, char * * argv)
{
    glutInit(&argc, argv);
    //设置窗口显示模式为 RGBA 方式,即彩色方式,并且图形缓存为单缓存
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize(500,500); //定义窗口的宽度,高度
    glutInitWindowPosition(0,0); //定义窗口的初始位置
    glutCreateWindow("OpenGL 程序");
    init();
    glutReshapeFunc (myReshape);
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;}

```

1.6.4 茶壶绘制项目运行

点击运行按钮进行测试,在屏幕中出现茶壶的线框图,说明 OpenGL 库与 VS2012 平台链接成功,效果如图 1.14 所示。

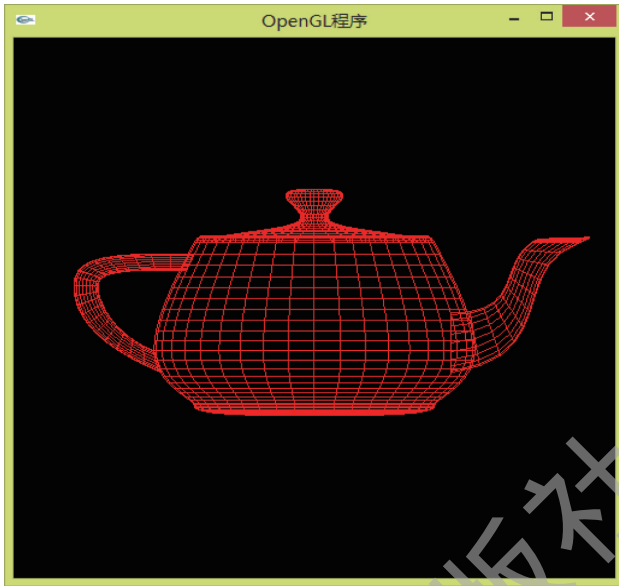


图 1.14 项目演示效果

【本章小结】

本章主要带大家进入 OpenGL 三维世界,简要介绍了 OpenGL 的发展现状以及绘制过程、库的链接方法等内容,最后用一个测试实例完成 OpenGL 与 VS2012 平台的项目搭建工作。

【课后作业】

1. 上网查阅 OpenGL 相关资料,了解最新 OpenGL 发展。
2. 使用 `glutWireTeapot()` 函数在窗口中绘制茶壶。
3. 基于 OpenGL 基本库函数,在显示函数中绘制圆柱、圆锥、立方体等基本几何体,如图 1.15 所示。

```
void display()  
{glClearColor(GL_COLOR_BUFFER_BIT);  
    glutWireTeapot();    //绘制茶壶  
    glutWireCube();      //绘制立方体  
    gluCylinder();       //绘制圆柱体  
    gluSphere();         //绘制圆锥体  
    .....  
    glFlush();}
```

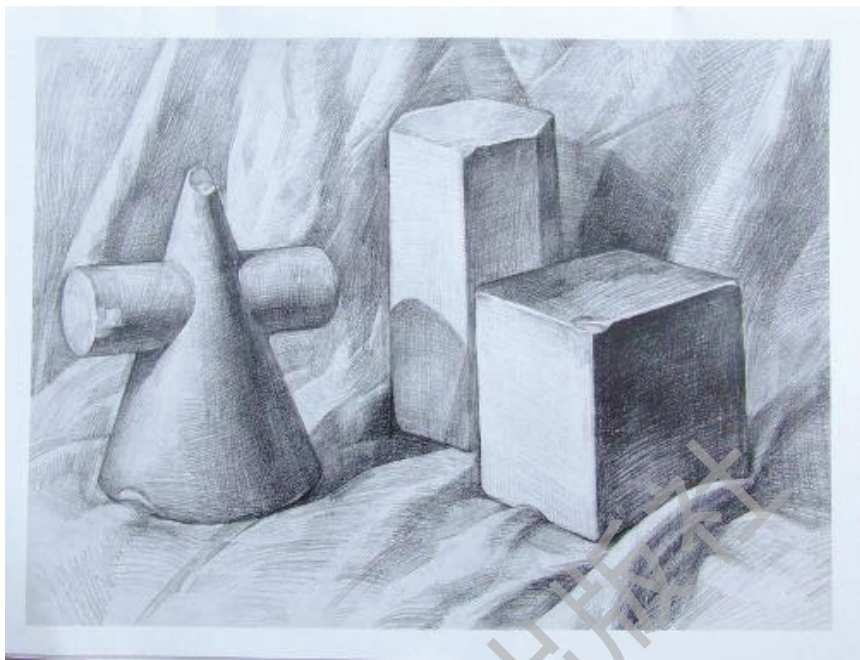


图 1.15 基本几何体素描图

东软电子出版社